Jørgen Bang-Jensen, Gregory Gutin

Digraphs
Theory, Algorithms and Applications

January 28, 2008

Springer-Verlag

Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

We dedicate this book to our parents, especially to our fathers, Børge Bang-Jensen and the late Mikhail Gutin, who, through their very broad knowledge, stimulated our interest in science enormously.

Preface

Graph theory is a very popular area of discrete mathematics with not only numerous theoretical developments, but also countless applications to practical problems. As a research area, graph theory is still relatively young, but it is maturing rapidly with many deep results having been discovered over the last couple of decades.

The theory of graphs can be roughly partitioned into two branches: the areas of undirected graphs and directed graphs (digraphs). Even though both areas have numerous important applications, for various reasons, undirected graphs have been studied much more extensively than directed graphs. One of the reasons is that undirected graphs form in a sense a special class of directed graphs (symmetric digraphs) and hence problems that can be formulated for both directed and undirected graphs are often easier for the latter. Another reason is that, unlike for the case of undirected graphs, for which there are several important books covering both classical and recent results, no previous book covers more than a small fraction of the results obtained on digraphs within the last 25 years. Typically, digraphs are considered only in one chapter or by a few elementary results scattered throughout the book.

Despite all this, the theory of directed graphs has developed enormously within the last three decades. There is an extensive literature on digraphs (more than 3000 papers). Many of these papers contain, not only interesting theoretical results, but also important algorithms as well as applications. This clearly indicates a real necessity for a book, covering not only the basics on digraphs, but also deeper, theoretical as well as algorithmic, results and applications.

The present book is an attempt to fill this huge gap in the literature and may be considered as a handbook on the subject. It starts at a level that can be understood by readers with only a basic knowledge in university mathematics and goes all the way up to the latest research results in several areas (including connectivity, orientations of graphs, submodular flows, paths and cycles in digraphs, generalizations of tournaments and generalizations of digraphs). The book contains more than 700 exercises and a number of applications as well as sections on highly applicable subjects. Due to the fact that we wish to address different groups of readers (advanced undergraduate and graduate students, researchers in discrete mathematics and researchers in various areas including computer science, operations research, artificial intelligence, social sciences and engineering) not all topics will be equally interesting to all potential readers. However, we strongly believe that all readers will find a number of topics of special interest to them.

Even though this book should not be seen as an encyclopedia on directed graphs, we included as many interesting results as possible. The book contains a considerable number of proofs, illustrating various approaches and techniques used in digraph theory and algorithms.

One of the main features of this book is the strong emphasis on algorithms. This is something which is regrettably omitted in some books on graphs. Algorithms on (directed) graphs often play an important role in problems arising in several areas, including computer science and operations research. Secondly, many problems on (directed) graphs are inherently algorithmic. Hence, whenever possible we give constructive proofs of the results in the book. ¿From these proofs one can very often extract an efficient algorithm for the problem studied. Even though we describe many algorithms, partly due to space limitations, we do not supply all the details necessary in order to implement these algorithms. The later (often highly non-trivial step) is a science in itself and we refer the reader to books on data structures.

Another important feature is the vast number of exercises which not only help the reader to train his or her understanding of the material, but also complements the results introduced in the text by covering even more material. Attempting these exercises (most of which appear in a book for the first time) will help the reader to master the subject and its main techniques.

Through its broad coverage and the exercises, stretching from easy to quite difficult, the book will be useful for courses on subjects such as (di)graph theory, combinatorial optimization and graph algorithms. Furthermore, it can be used for more focused courses on topics such as flows, cycles and connectivity. The book contains a large number of illustrations. This will help the reader to understand otherwise difficult concepts and proofs.

To facilitate the use of this book as a reference book and as a graduate textbook, we have added comprehensive symbol and subject indexes. It is our hope that the detailed subject index will help many readers to find what they are looking for without having to read through whole chapters. In particular, there are entries for open problems and conjectures. Every class of digraphs which is studied in the book has its own entry containing the majority of pages on which this class is treated. As sub-entries to the entry 'proof techniques' we have indexed different proof techniques and some representative pages where the technique is illustrated.

Due to our experience, we think that the book will be a useful teaching and reference resource for several decades to come.

Highlights

In this book we cover the majority of the important topics on digraphs ranging from quite elementary to very advanced ones. Below we give a brief outline of some of the main highlights of this book. Readers who are looking for more detailed information are advised to consult the list of contents or the subject index at the end of the book.

Chapter 1 contains most of the terminology and notation used in this book as well several basic results. These are not only used frequently in other chapters, but also serve as illustrations of digraph concepts. Furthermore, several applications of directed graphs are based on these elementary results. One such application is provided in the last section of the chapter. Basic concepts on algorithms and complexity can also be found in the chapter. Due to the comprehensive subject and notation indices, it is by no means necessary to read the whole chapter before moving on to other chapters.

Chapters 2 and 3 cover distances and flows in networks. Although the basic concepts of these two topics are elementary, both theoretical and algorithmic aspects of distances in digraphs as well as flows in networks are of great importance, due to their high applicability to other problems on digraphs and large number of practical applications, in particular, as a powerful modeling tool.

We start with the shortest path problem and a collection of classical algorithms for distances in weighted and unweighted digraphs. The main part of Chapter 2 is devoted to minimization and maximization of distance parameters in digraphs. We conclude the chapter by the following applications: the one-way street problem, the gossip problem and exponential neighbourhood local search, a new approach to find near optimal solutions to combinatorial optimization problems.

In Chapter 3 we cover basic, as well as some more advanced topics on flows in networks. These include several algorithms for the maximum flow problem, feasible flows and circulations, minimum cost flows in networks and applications of flows. We also illustrate the primal-dual algorithm approach for linear programming by applying it to the transportation problem. Although there are several comprehensive books on flows, we believe that our fairly short and yet quite detailed account of the topic will give the majority of readers sufficient knowledge of the area. The reader who masters the techniques described in this chapter will be well equipped for solving many problems arising in practice.

Chapter 4 is devoted to describing several important classes of directed graphs, such as line digraphs, the de Brujn and Kautz digraphs, series-parallel digraphs, generalizations of tournaments and planar digraphs. We concentrate on characterization, recognition and decomposition of these classes. Many properties of these classes are studied in more detail in the rest of the book.

In Chapter 5 we give a detailed account of results concerning the existence of hamiltonian paths and cycles in digraphs as well as some extensions to spanning collections of paths and cycles, in particular, the Gallai-Millgram theorem and Yeo's irreducible cycle factor theorem. We give a series of necessary conditions for hamiltonicity which 'converges' to hamiltonicity. Many results of this chapter deal with generalizations of tournaments. The reader will see that several of these much larger classes of digraphs share various nice properties with tournaments. In particular the hamiltonian path and cycle problems are polynomially solvable for most of these classes. The chapter illustrates various methods (such as the multi-insertion technique) for proving hamiltonicity.

In Chapter 6 we describe a number of interesting topics related to hamiltonicity. These include hamiltonian paths with prescribed end-vertices, pancyclicity, orientations of hamiltonian paths and cycles in tournaments and the problem of finding a strong spanning subdigraph of minimum size in a strong digraph. We cover one of the main ingredients in a recent proof by Havet and Thomassé of Rosenfeld's conjecture on orientations of hamiltonian paths in tournaments and outline a polynomial algorithm for finding a hamiltonian path with prescribed end-vertices in a tournament. We conclude the chapter with a brief introduction of a new approach to approximation algorithms, domination analysis. We illustrate this approach by applying results on hamiltonian cycles in digraphs to the travelling salesman problem.

Connectivity in (di)graphs is a very important topic. It contains numerous deep and beautiful results and has applications to other areas of graph theory and mathematics in general. It has various applications to other areas of research as well. We give a comprehensive account of connectivity topics in Chapters 7, 8 and 9 which deal with global connectivity issues, orientations of graphs and local connectivities, respectively.

Chapter 7 starts from basic topics such as ear-decompositions and the fundamental Menger's theorem and then moves on to advanced topics such as connectivity augmentation, properties of minimally k-(arc)-strong digraphs, highly connected orientations of digraphs and packing directed cuts in digraphs. We describe the splitting technique due to Mader and Lovász and illustrate its usefulness by giving an algorithm, due to Frank, for finding a minimum cardinality set of new arcs whose addition to a digraph D increases its arc-strong connectivity to a prescribed number. We illustrate a recent application due to Cheriyan and Thurimella of Mader's results on minimally k-(arc)-strong digraphs to the problem of finding a small certificate for k-(arc)-strong connectivity. Many of the proofs in the chapter illustrate the important proof technique based on the submodularity of degree functions in digraphs.

Chapter 8 covers important aspects of orientations of undirected and mixed graphs. These include underlying graphs of certain classes of digraphs. Nowhere zero integer flows, a special case of flows, related to (edge-)colourings of undirected graphs is discussed along with Tutte's 5-flow conjecture, which is one of the main open problems in graph theory. The famous theorem by Nash-Williams on orientations preserving a high degree of arc-strong connectivity is described and the weak version dealing with uniform arc-strong connectivities is proved using splitting techniques. Submodular flows form a powerful generalization of circulations in networks. We introduce submodular flows and illustrate how to use this tool to obtain (algorithmic) proofs of many important results in graph theory (including the Lucchesi-Younger theorem and the uniform version of Nash-Williams' orientation theorem). Finally we describe in detail an application, due to Frank, of submodular flows to the problem of orienting a mixed graph in order to maintain a prescribed degree of arc-strong connectivity.

Chapter 9 deals with problems concerning (arc-)disjoint paths and trees in digraphs. We prove that the 2-path problem is $\mathcal{NP}$-complete for arbitrary digraphs, but polynomially solvable for acyclic digraphs. Linkings in planar digraphs, eulerian digraphs as well as several generalizations of tournaments are discussed. Edmonds' theorem on arc-disjoint branchings is proved and several applications of this important result are described. The problem of finding a minimum cost out-branching in a weighted digraph generalizes the minimum spanning tree problem. We describe an extension, due to Frank, of Fulkerson's two-phase greedy algorithm for finding such a branching.

Chapter 10 describes results on (generally) non-hamiltonian cycles in digraphs. We cover cycle spaces, polynomial algorithms to find paths and cycles of 'logarithmic' length, disjoint cycles and feedback sets, including a scheme of a solution of Younger's conjecture by Reed, Robertson, Seymour and Thomas, applications of cycles in digraphs to Markov chains and the even cycle problem, including Thomassen's even cycle theorem. We also cover short cycles in multipartite tournaments, the girth of a digraph, chords of cycles and Ádám's conjecture. The chapter features various proof techniques including several algebraic, algorithmic, combinatorial and probabilistic methods.

Digraphs may be generalized in at least two different ways, by considering edge-coloured graphs or by considering directed hypergraphs. Alternating cycles in 2-edge-coloured graphs generalize the concept of cycles in bipartite digraphs. Certain results on cycles in bipartite digraphs, such as the characterization of hamiltonian bipartite tournaments, are special cases of results for edge-coloured complete graphs. There are useful implications in the other direction as well. In particular, using results on hamiltonian cycles in bipartite tournaments, we prove a characterization of those 2-edge-coloured complete graphs which have an alternating hamiltonian cycle. We describe an application of alternating hamiltonian cycles to a problem in genetics. Generalizations of the classical theorems by Camion, Landau and Rédei to hypertournaments are described.

Chapter 12 contains some topics that were not covered in other chapters. These include: an elementary proof of Seymour's second neighbourhood conjecture in the case of tournaments, various types of orderings of the vertices of digraphs of paired comparisons, kernels, a recent proof by Galvin of the Dinitz conjecture on list colourings using kernels in digraphs, and homomorphisms (an elegant generalization of colouring and also a useful vehicle for studying the borderline between $\mathcal{P}$ and $\mathcal{NP}$-complete problems). We describe basic concepts on matroids as well as questions related to the efficiency of matroid algorithms. We give a brief account on simulated annealing, a broadly applicable meta-heuristic which can be used to obtain near optimal solutions to optimization problems, in particular, on digraphs. We discuss briefly how to implement and tune simulated annealing algorithms so that they may produce good solutions.

Technical remarks

We have tried to rank exercises according to their expected difficulty. Marks range from $(-)$ to $(++)$ in order of increasing difficulty. The majority of exercises have no mark, indicating that they are of moderate difficulty. An exercise marked $(-)$ requires not much more than the understanding of the main definitions and results. A $(+)$ exercise requires a non-trivial idea, or involves substantial work. Finally, the few exercises which are ranked $(++)$ require several deep ideas. Inevitably, this labelling is subjective and some readers may not agree with this ranking in certain cases. Some exercises have a header in bold face, which means that they cover an important or useful result not discussed in the text in detail.

We use the symbol $\square$ to denote the end of a proof, or to indicate that either no proof will be given or is left as an exercise.

A few sections of the book require some basic knowledge of linear programming, in particular the duality theorem. A few others require basic knowledge of probability theory.

We would be grateful to receive comments on the book. They may be sent to us by email to jbj@imada.sdu.dk. We plan to have a web page containing information about misprints and other information about the book, see

http://www.imada.sdu.dk/Research/Digraphs/

Acknowledgements

We wish to thank the following colleagues for helpful assistance and suggestions regarding various versions of the manuscript.

Adrian Bondy, Thomas Böhme, Samvel Darbinyan, Reinhard Diestel, Odile Favaron, Herbert Fleischner, András Frank, Vladimir Gurvich, Frédéric

Havet, Bill Jackson, Hao Li, Martin Loebl, Wolfgang Mader, Crispin Nash-Williams, Jarik Nešetřil, Gert Sabidussi, Paul Seymour, Alexander Schrijver, Stéphan Thomassé, Carsten Thomassen, Bjarne Toft and Ke-Min Zhang.

We extend special thanks to the following colleagues who read parts of the book and provided invaluable input to the project:

Noga Alon, Alex Berg, Jens Clausen, Charles Delorme, Yubao Guo, Jing Huang, Alice Hubenko, Tommy Jensen, Tibor Jordán, Thor Johnson, Ilia Krasikov, Gary MacGillivray, Steven Noble, Erich Prisner, Eng Guan Tay, Meike Tewes, Lutz Volkmann and Anders Yeo.

Of course, any misprint or error that remains is entirely our responsibility.

We wish to thank Springer-Verlag, London and in particular David Anderson, Karen Barker, Beverly Ford, Stephanie Harding, Sally Tickner and Nicolas Wilson for all their help and encouragement. We also thank the anonymous reviewers used by Springer when we submitted our proposal. They provided us with encouragement and useful feedback.

We are grateful to our colleagues and the technical staff at Department of Mathematics and Computer Science, University of Southern Denmark at Odense, for their encouragement and help as well as to the department itself for financial support. In particular, we wish to thank Andrew Swann for all his help with the formatting of the manuscript. We thank the Danish National Research Council for financial support through grant no 9701393.

Last, but most importantly, we wish to thank our families, in particular our wives Lene and Irina, without whose constant support we would never have succeeded in completing this project.

Odense, Denmark
London, UK
August 2000

Jørgen Bang-Jensen
Gregory Gutin

Contents

1. Basic Terminology, Notation and Results

In this chapter we will provide most of the terminology and notation used in this book. Various examples, figures and results should help the reader to better understand the notions introduced in the chapter. The results covered in this chapter constitute a collection of simple yet important facts on digraphs. Most of our terminology and notation are standard. Therefore, some readers may proceed to other chapters after a quick look through this chapter (unfamiliar terminology and notation can be clarified later by consulting the indexes supplied at the end of this book).

In Section 1.1 we provide basic terminology and notation on sets and matrices. Digraphs, directed pseudographs, subdigraphs, weighted directed pseudographs, neighbourhoods, semi-degrees and other basic concepts of directed graph theory are introduced in Section 1.2. Isomorphism and basic operations on digraphs are considered in Section 1.3. In Section 1.4, we introduce walks, trails, paths and cycles, and study some properties of tournaments and acyclic digraphs. Basic notions and results on strong and unilateral connectivity are considered in Section 1.5. Undirected graphs are formally introduced in Section 1.6; in this section we also characterize eulerian directed multigraphs, digraphs with out-branchings (in-branchings) and graphs having strong orientations. Hypergraphs and mixed graphs are defined in Section 1.7. Several important classes of directed and undirected graphs are introduced in Section 1.8. Some basic notions on algorithms are given in Section 1.9. The last section is devoted to a solution of the 2-satisfiability problem using some properties of digraphs.

1.1 Sets, Subsets, Matrices and Vectors

For the sets of real numbers, rational numbers and integers we will use $\mathcal{R}$, $\mathcal{Q}$ and $\mathcal{Z}$, respectively. Also, let $\mathcal{Z}_+ = \{z \in \mathcal{Z} :\ z > 0\}$ and $\mathcal{Z}_0 = \{z \in \mathcal{Z} :\ z \geq 0\}$. The sets $\mathcal{R}_+$, $\mathcal{R}_0$, $\mathcal{Q}_+$ and $\mathcal{Q}_0$ are defined similarly.

The main aim of this section is to establish some notation and terminology on finite sets used in this book. We assume that the reader is familiar with the following basic operations for a pair A, B of sets: the **intersection** $A \cap B$, the **union** $A \cup B$ (if $A \cap B = \emptyset$, then we will sometimes use $A + B$ instead of $A \cup B$), and the **difference** $A \backslash B$ (often denoted by $A - B$). Sets A and

B are **disjoint** if $A \cap B = \emptyset$. We will often not distinguish between a single element set (singleton) $\{x\}$ and the element x itself. For example, we may write $A \cup b$ instead of $A \cup \{b\}$. The **Cartesian product** of sets $X_1, X_2, \ldots, X_p$ is $X_1 \times X_2 \times \ldots \times X_p = \{(x_1, x_2, \ldots, x_p) : \ x_i \in X_i, \ 1 \leq i \leq p\}$.

For sets A, B, $A \subseteq B$ means that A is a subset of B; $A \subset B$ stands for $A \subseteq B$ and $A \neq B$. A non-empty set B is a **proper subset** of a set A if $B \subset A$. A collection $S_1, S_2, \ldots, S_t$ of (not necessarily non-empty) subsets of a set S is a **subpartition** of S if $S_i \cap S_j = \emptyset$ for all $1 \leq i \neq j \leq t$. A subpartition $S_1, S_2, \ldots, S_t$ is a **partition** of S if $\cup_{i=1}^{t} S_i = S$. We will often use the name **family** for a collection of sets. A family $\mathcal{F} = \{X_1, X_2, \ldots, X_n\}$ of sets is **covered** by a set S if $S \cap X_i \neq \emptyset$ for every $i = 1, 2, \ldots, n$. We say that S is a **cover** of $\mathcal{F}$. For a finite set X, the number of elements in X (i.e. its **cardinality**) is denoted by $|X|$. We also say that X is an **$|X|$-element set** (or just an **$|X|$-set**). A set S satisfying a property $\mathcal{P}$ is a **maximum** (**maximal**) set with property $\mathcal{P}$ if there is no set S' satisfying $\mathcal{P}$ and $|S'| > |S|$ ($S \subset S'$). Similarly, one can define minimum (minimal) sets satisfying a property $\mathcal{P}$.

In this book, we will also use **multisets** which, unlike sets, are allowed to have repeated (multiple) elements. The **cardinality** $|S|$ of a multiset M is the total number of elements in S (including repetitions). Often, we will use the words 'family' and 'collection' instead of 'multiset'.

For an $m \times n$ matrix $S = [s_{ij}]$ the **transposed matrix** (of S) is the $n \times m$ matrix $S^T = [t_{kl}]$ such that $t_{ji} = s_{ij}$ for every $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. Unless otherwise specified, the vectors that we use are column-vectors. The operation of transposition is used to obtain row-vectors.

1.2 Digraphs, Subdigraphs, Neighbours, Degrees

A **directed graph** (or just **digraph**) D consists of a non-empty finite set $V(D)$ of elements called **vertices** and a finite set $A(D)$ of ordered pairs of distinct vertices called **arcs**. We call $V(D)$ the **vertex set** and $A(D)$ the **arc set** of D. We will often write $D = (V, A)$ which means that V and A are the vertex set and arc set of D, respectively. The **order** (**size**) of D is the number of vertices (arcs) in D; the order of D will be sometimes denoted by $|D|$. For example, the digraph D in Figure 1.1 is of order and size 6; $V(D) = \{u, v, w, x, y, z\}$, $A(D) = \{(u, v), (u, w), (w, u), (z, u), (x, z), (y, z)\}$. Often the order (size, respectively) of the digraph under consideration is denoted by n (m, respectively).

For an arc (u, v) the first vertex u is its **tail** and the second vertex v is its **head**. We also say that the arc (u, v) **leaves** u and **enters** v. The head and tail of an arc are its **end-vertices**; we say that the end-vertices are **adjacent**,

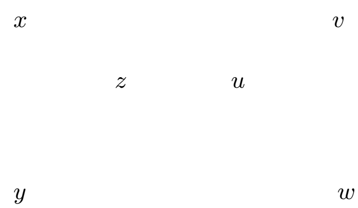


Figure 1.1 A digraph D

i.e. u is adjacent **to**[1] v and v is adjacent to u. If (u, v) is an arc, we also say that u **dominates** v (or v **is dominated by** u) and denote it by $u \to v$. We say that a vertex u is **incident** to an arc a if u is the head or tail of a. We will often denote an arc (x, y) by xy.

For a pair X, Y of vertex sets of a digraph D, we define

$$(X, Y)_D = \{xy \in A(D) : \ x \in X, y \in Y\},$$

i.e. $(X, Y)_D$ is the set of arcs with tail in X and head in Y. For example, for the digraph H in Figure 1.2, $(\{u, v\}, \{w, z\})_H = \{uw\}$, $(\{w, z\}, \{u, v\})_H = \{wv\}$, and $(\{u, v\}, \{u, v\})_H = \{uv, vu\}$.

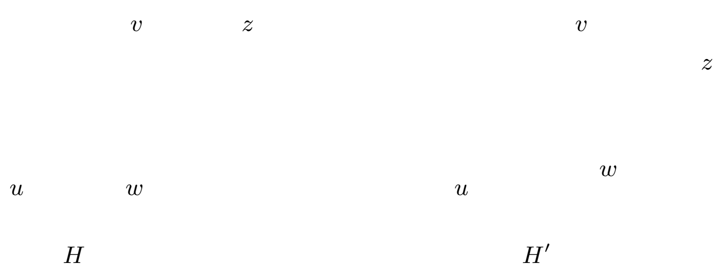


Figure 1.2 A digraph H and a directed pseudograph H'.

For disjoint subsets X and Y of $V(D)$, $X \to Y$ means that every vertex of X dominates every vertex of Y, $X \Rightarrow Y$ stands for $(Y, X)_D = \emptyset$, and $X \mapsto Y$ means that both $X \to Y$ and $X \Rightarrow Y$ hold. For example, in the digraph D of Figure 1.1, $u \to \{v, w\}$, $\{x, y, z\} \Rightarrow \{u, v, w\}$ and $\{x, y\} \mapsto z$.

The above definition of a digraph implies that we allow a digraph to have arcs with the same end-vertices (for example, uv and vu in the digraph H in Figure 1.2), but we do not allow it to contain **parallel** (also called **multiple**) arcs, that is, pairs of arcs with the same tail and the same head, or

[1] Some authors use the convention that x is adjacent to y to mean that there is an arc from x to y, rather than just that there is an arc xy **or** yx in D, as we will do in this book.

loops (i.e. arcs whose head and tail coincide). When parallel arcs and loops are admissible we speak of **directed pseudographs**; directed pseudographs without loops are **directed multigraphs**. In Figure 1.2 the directed pseudograph H' is obtained from H by appending a loop zz and two parallel arcs from u to w. Clearly, for a directed pseudograph D, $A(D)$ and $(X,Y)_D$ (for every pair X,Y of vertex sets of D) are multisets (parallel arcs provide repeated elements). We use the symbol $\mu_D(x,y)$ to denote the number of arcs from a vertex x to a vertex y in a directed pseudograph D. In particular, $\mu_D(x,y)=0$ means that there is no arc from x to y.

We will sometimes give terminology and notation for digraphs only, but we will provide necessary remarks on their extension to directed pseudographs, unless this is trivial.

Below, unless otherwise specified, $D=(V,A)$ is a directed pseudograph. For a vertex v in D, we use the following notation:

$$N_D^+(v)=\{u\in V-v:\ vu\in A\},\ N_D^-(v)=\{w\in V-v:\ wv\in A)\}.$$

The sets $N_D^+(v)$, $N_D^-(v)$ and $N_D(v)=N_D^+(v)\cup N_D^-(v)$ are called the **out-neighbourhood, in-neighbourhood** and **neighbourhood** of v. We call the vertices in $N_D^+(v)$, $N_D^-(v)$ and $N_D(v)$ the **out-neighbours**, **in-neighbours** and **neighbours** of v. In Figure 1.2, $N_H^+(u)=\{v,w\}$, $N_H^-(u)=\{v\}$, $N_H(u)=\{v,w\}$, $N_{H'}^+(w)=\{v,z\}$, $N_{H'}^-(w)=\{u,z\}$, $N_{H'}^+(z)=\{w\}$. For a set $W\subseteq V$, we let

$$N_D^+(W)=\bigcup_{w\in W}N_D^+(w)-W,\ N_D^-(W)=\bigcup_{w\in W}N_D^-(w)-W.$$

That is, $N_D^+(W)$ consists of those vertices from $V-W$ which are out-neighbours of at least one vertex from W. In Figure 1.2, $N_H^+(\{w,z\})=\{v\}$ and $N_H^-(\{w,z\})=\{u\}$.

For a set $W\subseteq V$, the **out-degree** of W (denoted by $d_D^+(W)$) is the number of arcs in D whose tails are in W and heads are in $V-W$, i.e. $d_D^+(W)=|(W,V-W)_D|$. The **in-degree** of W, $d_D^-(W)=|(V-W,W)_D|$. In particular, for a vertex v, the out-degree is the number of arcs, except for loops, with tail v. If D is a digraph (that is, it has no loops or multiple arcs), then the out-degree of a vertex equals the number of out-neighbours of this vertex . We call out-degree and in-degree of a set its **semi-degrees**. The **degree** of W is the sum of its semi-degrees, i.e. the number $d_D(W)=d_D^+(W)+d_D^-(W)$. For example, in Figure 1.2, $d_H^+(u)=2, d_H^-(u)=1, d_H(u)=3, d_{H'}^+(w)=2, d_{H'}^-(w)=4, d_{H'}^+(z)=d_{H'}^-(z)=1, d_H^+(\{u,v,w\})=d_H^-(\{u,v,w\})=1$. Sometimes, it is useful to count loops in the semi-degrees: the **out-pseudodegree** of a vertex v of a directed pseudograph D is the number of all arcs with tail v. Similarly, one can define the **in-pseudodegree** of a vertex. In Figure 1.2, both in-pseudodegree and out-pseudodegree of z in H' are equal to 2.

The **minimum out-degree** (**minimum in-degree**) of D is

$$\delta^+(D)=\min\{d_D^+(x):\ x\in V(D)\}\ \ (\delta^-(D)=\min\{d_D^-(x):\ x\in V(D)\}).$$

The **minimum semi-degree** of D is

$$\delta^0(D) = \min\{\delta^+(D), \delta^-(D)\}.$$

Similarly, one can define the **maximum out-degree** of D, $\Delta^+(D)$, and the **maximum in-degree**, $\Delta^-(D)$. The **maximum semi-degree** of D is

$$\Delta^0(D) = \max\{\Delta^+(D), \Delta^-(D)\}.$$

We say that D is **regular** if $\delta^0(D) = \Delta^0(D)$. In this case, D is also called $\boldsymbol{\delta^0(D)}$**-regular**.

For degrees, semi-degrees as well as for other parameters and sets of digraphs, we will usually omit the subscript for the digraph when it is clear which digraph is meant.

Since the number of arcs in a directed multigraph equals the number of their tails (or their heads) we obtain the following very basic result.

Proposition 1.2.1 *For every directed multigraph D,*

$$\sum_{x \in V(D)} d^-(x) = \sum_{x \in V(D)} d^+(x) = |A(D)|.$$

□

Clearly, this proposition is valid for directed pseudographs if in-degrees and out-degrees are replaced by in-pseudodegrees and out-pseudodegrees.

A digraph H is a **subdigraph** of a digraph D if $V(H) \subseteq V(D)$, $A(H) \subseteq A(D)$ and every arc in $A(H)$ has both end-vertices in $V(H)$. If $V(H) = V(D)$, we say that H is a **spanning subdigraph** (or a **factor**) of D . The digraph L with vertex set $\{u, v, w, z\}$ and arc set $\{uv, uw, wz\}$ is a spanning subdigraph of H in Figure 1.2. If every arc of $A(D)$ with both end-vertices in $V(H)$ is in $A(H)$, we say that H is **induced** by $X = V(H)$ (we write $H = D\langle X\rangle$) and call H an **induced** subdigraph of D. If L is a non-induced subdigraph of D, then there is an arc xy such that $x, y \in V(L)$ and $xy \in A(D) - A(L)$. Such an arc xy is called a **chord** of L (in D). The digraph G with vertex set $\{u, v, w\}$ and arc set $\{uw, wv, vu\}$ is a subdigraph of the digraph H in Figure 1.2; G is neither a spanning subdigraph nor an induced subdigraph of H. The digraph G along with the arc uv (which is a chord of G) is an induced subdigraph of H. For a subset $A' \subseteq A(D)$ the subdigraph **arc-induced** by A' is the digraph $D\langle A'\rangle = (V', A')$, where V' is the set of vertices in V which are incident with at least one arc from A'. For example, in Figure 1.2, $H\langle\{zw, uw\}\rangle$ has vertex set $\{u, w, z\}$ and arc set $\{zw, uw\}$. If H is a subdigraph of D, then we say that D is a **superdigraph** of H.

It is trivial to extend the above definitions of subdigraphs to directed pseudographs. To avoid lengthy terminology, we call the ‘parts’ of directed pseudographs just **subdigraphs** (instead of, say, directed subpseudographs).

For vertex-disjoint subdigraphs H, L of a digraph D, we will often use the shorthand notation $(H, L)_D$, $H \rightarrow L$, $H \Rightarrow L$ and $H \mapsto L$ instead of $(V(H), V(L))_D$, $V(H) \rightarrow V(L)$, $V(H) \Rightarrow V(L)$ and $V(H) \mapsto V(L)$.

A **weighted directed pseudograph** is a directed pseudograph D along with a mapping $c: A(D) \rightarrow \mathcal{R}$. Thus, a weighted directed pseudograph is a triple $D = (V(D), A(D), c)$. We will also consider **vertex-weighted directed pseudographs**, i.e. directed pseudographs D along with a mapping $c: V(D) \rightarrow \mathcal{R}$. (See Figure 1.3.) If a is an **element** (i.e. a vertex or an arc) of a weighted directed pseudograph $D = (V(D), A(D), c)$, then $c(a)$ is called the **weight** or the **cost** of a . An (unweighted) directed pseudograph can be viewed as a (vertex-)weighted directed pseudograph whose elements are all of weight one. For a set B of arcs of a weighted directed pseudograph $D = (V, A, c)$, we define the weight of B as follows: $c(B) = \sum_{a \in B} c(a)$. Similarly, one can define the weight of a set of vertices in a vertex-weighted directed pseudograph. The **weight of a subdigraph** H of a weighted (vertex-weighted) directed pseudograph D is the sum of the weights of the arcs in H (vertices in H). For example, in the weighted directed pseudograph D in Figure 1.3 the set of arcs $\{xy, yz, zx\}$ has weight 9.5 (here we have assumed that we used the arc zx of weight 1). In the directed pseudograph H in Figure 1.3 the subdigraph $U = (\{u, x, z\}, \{xz, zu\})$ has weight 5.

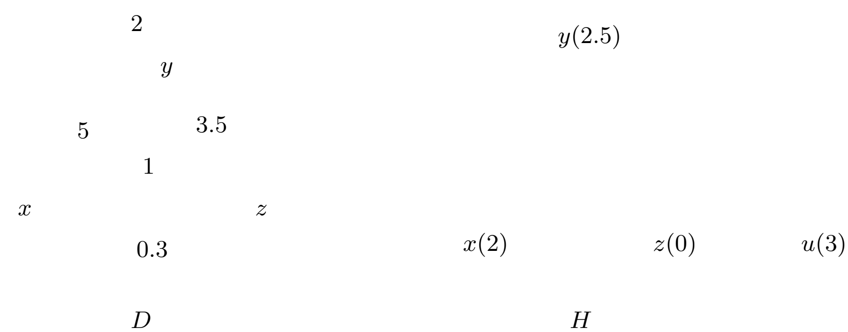


Figure 1.3 Weighted and vertex-weighted directed pseudographs (the vertex weights are given in brackets).

1.3 Isomorphism and Basic Operations on Digraphs

Suppose $D = (V, A)$ is a directed multigraph. A directed multigraph obtained from D by **deleting multiple arcs** is a digraph $H = (V, A')$ where $xy \in A'$ if and only if $\mu_D(x, y) \geq 1$. Let xy be an arc of D. By **reversing the arc** xy, we mean that we replace the arc xy by the arc yx. That is, in the resulting directed multigraph D' we have $\mu_{D'}(x,y) = \mu_D(x,y) - 1$ and $\mu_{D'}(y,x) = \mu_D(y,x) + 1$.

A pair of (unweighted) directed pseudographs D and H are **isomorphic** (denoted by $D \cong H$) if there exists a bijection $\phi : V(D) \to V(H)$ such that $\mu_D(x,y) = \mu_H(\phi(x), \phi(y))$ for every ordered pair x, y of vertices in D. The mapping ϕ is an **isomorphism**. Quite often, we will not distinguish between isomorphic digraphs or directed pseudographs. For example, we may say that there is only one digraph on a single vertex and there are exactly three digraphs with two vertices. Also, there is only one digraph of order 2 and size 2, but there are two directed multigraphs and six directed pseudographs of order and size 2 (Exercise 1.4). For a set of directed pseudographs Ψ, we say that a directed pseudograph D **belongs** to Ψ or is a **member** of Ψ (denoted $D \in \Psi$) if D is isomorphic to a directed pseudograph in Ψ. Since we usually do not distinguish between isomorphic directed pseudographs, we will often write $D = H$ instead of $D \cong H$ for isomorphic D and H.

In case we do want to distinguish between isomorphic digraphs, we speak of **labeled digraphs**. In this case, a pair of digraphs D and H is indistinguishable if and only if they completely coincide (i.e. $V(D) = V(H)$ and $A(D) = A(H)$). In particular, there are four labeled digraphs with vertex set $\{1,2\}$. Indeed, the labeled digraphs $(\{1,2\}, \{(1,2)\})$ and $(\{1,2\}, \{(2,1)\})$ are distinct, even though they are isomorphic.

The **converse** of a directed multigraph D is the directed multigraph H which one obtains from D by reversing all arcs. It is easy to verify, using only the definitions of isomorphism and converse, that a pair of directed multigraphs are isomorphic if and only if their converses are isomorphic (Exercise 1.9). To obtain subdigraphs, we use the following operations of **deletion**. For a directed multigraph D and a set $B \subseteq A(D)$, the directed multigraph $D - B$ is the spanning subdigraph of D with arc set $A(D) - B$. If $X \subseteq V(D)$, the directed multigraph $D - X$ is the subdigraph induced by $V(D) - X$, i.e. $D - X = D\langle V(D) - X\rangle$.For a subdigraph H of D, we define $D - H = D - V(H)$. Since we do not distinguish between a single element set $\{x\}$ and the element x itself, we will often write $D - x$ rather than $D - \{x\}$. If H is a non-induced subdigraph of D, we can construct another subdigraph H' of D by adding a chord a of H; $H' = H + a$.

Let G be a subdigraph of a directed multigraph D. The **contraction** of G in D is a directed multigraph D/G with $V(D/G) = \{g\} \cup (V(D) - V(G))$, where g is a 'new' vertex not in D, and $\mu_{D/G}(x,y) = \mu_D(x,y)$,

$$\mu_{D/G}(x,g) = \sum_{v \in V(G)} \mu_D(x,v), \ \mu_{D/G}(g,y) = \sum_{v \in V(G)} \mu_D(v,y)$$

for all distinct vertices $x, y \in V(D) - V(G)$. (Note that there is no loop in D/G.) Let $G_1, G_2, \ldots, G_t$ be vertex-disjoint subdigraphs of D. Then

$$D/\{G_1, G_2, \ldots, G_t\} = (\ldots((D/G_1)/G_2)\ldots)/G_t.$$

Clearly, the resulting directed multigraph $D/\{G_1, G_2, \ldots, G_t\}$ does not depend on the order of $G_1, G_2, \ldots, G_t$. Contraction can be defined for sets of vertices, rather than subdigraphs. It suffices to view a set of vertices X as a subdigraph with vertex set X and no arcs. Figure 1.4 depicts a digraph H and the contraction H/L, where L is the subdigraph of H induced by the vertices y and z.

To construct 'bigger' digraphs from 'smaller' ones, we will often use the following operation called **composition**. Let D be a digraph with vertex set $\{v_1, v_2, \ldots, v_n\}$, and let $G_1, G_2, \ldots, G_n$ be digraphs which are pairwise vertex-disjoint. The composition $D[G_1, G_2, \ldots, G_n]$ is the digraph L with vertex set $V(G_1) \cup V(G_2) \cup \ldots \cup V(G_n)$ and arc set $(\cup_{i=1}^{n} A(G_i)) \cup \{g_i g_j : g_i \in V(G_i), g_j \in V(G_j), v_i v_j \in A(D)\}$. Figure 1.5 shows the composition $T[G_x, G_l, G_v]$, where G_x consists of a pair of vertices and an arc between them, G_l has a single vertex, G_v consists of a pair of vertices and the pair of mutually opposite arcs between them, and the digraph T is from Figure 1.4.

y ℓ

x z x

v v

H $T = H/L$, $L = H\langle\{y, z\}\rangle$

Figure 1.4 Contraction.

G_ℓ

G_x

G_v

Figure 1.5 $T[G_x, G_\ell, G_v]$

Let Φ be a set of digraphs. A digraph D is ***Φ*-decomposable** if D is a member of Φ or $D = H[S_1, \ldots, S_h]$ for some $H \in \Phi$ with $h = |V(H)| \geq 2$

and some choice of digraphs $S_1, S_2, \ldots, S_h$ (we call this decomposition a **Φ-decomposition**). A digraph D is called **totally Φ-decomposable** if either $D \in \Phi$ or there is a Φ-decomposition $D = H[S_1, \ldots, S_h]$ such that $h \geq 2$, and each S_i is totally Φ-decomposable. In this case, if $D \notin \Phi$, a Φ-decomposition of D, Φ-decompositions $S_i = H_i[S_{i1}, \ldots, S_{ih_i}]$ of all S_i which are not in Φ, Φ-decompositions of those of S_{ij} which are not in Φ, and so on, form a sequence of decompositions which will be called a **total Φ-decomposition** of D. If $D \in \Phi$, we assume that the (unique) total Φ-decomposition of D consists of itself.

To illustrate the last paragraph of definitions, consider $\Psi = \{\overleftrightarrow{K}_1, \overleftrightarrow{K}_2, D_2\}$, where $\overleftrightarrow{K}_1$ is the digraph with a single vertex, $\overleftrightarrow{K}_2$ is the (complete) digraph with two vertices and two arcs, and D_2 has two vertices $\{1, 2\}$ and the arc $(1, 2)$. Construct the digraph D by deleting from the digraph in Figure 1.5 the pair of arcs going from G_ℓ to G_x. The digraph D is totally Ψ-decomposable. Indeed, $D = D_2[D_2, Q]$ is a Ψ-decomposition of D, where Q is the subdigraph of D induced by $V(G_\ell) \cup V(G_v)$. Moreover, $Q = D_2[\overleftrightarrow{K}_1, \overleftrightarrow{K}_2]$ is a Ψ-decomposition of Q. The above two decompositions form a total Φ-decomposition of D.

If $D = H[S_1, \ldots, S_h]$ and none of the digraphs $S_1, \ldots, S_h$ has an arc, then D is an **extension** of H. Distinct vertices x, y are **similar** if x, y have the same in- and out-neighbours in $D - \{x, y\}$. For every $i = 1, 2, \ldots, h$, the vertices of S_i are similar in D. For any set Φ of digraphs, Φ^{ext} denotes the (infinite) set of all extensions of digraphs in Φ, which are called **extended** Φ-digraphs. We say that Φ is **extension-closed** if $\Phi = \Phi^{ext}$.

The **Cartesian product** of a family of digraphs $D_1, D_2, \ldots, D_n$, denoted by $D_1 \times D_2 \times \ldots \times D_n$ or $\prod_{i=1}^n D_i$, where $n \geq 2$, is the digraph D having

$$\begin{aligned} V(D) &= V(D_1) \times V(D_2) \times \ldots \times V(D_n) \\ &= \{(w_1, w_2, \ldots, w_n) : \ w_i \in V(D_i), i = 1, 2, \ldots, n\} \end{aligned}$$

and a vertex $(u_1, u_2, \ldots, u_n)$ dominates a vertex $(v_1, v_2, \ldots, v_n)$ of D if and only if there exists an $r \in \{1, 2, \ldots, n\}$ such that $u_r v_r \in A(D_r)$ and $u_i = v_i$ for all $i \in \{1, 2, \ldots, n\} - \{r\}$. (See Figure 1.6.)

The operation of **splitting** a vertex v of a directed multigraph D consists of replacing v by two (new) vertices u, w so that uw is an arc, all arcs of the form xv by arcs xu and all arcs of the form vy by wy. The **subdivision** of an arc uv of D consists of replacing uv by two arcs uw, wv, where w is a new vertex. If H can be obtained from D by subdividing one or more arcs (here we allow subdividing arcs that are already subdivided), then H is a **subdivision** of D. For a positive integer p and a digraph D, the **pth power** D^p of D is defined as follows: $V(D^p) = V(D)$, $x{\rightarrow}y$ in D^p if $x \neq y$ and there are $k \leq p - 1$ vertices $z_1, z_2 \ldots, z_k$ such that $x{\rightarrow}z_1{\rightarrow}z_2{\rightarrow}\ldots{\rightarrow}z_k{\rightarrow}y$ in D. According to this definition $D^1 = D$. For example, for the digraph $H_n = (\{1, 2, \ldots, n\}, \{(i, i + 1) : \ i = 1, 2, \ldots, n - 1\})$,

$(2, b)$

$(1, b)$ $(3, b)$

2 b

$(2, a)$

1 3 a

$(1, a)$ $(3, a)$

D H $D \times H$

Figure 1.6 The Cartesian product of two digraphs.

we have $H_n^2 = (\{1, 2, \dots, n\}, \{(i, j) :\ 1 \le i < j \le i + 2 \le n\} \cup \{(n-1, n)\})$. See Figure 1.7 for the second power of a digraph.

D D^2

Figure 1.7 A digraph D and its second power D^2.

Let H and L be a pair of directed pseudographs. The **union** $H \cup L$ of H and L is the directed pseudograph D such that $V(D) = V(H) \cup V(L)$ and $\mu_D(x, y) = \mu_H(x, y) + \mu_L(x, y)$ for every pair x, y of vertices in $V(D)$. Here we assume that the function μ_H is naturally extended, i.e. $\mu_H(x, y) = 0$ if at least one of x, y is not in $V(H)$ (and similarly for μ_L). Figure 1.8 illustrates this definition.

1.4 Walks, Trails, Paths, Cycles and Path-Cycle Subdigraphs

In the following, D is always a directed pseudograph, unless otherwise specified. A **walk** in D is an alternating sequence $W = x_1 a_1 x_2 a_2 x_3 \dots x_{k-1} a_{k-1} x_k$ of vertices x_i and arcs a_j from D such that the tail of a_i is x_i and the head of a_i is x_{i+1} for every $i = 1, 2, \dots, k-1$. A walk W is **closed** if $x_1 = x_k$,

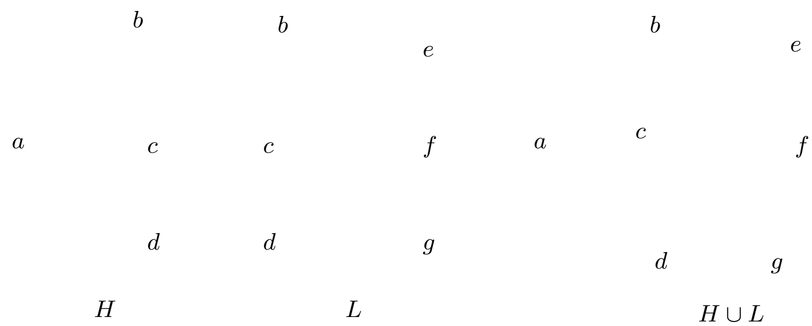


Figure 1.8 The union $D = H \cup L$ of the directed pseudographs H and L.

and **open** otherwise. The set of vertices $\{x_1, x_2, \ldots, x_k\}$ is denoted by $V(W)$; the set of arcs $\{a_1, a_2, \ldots, a_{k-1}\}$ is denoted by $A(W)$. We say that W is a walk **from** x_1 **to** x_k or an $\boldsymbol{(x_1, x_k)}$**-walk**. If W is open, then we say that the vertex x_1 is the **initial** vertex of W, the vertex x_k is the **terminal** vertex of W, and x_1 and x_k are **end-vertices** of W. The **length** of a walk is the number of its arcs. Hence the walk W above has length $k-1$. A walk is **even** (**odd**) if its length is even (odd). When the arcs of W are defined from the context or simply unimportant, we will denote W by $x_1x_2\ldots x_k$.

A **trail** is a walk in which all arcs are distinct. Sometimes, we identify a trail W with the *directed pseudograph* $(V(W), A(W))$, which is a *subdigraph* of D. If the vertices of W are distinct, W is a **path**. If the vertices $x_1, x_2, \ldots, x_{k-1}$ are distinct, $k \geq 3$ and $x_1 = x_k$, W is a **cycle**. Since paths and cycles are special cases of walks, the **length** of a path and a cycle is already defined. The same remark is valid for other parameters and notions, e.g. an $\boldsymbol{(x, y)}$**-path**. A path P is an $\boldsymbol{[x, y]}$**-path** if P is a path between x and y, e.g. P is either an (x, y)-path or a (y, x)-path. A **longest** path (cycle) in D is a path (cycle) of maximal length in D.

When W is a cycle and x is a vertex of W, we say that W is a cycle **through** x. In a directed pseudograph D, a loop is also considered a cycle (of length one). A $\boldsymbol{k}$**-cycle** is a cycle of length k. The minimum integer k for which D has a k-cycle is the **girth** of D; denoted by $g(D)$. If D does not have a cycle, we define $g(D) = \infty$. If $g(D)$ is finite then we call a cycle of length $g(D)$ a **shortest cycle** in D.

For subsets X, Y of $V(D)$, an (x, y)-path P is an $\boldsymbol{(X, Y)}$**-path** if $x \in X$, $y \in Y$ and $V(P) \cap (X \cup Y) = \{x, y\}$. Note that, if $X \cap Y \neq \emptyset$ then a vertex $x \in X \cap Y$ forms an (X, Y)-path by itself. Sometimes we will talk about an (H, H')-path when H and H' are subdigraphs of D. By this we mean an $(V(H), V(H'))$-path in D.

An (x_1, x_n)-path $P = x_1x_2\ldots x_n$ is **minimal** if, for every (x_1, x_n)-path Q, either $V(P) = V(Q)$ or Q has a vertex not in $V(P)$. For a cycle $C = x_1x_2\ldots x_px_1$, the subscripts are considered modulo p, i.e. $x_s = x_i$ for every s

and i such that $i \equiv s \bmod p$. As pointed out above (for trails), we will often view paths and cycles as subdigraphs. We can also consider paths and cycles as digraphs themselves. Let $\vec{P}_n$ ($\vec{C}_n$) denote a path (a cycle) with n vertices, i.e. $\vec{P}_n = (\{1, 2, \ldots, n\}, \{(1, 2), (2, 3), \ldots, (n-1, n)\})$ and $\vec{C}_n = \vec{P}_n + (n, 1)$.

A walk (path, cycle) W is a **Hamilton** (or **hamiltonian**) walk (path, cycle) if $V(W) = V(D)$. A digraph D is **hamiltonian** if D contains a Hamilton cycle; D is **traceable** if D possesses a Hamilton path. A trail $W = x_1x_2 \ldots x_k$ is an **Euler** (or **eulerian**) trail if $A(W) = A(D)$, $V(W) = V(D)$ and $x_1 = x_k$; a directed multigraph D is **eulerian** if it has an Euler trail.

To illustrate these definitions, consider Figure 1.9.

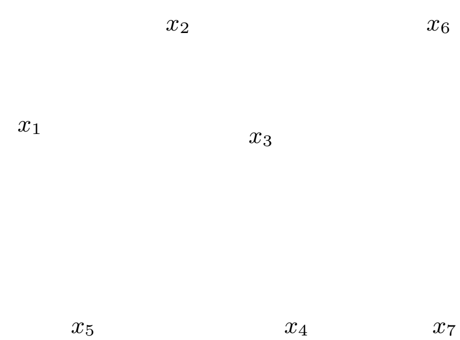


Figure 1.9 A directed graph H.

The walk $x_1x_2x_6x_3x_4x_6x_7x_4x_5x_1$ is a hamiltonian walk in D. The path $x_5x_1x_2x_3x_4x_6x_7$ is hamiltonian path in D. The path $x_1x_2x_3x_4x_6$ is an (x_1, x_6)-path and $x_2x_3x_4x_6x_3$ is an (x_2, x_3)-trail. The cycle $x_1x_2x_3x_4x_5x_1$ is a 5-cycle in D. The girth of D is 3 and the longest cycle in D has length 6.

Let $W = x_1x_2 \ldots x_k$, $Q = y_1y_2 \ldots y_t$ be a pair of walks in a digraph D. The walks W and Q are **disjoint** if $V(W) \cap V(Q) = \emptyset$ and **arc-disjoint** if $A(W) \cap A(Q) = \emptyset$. If W and Q are open walks, they are called **internally disjoint** if $\{x_2, x_3, \ldots, x_{k-1}\} \cap V(Q) = \emptyset$ and $V(W) \cap \{y_2, y_3, \ldots, y_{t-1}\} = \emptyset$.

We will use the following notation for a path or a cycle $W = x_1x_2 \ldots x_k$ (recall that $x_1 = x_k$ if W is a cycle):

$$W[x_i, x_j] = x_ix_{i+1} \ldots x_j.$$

It is easy to see that $W[x_i, x_j]$ is a path for $x_i \neq x_j$; we call it the **subpath** of W from x_i to x_j. If $1 < i \leq k$ then the **predecessor** of x_i on W is the vertex x_{i-1} and is also denoted by x_i^-. If $1 \leq i < k$, then the **successor** of x_i on W is the vertex x_{i+1} and is also denoted by x_i^+. Similarly, one can define $x_i^{++} = (x_i^+)^+$ and $x_i^{--} = (x_i^-)^-$, when these exist (which they always do if W is a cycle).

Also, for a set $X \subseteq V(W)$, we set $X^+ = \{x^+ : x \in X\}$, $X^- = \{x^- : x \in X\}$, $X^{++} = (X^+)^+$, etc. We will normally use such notation when a vertex x under consideration belongs to a unique walk W, otherwise W is given as a subscript, for example, x^+_W.

Proposition 1.4.1 *Let D be a digraph and let x, y be a pair of distinct vertices in D. If D has an (x, y)-walk W, then D contains an (x, y)-path P such that $A(P) \subseteq A(W)$. If D has a closed (x, x)-walk W, then D contains a cycle C through x such that $A(C) \subseteq A(W)$.*

Proof: Consider a walk P from x to y of minimum length among all (x, y)-walks whose arcs belong to $A(W)$. We show that P is a path. Let $P = x_1x_2 \ldots x_k$, where $x = x_1$ and $y = x_k$. If $x_i = x_j$ for some $1 \leq i < j \leq k$, then the walk $P[x_1, x_i]P[x_{j+1}, x_k]$ is shorter than P; a contradiction. Thus, all vertices of P are distinct, so P is a path with $A(P) \subseteq A(W)$.

Let $W = z_1z_2 \ldots z_k$ be a walk from $x = z_1$ to itself ($x = z_k$). Since D has no loop, $z_{k-1} \neq z_k$. Let $y_1y_2 \ldots y_t$ be a shortest walk from $y_1 = z_1$ to $y_t = z_{k-1}$. We have proved above that $y_1y_2 \ldots y_t$ is a path. Thus, $y_1y_2 \ldots y_ty_1$ is a cycle through $y_1 = x$. □

A digraph D is **acyclic** if it has no cycle. Acyclic digraphs form a well-studied family of digraphs, in particular, due to the following important properties.

Proposition 1.4.2 *Every acyclic digraph has a vertex of in-degree zero as well as a vertex of out-degree zero.*

Proof: Let D be a digraph in which all vertices have positive out-degrees. We show that D has a cycle. Choose a vertex v_1 in D. Since $d^+(v_1) > 0$, there is a vertex v_2 such that $v_1 \rightarrow v_2$. As $d^+(v_2) > 0$, v_2 dominates some vertex v_3. Proceeding in this manner, we obtain walks of the form $v_1v_2 \ldots v_k$. As $V(D)$ is finite, there exists the least $k > 2$ such that $v_k = v_i$ for some $1 \leq i < k$. Clearly, $v_iv_{i+1} \ldots v_k$ is a cycle.

Thus an acyclic digraph D has a vertex of out-degree zero. Since the converse H of D is also acyclic, H has a vertex v of out-degree zero. Clearly, the vertex v has in-degree zero in D. □

Proposition 1.4.2 allows one to check whether a digraph D is acyclic: if D has a vertex of out-degree zero, then delete this vertex from D and consider the resulting digraph; otherwise, D contains a cycle.

Let D be a digraph and let $x_1, x_2, \ldots, x_n$ be an ordering of its vertices. We call this ordering an **acyclic ordering** if, for every arc x_ix_j in D, we have $i < j$. Clearly, an acyclic ordering of D induces an acyclic ordering of every subdigraph H of D. Since no cycle has an acyclic ordering, no digraph with a cycle has an acyclic ordering. On the other hand, the following holds:

Proposition 1.4.3 *Every acyclic digraph has an acyclic ordering of its vertices.*

Proof: We give a constructive proof by describing a procedure that generates an acyclic ordering of the vertices in an acyclic digraph D. At the first step, we choose a vertex v with in-degree zero. (Such a vertex exists by Proposition 1.4.2.) Set $x_1 = v$ and delete x_1 from D. At the ith step, we find a vertex u of in-degree zero in the remaining acyclic digraph, set $x_i = u$ and delete x_i from the remaining acyclic digraph. The procedure has $|V(D)|$ steps.

Suppose that $x_i \rightarrow x_j$ in D, but $i > j$. As x_j was chosen before x_i, it means that x_j was not of in-degree zero at the jth step of the procedure; a contradiction. □

The notion of complexity of algorithms is discussed in Section 1.9. In Exercise 1.69, the reader is asked to show that the algorithm above can be performed in time $O(|V(D)| + |A(D)|)$.

Proposition 1.4.4 *Let D be an acyclic digraph with precisely one vertex x (y) of in-degree (out-degree) zero in D. For every vertex $v \in V(D)$ there is an (x, v)-path and a (v, y)-path in D.*

Proof: A longest path starting at v (terminating at v) is certainly a (v, y)-path (an (x, v)-path). □

An **oriented** graph is a digraph with no cycle of length two. A **tournament** is an oriented graph where every pair of distinct vertices are adjacent. In other words, a digraph T with vertex set $\{v_1, v_2, \dots, v_n\}$ is a tournament if exactly one of the arcs v_iv_j and v_jv_i is in T for every $i \neq j \in \{1, 2, \dots, n\}$. In Figure 1.10, one can see a pair of tournaments. It is an easy exercise to verify that each of them contains a Hamilton path. Actually, this is no coincidence by the following theorem of Rédei [625]. (In fact, Rédei proved a stronger result: every tournament contains an odd number of Hamilton paths.)

Figure 1.10 Tournaments.

Theorem 1.4.5 *Every tournament is traceable.*

Proof: Let T be a tournament with vertex set $\{v_1, v_2, \dots, v_n\}$. Suppose that the vertices of T are labeled in such a way that the number of backward arcs, i.e. arcs of the form v_jv_i, $j > i$, is minimum. Then, $v_1v_2 \dots v_n$ is a Hamilton path in T. Indeed, if this is not the case, there exists a subscript $i < n$ such that $v_i v_{i+1} \notin A(T)$. Thus, $v_{i+1}v_i \in A(T)$. However, in this case we can switch the vertices v_i and v_{i+1} in the labelling and decrease the number of backward arcs; a contradiction. $\square$

A ***q*-path-cycle subdigraph** $\mathcal{F}$ of a digraph D is a collection of q paths $P_1, \ldots, P_q$ and t cycles $C_1, \ldots, C_t$ such that all of $P_1, \ldots, P_q, C_1, \ldots, C_t$ are pairwise disjoint (possibly, $q = 0$ or $t = 0$). We will denote $\mathcal{F}$ by $\mathcal{F} = P_1 \cup \ldots \cup P_q \cup C_1 \cup \ldots \cup C_t$ (the paths always being listed first). Quite often, we will consider ***q*-path-cycle factors**, i.e. spanning q-path-cycle subdigraphs. If $t = 0$, $\mathcal{F}$ is a ***q*-path subdigraph** and it is a ***q*-path factor** (or just a **path-factor**) if it is spanning. If $q = 0$ we say that $\mathcal{F}$ is a ***t*-cycle subdigraph** (or just a **cycle subdigraph**) and it is a ***t*-cycle factor** (or just a **cycle factor**) if it is spanning. In Figure 1.11, $abc \cup defd$ is a 1-path-cycle factor, and $abcea \cup dfd$ is a cycle factor (or, more precisely, a 2-cycle factor).

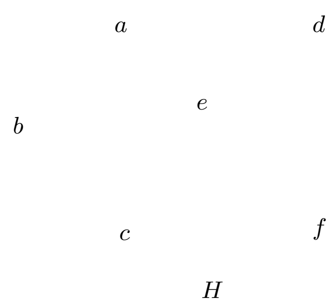


Figure 1.11 A digraph H.

The **path covering number** $\mathrm{pc}(D)$ of D is the minimum positive integer k such that D contains a k-path factor. In particular, $\mathrm{pc}(D) = 1$ if and only if D is traceable. The **path-cycle covering number** $\mathrm{pcc}(D)$ of D is the minimum positive integer k such that D contains a k-path-cycle factor. Clearly, $\mathrm{pcc}(D) \leq \mathrm{pc}(D)$. The proof of the following simple yet helpful assertion on the path covering number is left as an easy exercise to the reader (Exercise 1.34).

Proposition 1.4.6 *Let D be a digraph, and let k be a positive integer. Then the following statements are equivalent:*

(i) $\mathrm{pc}(D) = k$.
(ii) There are $k-1$ (new) arcs $e_1, \ldots, e_{k-1}$ such that $D + \{e_1, \ldots, e_{k-1}\}$ is traceable, but there is no set of $k-2$ arcs with this property.
(iii) $k-1$ is the minimum integer s such that addition of s new vertices to D together with all possible arcs between $V(D)$ and these new vertices results in a traceable digraph.

$\square$

1.5 Strong and Unilateral Connectivity

In a digraph D a vertex y is **reachable** from a vertex x if D has an (x, y)-walk. In particular, a vertex is reachable from itself. By Proposition 1.4.1, y is reachable from x if and only if D contains an (x, y)-path. A digraph D is **strongly connected** (or, just, **strong**) if, for every pair x, y of distinct vertices in D, there exists an (x, y)-walk and a (y, x)-walk. In other words, D is strong if every vertex of D is reachable from every other vertex of D. We define a digraph with one vertex to be strongly connected. It is easy to see that D is strong if and only if it has a closed Hamilton walk (Exercise 1.47). As $\vec{C}_n$ is strong, every hamiltonian digraph is strong. The following basic result on tournaments is due to Moon [570].

Theorem 1.5.1 (Moon's theorem) *[570] Let T be a strong tournament on $n \geq 3$ vertices. For every $x \in V(T)$ and every integer $k \in \{3, 4, \ldots, n\}$, there exists a k-cycle through x in T. In particular, a tournament is hamiltonian if and only if it is strong.*

Proof: Let x be a vertex in a strong tournament T on $n \geq 3$ vertices. The theorem is shown by induction on k. We first prove that T has a 3-cycle through x. Since T is strong, both $O = N^+(x)$ and $I = N^-(x)$ are non-empty. Moreover, (O, I) is non-empty; let $yz \in (O, I)$. Then, $xyzx$ is a 3-cycle through x. Let $C = x_0 x_1 \ldots x_t$ be a cycle in T with $x = x_0 = x_t$ and $t \in \{3, 4, \ldots, n-1\}$. We prove that T has a $(t+1)$-cycle through x.

If there is a vertex $y \in V(T) - V(C)$ which dominates a vertex in C and is dominated by a vertex in C, then it is easy to see that there exists an index i such that $x_i \rightarrow y$ and $y \rightarrow x_{i+1}$. Therefore, $C[x_0, x_i]yC[x_{i+1}, x_t]$ is a $(t+1)$-cycle through x. Thus, we may assume that every vertex outside of C either dominates every vertex in C or is dominated by every vertex in C. The vertices from $V(T) - V(C)$ that dominate all vertices from $V(C)$ form a set R; the rest of the vertices in $V(T) - V(C)$ form a set S. Since T is strong, both S and R are non-empty and the set (S, R) is non-empty. Hence taking $sr \in (S, R)$ we see that $x_0 srC[x_2, x_0]$ is a $(t+1)$-cycle through $x = x_0$. □

Corollary 1.5.2 (Camion's theorem) *[140] Every strong tournament is hamiltonian.* □

A digraph D is **complete** if, for every pair x, y of distinct vertices of D, both xy and yx are in D. For a strong digraph $D = (V, A)$, a set $S \subset V$ is a **separator** (or a **separating set**) if $D - S$ is not strong. A digraph D is ***k*-strongly connected** (or ***k*-strong**) if $|V| \geq k + 1$ and D has no separator with less than k vertices. It follows from the definition of strong connectivity that a complete digraph with n vertices is $(n-1)$-strong, but is not n-strong. The largest integer k such that D is k-strongly connected is the **vertex-strong connectivity** of D (denoted by $\kappa(D)$). If a digraph D is not strong, we set $\kappa(D) = 0$. For a pair s, t of distinct vertices of a digraph D, a set $S \subseteq V(D) - \{s, t\}$ is an **(s, t)-separator** if $D - S$ has no (s, t)-paths. For a strong digraph $D = (V, A)$, a set of arcs $W \subseteq A$ is a **cut** (or a **cut set**) if $D - A$ is not strong. A digraph D is **k-arc-strong** (or **k-arc-strongly connected**) if D has no cut with less than k arcs. The largest integer k such that D is k-arc-strongly connected is the **arc-strong connectivity** of D (denoted by $\lambda(D)$). If D is not strong, we set $\lambda(D) = 0$. Note that $\lambda(D) \geq k$ if and only if $d^+(X), d^-(X) \geq k$ for all proper subsets X of V.

A **strong component** of a digraph D is a maximal induced subdigraph of D which is strong. If $D_1, \ldots, D_t$ are the strong components of D, then clearly $V(D_1) \cup \ldots \cup V(D_t) = V(D)$ (recall that a digraph with only one vertex is strong). Moreover, we must have $V(D_i) \cap V(D_j) = \emptyset$ for every $i \neq j$ as otherwise all the vertices $V(D_i) \cup V(D_j)$ are reachable from each other, implying that the vertices of $V(D_i) \cup V(D_j)$ belong to the same strong component of D. We call $V(D_1) \cup \ldots \cup V(D_t)$ the **strong decomposition** of D. The **strong component digraph** $SC(D)$ of D is obtained by contracting strong components of D and deleting any parallel arcs obtained in this process. In other words, if $D_1, \ldots, D_t$ are the strong components of D, then $V(SC(D)) = \{v_1, v_2, \ldots, v_t\}$ and $A(SC(D)) = \{v_i v_j : \ (V(D_i), V(D_j))_D \neq \emptyset\}$. The subdigraph of D induced by the vertices of a cycle in D is strong, i.e. is contained in a strong component of D. Thus, $SC(D)$ is acyclic. By Proposition 1.4.3, the vertices of $SC(D)$ have an acyclic ordering. This implies that the strong components of D can be labeled $D_1, \ldots, D_t$ such that there is no arc from D_j to D_i unless $j < i$. We call such an ordering an **acyclic ordering** of the strong components of D. The strong components of D corresponding to the vertices of $SC(D)$ of in-degree (out-degree) zero are the **initial (terminal) strong components** of D. The remaining strong components of D are called **intermediate strong components** of D.Figure 1.12 shows a digraph D and its strong component digraph $SC(D)$.

It is easy to see that the strong component digraph of a tournament T is an acyclic tournament. Thus, there is a unique acyclic ordering of the strong components of T, namely, $T_1, \ldots, T_t$ such that $T_i \rightarrow T_j$ for every $i < j$. Clearly, every tournament has only one initial (terminal) strong component.

A digraph D is **unilateral** if, for every pair x, y of vertices of D, either x is reachable from y or y is reachable from x (or both). Clearly, every strong digraph is unilateral. A path $\vec{P}_n$ is unilateral; hence, being unilateral is a necessary condition for traceability of digraphs. The following is a characterization of unilateral digraphs.

Proposition 1.5.3 *A digraph D is unilateral if and only if there is a unique acyclic ordering $D_1, D_2, \ldots, D_t$ of the strong components of D and $(V(D_i), V(D_{i+1})) \neq \emptyset$ for every $i = 1, 2, \ldots, t-1$.*

Proof: The sufficiency is trivial. To see the necessity, observe that if $(V(D_i), V(D_{i+1})) = \emptyset$, then no vertex of $V(D_{i+1})$ is reachable from any vertex of $V(D_i)$ and vice versa. Finally, observe that, if $(V(D_i), V(D_{i+1})) \neq \emptyset$

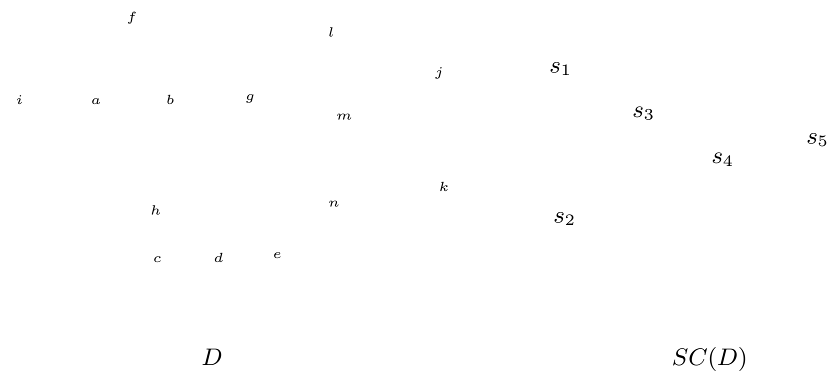


Figure 1.12 A digraph D and its strong component digraph $SC(D)$. The vertices s_1, s_2, s_3, s_4, s_5 are obtained by contracting the sets $\{a,b\}, \{c,d,e\}, \{f,g,h,i\}, \{j,k\}$ and $\{l,m,n\}$ which correspond to the strong components of D. The digraph D has two initial components, D_1, D_2 with $V(D_1) = \{a,b\}$ and $V(D_2) = \{c,d,e\}$. It has one terminal component D_5 with vertices $V(D_5) = \{l,m,n\}$ and two intermediate components D_3, D_4 with vertices $V(D_3) = \{f,g,h,i\}$ and $V(D_4) = \{j,k\}$.

for every $i = 1, 2, \ldots, t-1$, then $D_1, D_2, \ldots, D_t$ is the unique acyclic ordering of the strong components of D, because $SC(D)$ has a hamiltonian path (see Exercise 1.18). □

1.6 Undirected Graphs, Biorientations and Orientations

An **undirected graph** (or a **graph**) $G = (V, E)$ consists of a non-empty finite set $V = V(G)$ of elements called **vertices** and a finite set $E = E(G)$ of unordered pairs of distinct vertices called **edges**. We call $V(G)$ the **vertex set** and $E(G)$ the **edge set** of G. In other words, an edge $\{x,y\}$ is a 2-element subset of $V(G)$. We will often denote $\{x,y\}$ just by xy. If $xy \in E(G)$, we say that the vertices x and y are **adjacent**. Notice that, in the above definition of a graph, we do not allow loops (i.e. pairs consisting of the same vertex) or parallel edges (i.e. multiple pairs with the same end-vertices). The **complement** $\overline{G}$ of a graph G is the graph with vertex set $V(G)$ in which two vertices are adjacent if and only if they are not adjacent in G.

When parallel edges and loops are admissible we speak of **pseudographs**; pseudographs with no loops are **multigraphs**. For a pair u, v of vertices in a pseudograph G, $\mu_G(u,v)$ denotes the number of edges between u and v. In particular, $\mu_G(u,u)$ is the number of loops at u. For a pseudograph G, a directed pseudograph D is called a **biorientation** of G if D is obtained from G by replacing each edge $\{x,y\}$ of G by either xy or yx or the pair xy and yx (except for a loop xx which is replaced by a (directed) loop at x). Note that different copies of the edge xy in G may be replaced by different arcs in D. Thus if $\mu_G(x,y) = 3$ then we may replace one edge $\{x,y\}$ by the arc

xy, another by the arc yx and the third by the pair of arcs xy and yx. An **orientation** of a graph G is a biorientation of G which is an oriented graph (i.e. digraph having no 2-cycle and no loops). Clearly, every digraph is a biorientation and every oriented graph an orientation of some undirected graph. The **underlying graph** $UG(D)$ of a digraph D is the unique graph G such that D is a biorientation of G. For a graph G, the **complete biorientation** of G (denoted by $\overleftrightarrow{G}$) is a biorientation D of G such that $xy \in A(D)$ implies $yx \in A(D)$. A digraph $D = (V, A)$ is **symmetric** if $xy \in A$ implies $yx \in A$. Clearly, D is symmetric if and only if D is the complete biorientation of some graph. An **oriented path (cycle)** is an orientation of a path (cycle).

A pseudograph G is **connected** if its complete biorientation $\overleftrightarrow{G}$ is strongly connected. Similarly, G is ***k*-connected** if $\overleftrightarrow{G}$ is k-strong. Strong components in $\overleftrightarrow{G}$ are **connected components**, or just **components** in G. A **bridge** in a connected pseudograph G is an edge whose deletion from G makes G disconnected. A pseudograph G is ***k*-edge-connected** if the graph obtained from G after deletion of at most $k-1$ edges is connected. Clearly, a pseudograph is bridgeless if and only if it is 2-edge-connected. The **neighbourhood** $N_G(x)$ of a vertex x in G is the set of vertices adjacent to x. The **degree** $d(x)$ of a vertex x is the number of edges except loops having x as an end-vertex. The **minimum (maximum) degree** of G is

$$\delta(G) = \min\{d(x) : \ x \in V(G)\} \ (\Delta(G) = \max\{d(x) : \ x \in V(G)\}).$$

We say that G is **regular** (or **$\delta(G)$-regular**) if $\delta(G) = \Delta(G)$. A pair of graphs G and H is **isomorphic** if $\overleftrightarrow{G}$ and $\overleftrightarrow{H}$ are isomorphic.

A digraph is **connected** if its underlying graph is connected. The notions of walks, trails, paths and cycles in undirected pseudographs are analogous to those for directed pseudographs (we merely disregard orientations). An ***xy*-path** in an undirected pseudograph is a path whose end-vertices are x and y. When we consider a digraph and its underlying graph $UG(D)$, we will often call walks of D **directed** (to distinguish between them and those in $UG(D)$). In particular, we will speak of directed paths, cycles and trails. An undirected graph is a **forest** if it has no cycle. A connected forest is a **tree**. It is easy to see (Exercise 1.41) that every connected undirected graph has a **spanning tree**, i.e. a spanning subgraph, which is a tree. A digraph D is an **oriented forest** (**tree**) if D is an orientation of a forest (tree). A subgraph T of a (connected) digraph D is a **spanning oriented tree** of D if $UG(T)$ is a spanning tree in $UG(D)$. A subdigraph T of a digraph D is an **in-branching** (**out-branching**) if T is a spanning oriented tree of D and T has only one vertex s of out-degree (in-degree) zero. The vertex s is the **root** of T. (See Figure 1.13.) We will often use the notation F_s^+ (F_s^-) to denote an out-branching (in-branching) rooted at s in the digraph in question.

Since each spanning oriented tree R of a connected digraph is acyclic as an undirected graph, R has at least one vertex of out-degree zero and at

s

r

D H L

Figure 1.13 The digraph D has an out-branching with root r (shown in bold); H contains an in-branching with root s (shown in bold); L possesses neither an out-branching nor an in-branching.

least one vertex of in-degree zero (see Proposition 1.4.2). Hence, the out-branchings and in-branchings capture the important cases of uniqueness of the corresponding vertices. The following is a characterization of digraphs with in-branchings (out-branchings).

Proposition 1.6.1 *A connected digraph D contains an out-branching (in-branching) if and only if D has only one initial (terminal) strong component.*

Proof: We prove this characterization only for out-branchings since the second claim follows from the first one by considering the converse of D.

Assume that D contains at least two initial strong components, D_1 and D_2. Let T be an arbitrary spanning oriented tree in D. Then each of $T\langle D_1\rangle$ and $T\langle D_2\rangle$ contains a vertex of in-degree zero. These vertices are of in-degree zero in T as well because of the definition of initial strong components. Thus, T is not an out-branching and D has no out-branchings. Therefore, if D has an out-branching, D contains only one initial strong component.

Now we suppose that D contains only one initial strong component D_1, and r is an arbitrary vertex of D_1. We prove that D has an out-branching with root r. In $SC(D)$, the vertex x corresponding to D_1 is the only vertex of in-degree zero and, hence, by Proposition 1.4.4, every vertex of $SC(D)$ is reachable from x. Thus, every vertex of D is reachable from r. We construct an oriented tree T as follows. In the first step T consists of r. In Step $i \geq 2$, for every vertex y appended to T in the previous step, we add to T a vertex z, such that $y \rightarrow z$ and $z \notin V(T)$, together with the arc yz. We stop when no vertex can be included in T. Since every vertex of D is reachable from r, T is spanning. Clearly, r is the only vertex of in-degree zero in T. Hence, T is an out-branching. $\square$

The oriented tree T constructed in the proof of Proposition 1.6.1 is a so-called BFS tree of D (see Chapter 2).

The following well-known theorem is due to Robbins.

Theorem 1.6.2 (Robbins' theorem) *[637] A connected graph G has a strongly connected orientation if and only if G has no bridge.*

Proof: Clearly, if G has a bridge, G has no strong orientation. So assume that G is bridgeless. Then every edge uv is contained in some cycle (see Exercise 1.38). Choose a cycle C in G. Orient C as a directed cycle T_1. Suppose that $T_1, T_2, \ldots, T_k$ are chosen and oriented in such a way that every T_{i+1} $(1 \leq i < k)$ is either a cycle having only one vertex in common with $T^i = T_1 \cup T_2 \cup \ldots \cup T_i$ or a path with only initial and terminal vertices in common with T^i. If $UG(T^k) = G$, then we are done as a simple induction shows that T^k is strong. Otherwise, there is an edge xy which is not in $UG(T^k)$ such that x is in $UG(T^k)$. Let C be a cycle containing xy. Orient C to obtain a (directed) cycle Z. Let z be a vertex in $UG(T^k)$ which is first encountered while traversing Z (after leaving x). Then, set $T_{k+1} = Z[x, z]$. The path (or cycle) T_{k+1} satisfies the above-mentioned properties. Since $E(G)$ is finite, after a certain number of iterations $\ell \leq m - 1$ we have $UG(T^\ell) = G$. □

We formulate and prove the following well-known characterization of eulerian directed multigraphs (clearly, the deletion of loops in a directed pseudograph D does not change the property of D of being eulerian or otherwise). The 'undirected' version of this theorem marks the beginning of graph theory [225] (see the book [240] by Fleischner for a reprint of Euler's original paper and a translation into English, and see the book [119] by Biggs, Lloyd and Wilson or Wilson's paper [737] for a discussion of the historical record).

Theorem 1.6.3 (Euler's theorem[2]) *A directed multigraph D is eulerian if and only if D is connected and $d^+(x) = d^-(x)$ for every vertex x in D.*

Proof: Clearly, both conditions are necessary. We give a constructive proof of sufficiency by building an Euler trail T. Let T be initially empty. Choose an arbitrary vertex x in D. Since D is connected, there is a vertex $y \in N^+(x)$. Append x to T as well as an arc from x to y. Since $d^-(y) = d^+(y)$, there is an arc yz with tail y. Add both y and yz to T. We proceed similarly: after an arc uv is included in T, we append v to T together with an arc $a \notin T$ whose tail is v. Due to the condition $d^+(w) = d^-(w)$ for every vertex w, the above process can terminate only if the last arc appended to T is an arc whose head is the vertex x and the arcs of D with tail x are already in T. If all arcs of D are in T, we are done. Assume it is not so. Since D is connected, this means that T contains a vertex p which is a tail of an arc pq not in T. 'Shift' cyclically the vertices and arcs of T such that T starts and terminates at p. Add the arc pq to T and apply the process described above. This can terminate only when the last appended arc's tail is p and all arcs leaving p are already in T. Again, either we are done (all arcs are already in T) or we can find a new vertex to restart the above process. Since $V(D)$ is finite, after several steps all arcs of D will be included in T. □

[2] Euler's original paper [226] only dealt with undirected graphs, but it is easy to see that the directed case generalizes the undirected case (see also Exercise 1.44).

The algorithm described in this proof can be implemented to run in $O(|V(D)| + |A(D)|)$ time (see Exercise 1.72). A generalization of the last theorem is given in Theorem 11.1.2. For eulerian directed multigraphs, the following stronger condition on out-degrees and in-degrees holds.

Corollary 1.6.4 *Let D be an eulerian directed multigraph and let $\emptyset \neq W \subset V(D)$. Then, $d^+(W) = d^-(W)$.*

Proof: Observe that

$$\sum_{w \in W} d^+(w) = |(W,W)| + d^+(W), \quad \sum_{w \in W} d^-(w) = |(W,W)| + d^-(W). \quad (1.1)$$

By Theorem 1.6.3, $\sum_{w \in W} d^+(w) = \sum_{w \in W} d^-(w)$. The corollary follows from this equality and (1.1). □

A **matching** M in a directed (an undirected) pseudograph G is a set of arcs (edges) with no common end-vertices. We also require that no element of M is a loop. If M is a matching then we say that the edges (arcs) of M are **independent**. A matching M in G is **maximum** if M contains the maximum possible number of edges. A maximum matching is **perfect** if it has $n/2$ edges, where n is the order of G. A set Q of vertices in a directed or undirected pseudograph H is **independent** if the graph $H\langle Q\rangle$ has no edges (arcs). The **independence number**, $\alpha(H)$, of H is the maximum integer k such that H has an independent set of cardinality k. A **(proper) colouring** of a directed or undirected graph H is a partition of $V(H)$ into (disjoint) independent sets. The minimum number, $\chi(H)$, of independent sets in a proper colouring of H is the **chromatic number** of H.

In Section 1.3, the operation of composition of digraphs was introduced. Considering complete biorientations of undirected graphs, one can easily define the operation of **composition** of undirected graphs. Let H be a graph with vertex set $\{v_1, v_2, \ldots, v_n\}$, and let $G_1, G_2, \ldots, G_n$ be graphs which are pairwise vertex-disjoint. The composition $H[G_1, G_2, \ldots, G_n]$ is the graph L with vertex set $V(G_1) \cup V(G_2) \cup \ldots \cup V(G_n)$ and edge set

$$\cup_{i=1}^n E(G_i) \cup \{g_i g_j : \; g_i \in V(G_i), g_j \in V(G_j), v_i v_j \in E(H)\}.$$

If none of the graphs $G_1, \ldots, G_n$ in this definition of $H[G_1, \ldots, G_n]$ have edges, then $H[G_1, \ldots, G_n]$ is an **extension** of H.

1.7 Mixed Graphs and Hypergraphs

Mixed graphs are useful by themselves as a common generalization of undirected and directed graphs. Moreover, mixed graphs are helpful in several proofs on biorientations of graphs.

A **mixed graph** $M = (V, A, E)$ contains both arcs (ordered pairs of vertices in A) and edges (unordered pairs of vertices in E). We do not allow loops or parallel arcs and edges, but M may have an edge and an arc with the same end-vertices. For simplicity, both edges and arcs of a mixed graph are called **edges**. Thus, an arc is viewed as an oriented edge (and an unoriented edge as an edge in the usual sense). A **biorientation** of a mixed graph $M = (V, A, E)$ is obtained from M by replacing every unoriented edge xy of E by the arc xy, the arc yx or the pair xy, yx of arcs. If no unoriented edge is replaced by a pair of arcs, we speak of an **orientation** of a mixed graph[3]. The **complete biorientation** of a mixed graph $M = (V, A, E)$ is a biorientation $\overleftrightarrow{M}$ of M such that every unoriented edge $xy \in E$ is replaced in $\overleftrightarrow{M}$ by the pair xy, yx of arcs. Using the complete biorientation of a mixed graph M, one can easily give the definitions of a walk, trail, path, and cycle in M. The only extra condition is that every pair of arcs in $\overleftrightarrow{M}$ obtained in replacement of an edge in M has to be treated as two appearances of one thing. For example, if one of the arcs in such a pair appears in a trail T, then the second one cannot be in T. A mixed graph M is **strong** if $\overleftrightarrow{M}$ is strong. Similarly, one can give the definition of strong components. A mixed graph M is **connected** if $\overleftrightarrow{M}$ is connected. An edge ℓ in a connected mixed graph M is a **bridge** if $M - \ell$ is not connected.

Figure 1.14 illustrates the notion of a mixed graph. The mixed graph M depicted in Figure 1.14 is strong; $u, (u, v), v, \{v, u\}, u$ is a cycle in M; $M - x$ has two strong components: one consists of the vertex y, the other is $M' = M\langle\{u, v, w\}\rangle$; the edge $\{v, w\}$ is a bridge in M', the arc (u, v) and the edge $\{u, v\}$ are not bridges in M'; M is bridgeless.

v w

u x

y

Figure 1.14 A mixed graph.

Theorem 1.7.1 below is due to Boesch and Tindell [120]. This result is an extension of Theorem 1.6.2. We give a short proof obtained by Volkmann

[3] Note that a mixed graph $M = (V, A, E)$ may have a directed 2-cycle in which case no orientation of M is an oriented graph (because some 2-cycles remain).

[730]. (Another proof which leads to a linear time algorithm is obtained by Chung, Garey and Tarjan [157].)

Theorem 1.7.1 *Let e be an unoriented edge in a strong mixed graph M. The edge e can be replaced by an arc (with the same end-vertices) such that the resulting mixed graph M' is strong if and only if e is not a bridge.*

Proof: If e is a bridge, then clearly there is no orientation of e that results in a strong mixed graph. Assume that e is not a bridge. Let $M' = M - e$. If M' is strong, then any orientation of e leads to a strong mixed graph; thus, assume that M' is not strong. Since e is not a bridge, M' is connected. Let $L_1, L_2, \ldots, L_k$ be strong components of M'. Since M is strong, there is only one initial strong component, say L_1, and only one terminal strong component, say L_k. Let u (v) be the end-vertex of e belonging to L_1 (L_k). By strong connectivity of $L_1, L_2, \ldots, L_k$ and Proposition 1.4.4 (applied to the strong component digraph of $\overleftrightarrow{M'}$), $M' + (v, u)$ is strong. □

An **orientation** of a digraph D is a subdigraph of D obtained from D by deleting exactly one arc between x and y for every pair $x \neq y$ of vertices such that both xy and yx are in D. See Figure 1.15 for an illustration of this definition.

D H H' H''

Figure 1.15 A digraph D and subdigraphs H, H' and H'' of D. The digraph H is an orientation of D but neither of H', H'' is an orientation of D.

Since we may transform a digraph to a mixed graph by replacing every 2-cycle with an undirected edge, we obtain the following reformulation of Theorem 1.7.1.

Corollary 1.7.2 *A strong digraph D has a strong orientation if and only if $UG(D)$ has no bridge.* □

A **hypergraph** is an ordered set $H = (V, \mathcal{E})$ such that V is a set (of **vertices** of H) and $\mathcal{E}$ is a family of subsets of V (called **edges** of H).

The **rank** of H is the cardinality of the largest edge of H. For example, $H_0 = (\{1,2,3,4\}, \{\{1,2,3\}, \{2,3\}, \{1,2,4\}\}$ is a hypergraph. The rank of H_0 is three. The number of vertices in H is its **order**. We say that H is **2-colourable** if there is a function $f: V \rightarrow \{0,1\}$ such that, for every edge $E \in \mathcal{E}$, there exist a pair of vertices $x, y \in E$ such that $f(x) \neq f(y)$. The function f is called a **2-colouring** of H. It is easy to verify that H_0 is 2-colourable. In particular, $f(1) = f(2) = 0,\ f(3) = f(4) = 1$ is a 2-colouring of H_0. A hypergraph is **uniform** if all its edges have the same cardinality. Thus an undirected graph is a 2-uniform hypergraph.

1.8 Classes of Directed and Undirected Graphs

In this section, we define certain families of directed and undirected multigraphs which will be used in various chapters of this book.

A multigraph G is **complete** if every pair of distinct vertices in G are adjacent. We will denote the complete graph on n vertices (which is unique up to isomorphism) by K_n. Its complement $\overline{K}_n$ has no edge.

A multigraph H is ***p*-partite** if there exists a partition $V_1, V_2, \ldots, V_p$ of $V(H)$ into p **partite sets** (i.e., $V(H) = V_1 \cup \ldots \cup V_p$, $V_i \cap V_j = \emptyset$ for every $i \neq j$) such that every edge of H has its end-vertices in different partite sets. The special case of a p-partite graph when $p = 2$ is called a **bipartite graph**. We often denote a bipartite graph B by $B = (V_1, V_2; E)$. A p-partite multigraph H is **complete *p*-partite** if, for every pair $x \in V_i$, $y \in V_j$ $(i \neq j)$, an edge xy is in H. A complete graph on n vertices is clearly a complete n-partite graph for which every partite set is a singleton. We denote the complete p-partite *graph* with partite sets of cardinalities $n_1, n_2, \ldots, n_p$ by $K_{n_1,n_2,\ldots,n_p}$. Complete p-partite graphs for $p \geq 2$ are also called **complete multipartite graphs**.

To obtain short proofs of various results on subdigraphs of a directed multigraph $D = (V, A)$ the following transformation to the class of bipartite (undirected) multigraphs is extremely useful. Let $BG(D) = (V', V''; E)$ denote the bipartite multigraph with partite sets $V' = \{v' :\ v \in V\},\ V'' = \{v'' :\ v \in V\}$ such that $\mu_{BG(D)}(u'w'') = \mu_D(uw)$ for every pair u, w of vertices in D. We call $BG(D)$ the **bipartite representation** of D; see Figure 1.16.

A ***p*-partite digraph** is a biorientation of a p-partite graph; see Figure 1.17 (b). Bipartite (i.e. 2-partite) digraphs are of special interest. It is well-known (and was proved already by König [497]) that an undirected graph is bipartite if and only if it has no cycle of odd length. The obvious extension of this statement to cycles in digraphs is not valid (the non-bipartite digraph with vertex set $\{x, y, z\}$ and arc set $\{xy, xz, yz\}$ is such an example that can easily be generalized). However, the obvious extension does hold for strong bipartite digraphs. Theorem 1.8.1 can be traced back to the book [404] by Harary, Norman and Cartwright.

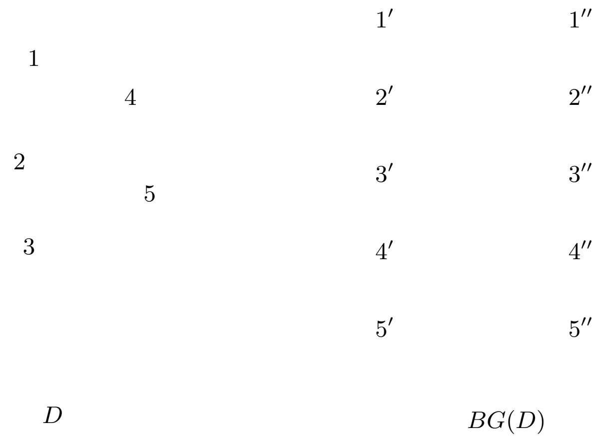


Figure 1.16 A directed multigraph and its bipartite representation.

Theorem 1.8.1 *A strongly connected digraph is bipartite if and only if it has no cycle of odd length.*

Proof: If D is bipartite, then it is easy to see that D cannot have an odd cycle. To prove sufficiency suppose that D has no odd cycle. Fix an arbitrary vertex x in D. We claim that for every vertex $y \in V(D) - x$ and every choice of an (x, y)-path P and a (y, x)-path Q, the length of P and Q are equal modulo 2.

Suppose this is not the case for some choice of y, P and Q. Then choose y, P and Q such that the parity of the lengths of P and Q differ and $|V(P)| + |V(Q)|$ is minimum among all such pairs of (x, y)- and (y, x)-paths whose lengths differ in parity. If $V(P) \cap V(Q) = \{x, y\}$, then PQ is an odd cycle, contradicting the assumption. Hence there is a vertex $z \notin \{x, y\}$ in $V(P) \cap V(Q)$. Let z be chosen as the first such vertex that we meet when we traverse Q from y towards x. Then $P[z, y]Q[y^+_Q, z]$ is a cycle and it is even by our assumption. But now it is easy to see that the parity of the paths $P[x, z]$ and $Q[z, x]$ are different and we get a contradiction to the choice of y, P and Q. This proves the claim and, in particular, it follows that for every $y \in V(D) - x$, the lengths of all paths from x to y have the same parity.

Now let

$$U = \{y : \text{ the length of every } (x, y)\text{-path is even}\},$$

$$U' = \{y : \text{ the length of every } (x, y)\text{-path is odd}\}.$$

This is a bipartition of $V(D)$ and neither U nor U' contains two vertices which are joined by an arc, since that would imply that some vertex had paths of two different parities from x. $\square$

An extension of this theorem to digraphs whose cycles are all of length 0 modulo $k \geq 2$ is given in Theorem 10.5.1.

Recall that tournaments are orientations of complete graphs. A **semicomplete digraph** is a biorientation of a complete graph (see Figure 1.17(a)). The complete biorientation of a complete graph is a **complete digraph** (denoted by $\overset{\leftrightarrow}{K}_n$). The notion of semicomplete digraphs and their special subclass, tournaments, can be extended in various ways. A biorientation of a complete p-partite (multipartite) graph is a **semicomplete *p*-partite (multipartite) digraph**; see Figure 1.17(c). A **multipartite tournament** is an orientation of a complete multipartite graph. A semicomplete 2-partite digraph is also called a **semicomplete bipartite digraph**. A **bipartite tournament** is a semicomplete bipartite digraph with no 2-cycles.

(a) K_4 and a semicomplete digraph of order four.

(b) A 3-partite graph G and a biorientation of G.

(c) The complete 3-partite graph $K_{2,1,2}$ and a semicomplete 3-partite digraph D.

Figure 1.17 Classes of graphs and digraphs.

One can use the operation of extension introduced in Section 1.3 to define 'extensions' of the above classes of digraphs. We will speak of **extended semicomplete digraphs** (i.e., extensions of semicomplete digraphs), **extended semicomplete multipartite digraphs**, etc. Clearly, every extended semicomplete digraph is a semicomplete multipartite digraph, which is not necessarily semicomplete, and every extended semicomplete multipartite digraph is a semicomplete multipartite digraph. Therefore, the class of semicomplete multipartite digraphs is **extension-closed**, and the class of semicomplete digraphs is not.

Recall that a digraph D is acyclic if D has no cycle. Obviously, every acyclic digraph is an oriented graph. A digraph D is **transitive** if, for every pair of arcs xy and yz in D such that $x \neq z$, the arc xz is also in D. It is easy to show that a tournament is transitive if and only it is acyclic (see Exercise 1.46). Sometimes, we will deal with transitive oriented graphs, i.e. transitive digraphs with no cycle of length two. A digraph D is **quasi-transitive** if, for every triple x, y, z of distinct vertices of D such that xy and yz are arcs of D, there is at least one arc between x and z. Clearly, a semicomplete digraph is quasi-transitive. Note that, if there is only one arc between x and z, it can have any direction; hence quasi-transitive digraphs are generally not transitive.

T $\qquad\qquad\qquad$ Q

Figure 1.18 A transitive digraph T and a quasi-transitive digraph Q.

1.9 Algorithmic Aspects

In this book we will often describe and analyze algorithms on digraphs. We will concentrate more on graph-theoretical aspects of these algorithms than on their actual implementation on a computer. (In particular, we will sometimes not prove the best possible complexity of an algorithm. However, in most such cases, we will provide a reference to a better complexity.) Still some very basic notions related to data structures and algorithms are required and will be given below. For more details on design and analysis of combinatorial algorithms, the reader is addressed to numerous books on the subject, e.g., to Aho, Hopcroft and Ullman [6], Brassard and Bratley [134] and Cormen, Leiserson and Rivest [169].

1.9.1 Algorithms and their Complexity

Recall that unless specified otherwise n (m) denotes the number of vertices (arcs) in the directed multigraph under consideration. In the following, all logarithms whose base is unspecified are of base 2. For a pair of given functions $f(k), g(k)$ of a non-negative integer argument k, we say that $f(k) = O(g(k))$ if there exist positive constants c and k_0 such that $0 \leq f(k) \leq cg(k)$ for all $k \geq k_0$. If there exist positive constants c and k_0 such that $0 \leq cf(k) \leq g(k)$ for all $k \geq k_0$, we say that $g(k) = \Omega(f(k))$. Clearly, $f(k) = O(g(k))$ if and only if $g(k) = \Omega(f(k))$. If both $f(k) = O(g(k))$ and $f(k) = \Omega(g(k))$ hold, then we say that $f(k)$ and $g(k)$ are of the **same order** and denote it by $f(k) = \Theta(g(k))$.

In the analysis of an algorithm, first of all we will be interested in its **time complexity** which must reflect the running time of the corresponding computer program on various computers. In order to make the time complexity measure sufficiently universal, it is usually assumed that computations are performed by some abstract computer. The computer executes **elementary operations**, that is, arithmetical operations, comparisons, data movements and control branching, each in constant time. Since we are interested only in the asymptotics of the execution time, the number of elementary operations of an algorithm will be considered as its time complexity. In the vast majority of cases, the time complexity (which will often be called just the **complexity**) of an algorithm depends on the size of its input. An algorithm $\mathcal{A}$ is an $O(g(n))$ algorithm for some function $g(n)$ of its input size if the running time of $\mathcal{A}$ on inputs of size n never exceeds $cg(n)$ for some constant c (depending only on $\mathcal{A}$).

Since the typical inputs to the algorithms considered in this book are (weighted) directed multigraphs, the size of inputs will be measured by the numbers of vertices and arcs, that is, by n and m, and, for digraphs with weights on the arcs (vertices), by $\log |c_{\max}|$, where $|c_{\max}|$ is the maximum of the absolute values of the weights of arcs (vertices). An algorithm of complexity $O(p(n, m, \log |c_{\max}|))$, where $p(n, m, \log |c_{\max}|)$ is a polynomial in n, m and $\log |c_{\max}|$, is a **polynomial-time** (or just **polynomial**) algorithm. The notion of equating efficient algorithms with polynomial algorithms is due to Edmonds [210] and is at present the most popular formalization for the intuitive notion of 'efficient' algorithms. Although we would normally not call an algorithm of complexity $\Theta(n^{1000})$, where n is the size of the input, an efficient algorithm, it is very rarely the case that polynomial algorithms have such a high degree of their associated polynomials.

There are two well-known and often-used ways to represent a digraph $D = (V, A)$ for computational purposes: as a collection of adjacency lists and as an adjacency matrix.

For the **adjacency matrix representation** of a directed multigraph $D = (V, A)$, we assume that the vertices of D are labeled $v_1, v_2, \dots, v_n$ in some arbitrary but fixed manner. The **adjacency matrix** $M(D) = [m_{ij}]$ of a digraph D is an $n \times n$-matrix such that $m_{ij} = 1$ if $v_i \rightarrow v_j$ and $m_{ij} = 0$ otherwise. For directed pseudographs we let $m_{ij} = \mu(v_i, v_j)$, that is, m_{ij} is the number of arcs from v_i to v_j. The adjacency matrix representation is a very convenient and fast tool for checking whether there is an arc from a vertex to another one. A drawback of this representation is the fact that to check all adjacencies, without using any other information besides the adjacency matrix, one needs $\Omega(n^2)$ time. Thus, the majority of algorithms using the adjacency matrix cannot have complexity lower than $\Omega(n^2)$ (this holds in particular if we include the time needed to construct the adjacency matrix).

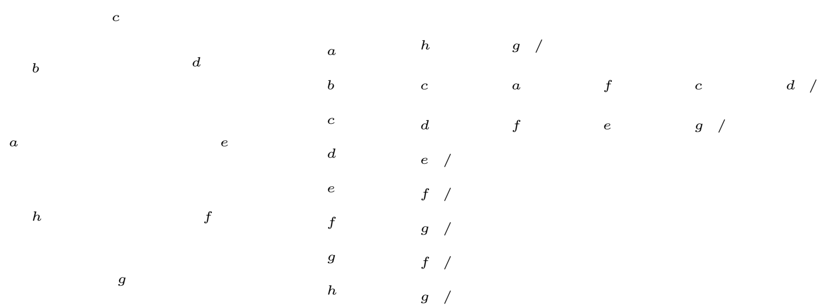


Figure 1.19 A directed multigraph and a representation by adjacency lists Adj^+.

The **adjacency list representation** of a directed pseudograph $D = (V, A)$ consists of a pair of arrays Adj^+ and Adj^-. Each of Adj^+ and Adj^- consists of $|V|$ (linked) lists, one for every vertex in V. For each $x \in V$, the linked list $Adj^+(x)$ ($Adj^-(x)$, respectively) contains all vertices dominated by x (dominating x, respectively) in some fixed order (see Figure 1.19). Using the adjacency list $Adj^+(x)$ ($Adj^-(x)$) one can obtain all out-neighbours (in-neighbours) of a vertex x in $O(|Adj^+(x)|)$ ($O(|Adj^-(x)|)$) time. A drawback of the adjacency list representation is the fact that one needs, in general, more than constant time to verify whether $x \rightarrow y$. Indeed, to decide this we have to search sequentially through $Adj^+(x)$ (or $Adj^-(x)$) until we either find y (x) or reach the end of the list.

To illustrate the concepts described in this section, let us consider the Hamilton path problem in tournaments. Theorem 1.4.5 states that every tournament is traceable. However, the proof that we have presented is non-constructive, i.e. it does not provide us with a polynomial algorithm to find a Hamilton path in a tournament. Now we give two constructive proofs of Theorem 1.4.5 and show how these lead to polynomial algorithms to construct a Hamilton path in a tournament.

Inductive Proof of Theorem 1.4.5: Clearly, the one vertex tournament has a Hamilton path (the vertex itself). Assume that the theorem holds for every tournament with less that $n(\geq 2)$ vertices. Consider a tournament T with n vertices and a vertex $x \in V(T)$. By induction, the tournament $T - x$ has a Hamilton path, $P = y_1y_2 \ldots y_{n-1}$. If $x \rightarrow y_1$, then xP is a Hamilton path in T; if $y_{n-1} \rightarrow x$, then Px is a Hamilton path in T. Assume that $y_1 \rightarrow x$ and $x \rightarrow y_{n-1}$. Then, it is easy to show that there exists an index $i < n-1$ such that $y_i \rightarrow x$ and $x \rightarrow y_{i+1}$. Thus, $P[y_1, y_i]xP[y_{i+1}, y_{n-1}]$ is a Hamilton path in T. □

This constructive proof gives rise to the following simple algorithm to find a Hamilton path in a tournament. One of the reasons for the simplicity of this algorithm is that it is recursive (for a discussion of recursive algorithms, see e.g. the book [169] by Cormen, Leiserson and Rivest).

HamPathTour:
Input: A tournament T on n vertices labelled $x_1, x_2, \ldots, x_n$ and its adjacency matrix $M = [m_{ij}]$.
Output: A Hamilton path in T.

1. Let $P := x_1$ and $i := 2$.
2. If $i > n$ go to Step 7.
3. Let $P = y_1y_2 \ldots y_{i-1}$ be the current path.
4. If $x_i \rightarrow y_1$ then $P := x_iP$. Let $i := i + 1$ and go to Step 2.
5. If $y_{i-1} \rightarrow x_i$ then $P := Px_i$. Let $i := i + 1$ and go to Step 2.
6. For $j = 1$ to $i - 2$ do: If $y_j \rightarrow x_i \rightarrow y_{j+1}$ then $P := P[y_1, y_j]x_iP[y_{j+1}, y_{i-1}]$. Let $i := i + 1$ and go to Step 2.
7. Return the path P.

The correctness of this algorithm follows from the above proof. To see that this algorithm can be implemented as an $O(n^2)$ algorithm, observe that the algorithm has two nested loops, each of which perform $O(n)$ operations (we count queries to the adjacency matrix as constant time) and all other steps take constant time. Thus, the complexity is $O(n^2)$.

The reader who is familiar with algorithms for sorting numbers might have noticed that HamPathTour is very similar to the algorithm *Insertion-Sort* which sorts numbers by inserting one at a time in a list (see e.g. [169, pp. 2-4]). This resemblance is no coincidence. In fact, given any set $\mathcal{S} = \{a_1, \ldots, a_n\}$ of n distinct real numbers we can form an acyclic tournament $T(\mathcal{S})$ with $V(T(\mathcal{S})) = \mathcal{S}$ and $A(T(\mathcal{S})) = \{a_ia_j : a_i < a_j, 1 \leq i \neq j \leq n\}$. The correct (sorted) increasing order on $\mathcal{S}$ corresponds to the unique Hamilton path $a_{\pi(1)}a_{\pi(2)} \ldots a_{\pi(n)}$ of $T(\mathcal{S})$ which again is the unique acyclic ordering of $V(T(\mathcal{S}))$ (see also Exercise 1.18). Thus any algorithm for finding a Hamilton path in a tournament can be used for sorting numbers (we compare numbers, by looking at the orientation of the arc between the corresponding vertices in[4] $T(\mathcal{S})$). Conversely, several sorting algorithms can be translated into algorithms for solving the more general problem of finding Hamilton paths in tournaments. One such example is the classical **Mergesort** algorithm (see e.g.[169, pp. 12-15]), which we now translate into the language of tournaments. For simplicity we shall assume that the number of vertices of the input tournament is a power of two. The reader can easily extend the algorithm to the general case, see Exercise 1.70. It is convenient to state the algorithm as a recursive algorithm (which is the reason why we specify a parameter for the algorithm). We assume that the tournament is available through its adjacency matrix.

MergeHamPathTour(T):

1. Split T into two tournaments T_1 and T_2 on the same number of vertices.
2. P_i:= **MergeHamPathTour**(T_i), $i = 1, 2$.
3. P:= **MergePaths**(P_1, P_2).
4. Return P.

Here **MergePaths** is a procedure, which given two disjoint paths P, P' in tournament T merges these two into one path P^* such that $V(P^*) = V(P) \cup V(P')$. This can be done in the same way as one would merge two sorted lists of numbers into one sorted list.

Procedure **MergePaths**(P, P'):
Input: Paths $P = x_1x_2 \ldots x_k$ and $P' = y_1y_2 \ldots y_r$.
Output: A path P^* such that $V(P^*) = V(P) \cup V(P')$.

1. If P' is empty then P^*:=P.
2. If P is empty then P^*:= P'.
3. If x_1 dominates y_1 then P^*:=x_1**MergePaths**($P - x_1$,P').
4. If y_1 dominates x_1 then P^*:=y_1**MergePaths**(P, $P' - y_1$).
5. Return P^*.

The classical analysis of the MergeSort algorithm (see e.g. [169]) shows that the algorithm uses $O(n \log n)$ comparisons to sort n real numbers. Similarly it follows from our description above that the algorithm MergeHamPathTour will find a Hamilton path in a tournament T with n vertices after making $O(n \log n)$ queries about adjacencies of vertices in T. Note that to implement the algorithm we do not need to construct the adjacency matrices of each of the tournaments considered in the recursive calls. Indeed, all adjacencies can be checked using the adjacency matrix of the original tournament. Hence, if we only count the number of times we need to check the direction of an arc, then MergeHamPathTour is a faster algorithm than HamPathTour.

[4] Note that this is only a virtual description, since we do not need to construct the adjacency matrix in this case. We simply compare the two numbers x and y and $x \rightarrow y$ holds if and only if $x < y$.

1.9.2 $\mathcal{NP}$-Complete and $\mathcal{NP}$-Hard Problems

There are many interesting algorithmic problems concerning (di)graphs for which no polynomial algorithm is known. Many of those problems (formulated in their decision form) belong to the class $\mathcal{NPC}$ of so-called **$\mathcal{NP}$-complete** problems. For a detailed introduction to the class of $\mathcal{NP}$-complete problems, see the book by Garey and Johnson [303]. A problem is a **decision** problem if it requires the answer 'yes' or 'no'. By a **problem** we understand actually a family of instances. For example, we will consider the **Hamilton cycle problem** in a digraph: given a digraph, decide whether or not it has a Hamilton cycle. Every digraph provides an **instance** of this problem. The so-called **travelling salesman problem** (TSP) is similar: given a weighted complete digraph D and a real number B, decide whether D contains a Hamilton cycle of weight at most B. An instance of the last problem consists of a complete digraph and a specification of the weights of its arcs.

A decision problem $\mathcal{S}$ belongs to the complexity class $\mathcal{P}$ if and only if there exists a polynomial algorithm $\mathcal{A}$ which, given any instance of $\mathcal{S}$, produces an answer in the set {'yes','no'} such that the answer from $\mathcal{A}$ on input x is 'yes' if and only if x is a 'yes' instance for[5] $\mathcal{S}$. Since $\mathcal{A}$ is polynomial, it follows that it produces its answer after at most $p(|x|)$ steps, where $|x|$ is the size of the input x and p is a fixed polynomial (depending on $\mathcal{S}$).

A decision problem belongs to the class $\mathcal{NP}$ (co-$\mathcal{NP}$) if, for every 'yes'-instance ('no'-instance) of the problem, there exists a short 'proof', called a **certificate**, of polynomial size (in n, m and $\log |c_{max}|$) such that, using the certificate, one can verify in polynomial time that the instance is indeed a 'yes' ('no') instance. The certificate depends on the instance of the problem, but it must have the same structure for all instances of the problem. To illustrate this definition, let us show that both the Hamilton cycle problem and travelling salesman problem are in $\mathcal{NP}$. Given a permutation π of the vertices in a digraph D (π is the certificate for hamiltonicity of D), it is easy to verify whether this permutation corresponds to a Hamilton cycle in D (note that this certificate has the same structure for each instance of the problem, namely it is a permutation of the vertices). Indeed, assuming that $V(D) = \{1, 2, \ldots, n\}$, we simply have to check that $\pi(i)\pi(i+1)$ is an arc of D for every $i = 1, 2, \ldots, n$, where the vertex $n+1$ is the same as the vertex 1. If we also have weights on the arcs, then it is also easy to verify that the weight of the proposed Hamilton cycle is no more than B. Notice that the situation here is not symmetric: it is unknown if the 'complement' of the Hamilton cycle problem (given a digraph, check whether it has no Hamilton cycle) is in $\mathcal{NP}$. Indeed, it is difficult to imagine what kind of certificate will enable a polynomial algorithm to check that a digraph is not hamiltonian. Actually, such a certificate would answer in affirmative the well-known complexity question: whether $\mathcal{NP}$ =co-$\mathcal{NP}$ (see e.g. [303, Theorem

[5] Thus a hypothetical polynomial algorithm for the Hamilton cycle problem must produce the answer 'yes' precisely when the input digraph has a Hamilton cycle.

7.2]). A positive answer to this question seems to be unlikely with our current knowledge of algorithms.

Given a pair of decision problems $\mathcal{S}$, $\mathcal{T}$, we say that $\mathcal{S}$ is **polynomially reducible** to $\mathcal{T}$ (denoted $\mathcal{S} \leq_{\mathcal{P}} \mathcal{T}$) if there is a polynomial algorithm $\mathcal{A}$ that transforms an instance x of $\mathcal{S}$ into an instance $\mathcal{A}(x)$ of $\mathcal{T}$ such that the second instance has the same answer as the first one. That is, x is a 'yes' instance of $\mathcal{S}$ if and only if $\mathcal{A}(x)$ is a 'yes' instance of $\mathcal{T}$. Some polynomial reductions are quite easy. For example, we can readily reduce the Hamilton cycle problem to the travelling salesman problem: given a digraph D consider a copy of a $\overleftrightarrow{K}_n$ such that $V(D) = V(\overleftrightarrow{K}_n)$, and, for every arc e in $\overleftrightarrow{K}_n$, its weight is 1 if $e \in A(D)$ and 2 otherwise. Let also $B = n$. Clearly, D is hamiltonian if and only if with the prescribed weights $\overleftrightarrow{K}_n$ has a Hamilton cycle of weight not exceeding B. Obviously, the above transformation can be carried out by a polynomial algorithm.

A decision problem is ***$\mathcal{NP}$*-hard** if all problems in $\mathcal{NP}$ can be polynomially reduced to this problem. If the problem is $\mathcal{NP}$-hard and also belongs to $\mathcal{NP}$, then it is ***$\mathcal{NP}$*-complete**. The class $\mathcal{NPC}$ consists of all $\mathcal{NP}$-complete problems. In order to show that a decision problem $\mathcal{W}$ is $\mathcal{NP}$-hard, we must show that every problem in $\mathcal{NP}$ can be polynomially reduced to $\mathcal{W}$ – a seemingly impossible task. However, polynomial transformations are closed under composition, that is, $\mathcal{S} \leq_{\mathcal{P}} \mathcal{T}$ and $\mathcal{T} \leq_{\mathcal{P}} \mathcal{K}$ implies that $\mathcal{S} \leq_{\mathcal{P}} \mathcal{K}$ (see Exercise 1.73). Hence, in order to prove that W is $\mathcal{NP}$-hard, it suffices to prove that there is *some* $\mathcal{NP}$-complete problem which is polynomially reducible to $\mathcal{W}$ (see Exercise 1.75). Of course this only works if we already have established that there is some problem that belongs to the class $\mathcal{NPC}$ of $\mathcal{NP}$-complete problems. This extremely important and non-trivial step was provided by Cook in 1971 [165] (independently, a similar discovery was made by Levin [513]).

Since there are a huge number of known $\mathcal{NP}$-complete problems, the task to prove that a given problem is $\mathcal{NP}$-complete is sometimes not too difficult. On the other hand, it is also highly non-trivial in many cases. We will give a number of examples of $\mathcal{NP}$-completeness and $\mathcal{NP}$-hardness proofs throughout this book. It is well-known that the Hamilton cycle problem is $\mathcal{NP}$-complete as shown by Karp in his classical paper [474]. ¿From the above transformation, it follows that the travelling salesman problem is $\mathcal{NP}$-complete as well.

Quite often we will deal with **optimization problems** rather than decision problems. Since an optimization problem consists of finding an optimal solution to a prescribed problem, such a problem very often has a decision analogue. For example, in the usual formulation of the travelling salesman problem the goal is to find a minimum weight Hamilton cycle in a weighted complete digraph. The decision analogue was stated above. If the decision analogue of an optimization problem is $\mathcal{NP}$-hard, then we will also say that the optimization problem is ***$\mathcal{NP}$*-hard**. So, the optimization version of the travelling salesman problem is $\mathcal{NP}$-hard. For a wealth of information on $\mathcal{NP}$-hard optimization problems and their approximability properties, see the book [33] by Ausiello, Crescenzi, Gambosi, Kann, Marchetti-Spaccamela and Protasi.

From a complexity point of view, there is no significant difference between a decision problem and its optimization analogue (if it exists). To illustrate this statement, let us consider the problem of deciding whether a strong digraph has a cycle of length at least k (here k is part of the input). The optimization analogue is the problem of finding a cycle of maximum length in a strong digraph. If we solve the optimization problem, we easily obtain a solution to the decision problem: just check whether the length of the longest cycle is at least k. On the other hand, using binary search one can find an answer to the optimization problem by solving a number of decision problems. In our example, we first check whether or not the digraph under consideration has a cycle of length at least $n/2$. Then, solve the analogous problem with $n/4$ (if D has no cycle of length at least $n/2$) or $3n/4$ (if D has a cycle of length at least $n/2$) instead of $n/2$, etc. So, we would need to solve $O(\log n)$ decision problems, in order to obtain an answer to the optimization problem.

1.10 Application: Solving the 2-Satisfiability Problem

In this section we deal with a problem that is not a problem on digraphs, but it has applications to several problems on graphs, in particular when we want to decide whether a given undirected graph has an orientation with certain properties. In Chapter 8 we will give examples of this. We will show how to solve this problem efficiently using the algorithm for strong components of digraphs from Chapter 4.

A **boolean variable** x is a variable that can assume only two values 0 and 1. The **sum** of boolean variables $x_1 + x_2 + \ldots + x_k$ is defined to be 1 if at least one of the x_i's is 1 and 0 otherwise. The **negation** $\overline{x}$ of a boolean variable x is the variable that assumes the value $1 - x$. Hence $\overline{\overline{x}} = x$. Let X be a set of boolean variables. For every $x \in X$ there are two **literals**, over x, namely x itself and $\overline{x}$. A **clause** C over a set of boolean variables X is a sum of literals over the variables from X. The **size** of a clause is the number of literals it contains. For example, if u, v, w are boolean variables with values $u = 0, v = 0$ and $w = 1$, then $C = (u + \overline{v} + \overline{w})$ is a clause of size 3, its value is 1 and the literals in C are u, $\overline{v}$ and $\overline{w}$. An assignment of values to the set of variables X of a boolean expression is called a **truth assignment**. If the variables are $x_1, \ldots, x_k$, then we denote a truth assignment by $t = (t_1, \ldots, t_k)$. Here it is understood that x_i will be assigned the value t_i for $i = 1, \ldots, k$.

The **2-satisfiability problem**, also called **2-SAT**, is the following problem. Let $X = \{x_1, \ldots, x_k\}$ be a set of boolean variables and let $C_1, \ldots, C_r$ be a collection of clauses, all of size 2, for which every literal is over X. Decide if there exists a truth assignment $t = (t_1, \ldots, t_k)$ to the variables in X such that the value of every clause will be 1. This is equivalent to asking whether or not the boolean expression $\mathcal{F} = C_1 * \ldots * C_p$ can take the value 1. Depending on whether this is possible or not, we say that $\mathcal{F}$ is **satisfiable** or **unsatisfiable**. Here '$*$' stands for **boolean multiplication**, that is, $1 * 1 = 1$, $1 * 0 = 0 * 1 = 0 * 0 = 0$. For a given truth assignment $t = (t_1, \ldots, t_k)$ and literal q we denote by $q(t)$ the value of q when we use the truth assignment t (i.e. if $q = \overline{x_3}$ and $t_3 = 1$, then $q(t) = 1 - 1 = 0$)

To illustrate the definitions, let $X = \{x_1, x_2, x_3\}$ and let $C_1 = (\overline{x_1} + \overline{x_3})$, $C_2 = (x_2 + \overline{x_3})$, $C_3 = (\overline{x_1} + x_3)$ and $C_4 = (x_2 + x_3)$. Then it is not difficult to check that $\mathcal{F} = C_1 * C_2 * C_3 * C_4$ is satisfiable and that taking $x_1 = 0, x_2 = 1, x_3 = 1$ we obtain $\mathcal{F} = 1$.

If we allow more than 2 literals per clause then we obtain the more general problem **Satisfiability** (also called **SAT**) which is $\mathcal{NP}$-complete, even if all clauses have size 3, in which case it is also called **3-SAT** (see e.g. page 359 in the book [600] by Papadimitriou and Steiglitz). (In his proof of the existence of an $\mathcal{NP}$-complete problem, Cook used the satisfiability problem and showed how every other problem in $\mathcal{NP}$ can be reduced to this problem.) Below we will show how to reduce 2-SAT to the problem of finding the strong components in a certain digraph. We shall also show how to find a satisfying truth assignment if one exists. This step is important in applications, such as those in Chapter 8.

Let $C_1, \ldots, C_r$ be clauses of size 2 such that the literals are taken among the variables $x_1, \ldots, x_k$ and their negations and let $\mathcal{F} = C_1 * \ldots * C_r$ be an instance of 2-SAT. Construct a digraph $D_{\mathcal{F}}$ as follows. Let $V(D_{\mathcal{F}}) = \{x_1, \ldots, x_k, \overline{x_1}, \ldots, \overline{x_k}\}$ (i.e. $D_{\mathcal{F}}$ has two vertices for each variable, one for the variable and one for its negation). For every choice of $p, q \in V(D_{\mathcal{F}})$ such that some C_i has the form $C_i = (p + q)$, $A(D_{\mathcal{F}})$ contains an arc from $\overline{p}$ to q and an arc from $\overline{q}$ to p (recall that $\overline{\overline{x}} = x$). See Figure 1.20 for examples of a 2-SAT expressions and the corresponding digraphs. The first expression is satisfiable, the second is not.

Lemma 1.10.1 *If $D_{\mathcal{F}}$ has a (p, q)-path, then it also has a $(\overline{q}, \overline{p})$-path. In particular, if p, q belong to the same strong component in $D_{\mathcal{F}}$, then $\overline{p}, \overline{q}$ belong to the same strong component in $D_{\mathcal{F}}$.*

Proof: This follows easily by induction on the length of a shortest (p, q)-path, using the fact that $(x, y) \in A(D_{\mathcal{F}})$ if and only if $(\overline{y}, \overline{x}) \in A(D_{\mathcal{F}})$. □

Lemma 1.10.2 *If $D_{\mathcal{F}}$ contains a path from p to q, then, for every satisfying truth assignment t, $p(t) = 1$ implies $q(t) = 1$.*

Proof: Observe that $\mathcal{F}$ contains a clause of the form $(\overline{a} + b)$ and every clause takes the value 1 under any satisfying truth assignment. Thus, by the fact that t is a satisfying truth assignment and by the definition of $D_{\mathcal{F}}$, we have that for every arc $(a, b) \in A(D_{\mathcal{F}})$, $a(t) = 1$ implies $b(t) = 1$. Now the claim follows easily by induction on the length of the shortest (p, q)-path in $D_{\mathcal{F}}$. □

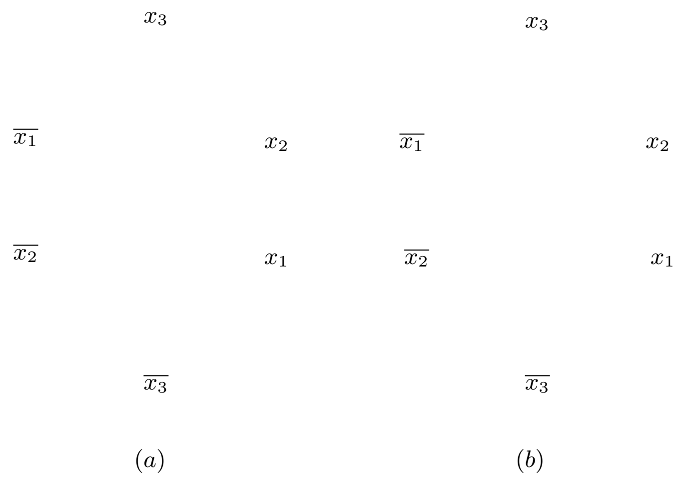


Figure 1.20 The digraph $D_{\mathcal{F}}$ is shown for two instances of 2-SAT. In (a) $\mathcal{F} = (\overline{x_1} + \overline{x_3}) * (x_2 + \overline{x_3}) * (\overline{x_1} + x_3) * (x_2 + x_3)$ and in (b) $\mathcal{F} = (x_1 + x_2) * (\overline{x_1} + x_2) * (\overline{x_2} + x_3) * (\overline{x_2} + \overline{x_3})$

The following is an easy corollary of Lemma 1.10.1 and Lemma 1.10.2.

Corollary 1.10.3 *If t is a satisfying truth assignment, then for every strong component D' of $D_{\mathcal{F}}$ and every choice of distinct vertices $p, q \in V(D')$ we have $p(t) = q(t)$. Furthermore we also have $\overline{p}(t) = \overline{q}(t)$.* □

Lemma 1.10.4 *$\mathcal{F}$ is satisfiable if and only if for every $i = 1, 2, \ldots, k$, no strong component of $D_{\mathcal{F}}$ contains both the variable x_i and its negation $\overline{x_i}$.*

Proof: Suppose t is a satisfying truth assignment for $\mathcal{F}$ and that there is some variable x_i such that x_i and $\overline{x_i}$ are in the same strong component in $D_{\mathcal{F}}$. Without loss of generality $x_i(t) = 1$ and now it follows from Lemma 1.10.2 and the fact that $D_{\mathcal{F}}$ contains a path from x_i to $\overline{x_i}$ that we also have $\overline{x_i}(t) = 1$ which is impossible. Hence if $\mathcal{F}$ is satisfiable, then for every $i = 1, 2, \ldots, k$, no strong component of $D_{\mathcal{F}}$ contains both the variable x_i and its negation $\overline{x_i}$.

Now suppose that for every $i = 1, 2, \ldots, k$, no strong component of $D_{\mathcal{F}}$ contains both the variable x_i and its negation $\overline{x_i}$. We will show that $\mathcal{F}$ is satisfiable by constructing a satisfying truth assignment for $\mathcal{F}$.

Let $D_1, \ldots, D_s$ denote some acyclic ordering of the strong components of $D_{\mathcal{F}}$ (i.e. there is no arc from D_j to D_i if $i < j$). Recall that by Proposition 1.4.3, such an ordering exists. We claim that the following algorithm will determine a satisfying truth assignment for $\mathcal{F}$: first mark all vertices 'unassigned' (meaning truth value still pending). Then going backwards starting from D_s and ending with D_1 we proceed as follows. If there is any vertex $v \in V(D_i)$ such that $\overline{v}$ has already been assigned a value, then assign all vertices in D_i the value 0 and otherwise assign all vertices in D_i the value 1. Observe that this means that, for every variable x_i, we will always assign the value 1 to whichever of $x_i, \overline{x_i}$ belongs to the strong component with the highest index. To see this, let p denote whichever of $x_i, \overline{x_i}$ belongs to the strong component of highest index j. Let $i < j$ be chosen such that $\overline{p} \in D_i$. Suppose we assign the value 0 to p. This means that at the time we considered p, there was some $q \in D_j$ such that $\overline{q} \in D_f$ for some $f > j$. But then $\overline{p} \in D_f$, by Lemma 1.10.1, contradicting the fact that $i < f$.

Clearly all vertices in $V(\mathcal{F})$ will be assigned a value, and by our previous argument this is consistent with a truth assignment for the variables of $\mathcal{F}$. Hence it suffices to prove that each clause has value 1 under the assignment. Suppose some clause $C_f = (p + q)$ attains the value 0 under our assignment. By our observation above, the reason we did not assign the value 1 to p was that at the time we considered p we had already given the value 1 to $\overline{p}$ and $\overline{p}$ belonged to a component D_j with a higher index than the component D_i containing p. Thus the existence of the arc $(\overline{p}, q) \in A(D_{\mathcal{F}})$ implies that $q \in D_h$ for some $h \geq j$. Applying the analogous argument to q we conclude that $\overline{q}$ is in some component D_g with $g > h$. But then, using the existence of the arc $(\overline{q}, p)$, we get that $i \geq g > h \geq j > i$, a contradiction. This shows that we have indeed found a correct truth assignment for $\mathcal{F}$ and hence the proof is complete. □

In Chapter 4 we will see that for any digraph D one can find the strong components of D and an acyclic ordering of these in $O(n+m)$ time. Since the number of arcs in $D_{\mathcal{F}}$ is twice the number of clauses in $D_{\mathcal{F}}$ and the number of vertices in $D_{\mathcal{F}}$ is twice the number of variables in $D_{\mathcal{F}}$, it is not difficult to see that the algorithm outlined above can be performed in time $O(k + r)$ and hence we have the following result.

Theorem 1.10.5 *The problem 2-SAT is solvable in linear time with respect to the number of clauses.* □

The approach we adopted is outlined briefly in Exercise 15.6 of the book [600] by Papadimitriou and Steiglitz, see also the paper [230] by Even, Itai and Shamir.

It is interesting to note that if, instead of asking whether $\mathcal{F}$ is satisfiable, we ask whether there exists some truth assignment such that at least ℓ clauses will get the value 1, then this problem, which is called **MAX-2-SAT**, is $\mathcal{NP}$-complete as shown by Garey, Johnson and Stockmeyer [304] (here ℓ is part of the input for the problem).

1.11 Exercises

1.1. Let X and Y be finite sets. Show that $|X \cup Y| + |X \cap Y| = |X| + |Y|$.

1.2. Let X and Y be finite sets. Show that $|X \cup Y|^2 + |X \cap Y|^2 \geq |X|^2 + |Y|^2$.

1.3. Find a mistake in the following 'definition' of a subdigraph: $H = (V', A')$ is a subdigraph of $D = (V, A)$ if and only if $V' \subseteq V$ and $A' \subseteq A$ hold.

1.4. **(−)** Draw the six non-isomorphic directed pseudographs of order and size 2.

1.5. **(−)** Prove that the number of vertices of odd degree in a digraph is always even. Hint: use Proposition 1.2.1.

1.6. Prove that for every $n \geq 2$ there exists a unique tournament T on n vertices for which all out-degrees of the vertices are distinct.

1.7. **(−)** Prove that every tournament on $n \geq 2k + 2$ vertices has a vertex of out-degree at least $k + 1$.

1.8. Prove that every undirected graph has two vertices with the same degree.

1.9. **(−)** Prove that, if D and H are isomorphic directed pseudographs, then their converses are also isomorphic.

1.10. Describe an infinite family $\mathcal{F}$ of directed pseudographs such that no $D \in \mathcal{F}$ is isomorphic to its converse.

1.11. **(−) Transitivity of paths.** Let D be a digraph and let x, y, z be vertices in D, $x \neq z$. Prove that, if D has an (x, y)-path and a (y, z)-path, then it contains an (x, z)-path as well.

1.12. **(−) Decomposing a closed walk into cycles.** Prove that every closed walk can be decomposed into a collection of (not necessarily disjoint) cycles.

1.13. **Open walk decomposition.** Prove that every open walk can be decomposed into a path and some cycles (not necessarily disjoint).

1.14. **(−)** Prove that, if the in-degree of every vertex in a digraph D is positive, then D has a cycle.

1.15. **(−)** Let x and y be distinct vertices of a digraph D. Suppose that there is a sequence of cycles $C_1, \ldots, C_k$ in D such that x is in C_1, y is in C_k and C_i and C_{i+1} have at least one common vertex for every $i \in \{1, 2, \ldots, k-1\}$. Prove that there exists an (x, y)-path in D.

1.16. Prove Proposition 1.4.6.

1.17. **(−)** Let G be an (undirected) multigraph. Using Proposition 1.2.1, prove that the sum of degrees of vertices in G equals twice the number of edges in G.

1.18. **Uniqueness of acyclic orderings.** Prove that an acyclic digraph D has a unique acyclic ordering if and only if D is traceable.

1.19. **(−)** Let D be the digraph in Figure 1.21.
   (a) Determine the set of out-neighbours and the set of in-neighbours for all vertices of D.
   (b) Determine the semi-degrees of D.
   (c) Determine $\delta^0(D)$ and $\Delta^0(D)$.
   (d) Is D regular?

1.20. **(−)** Let D be the digraph in Figure 1.21.

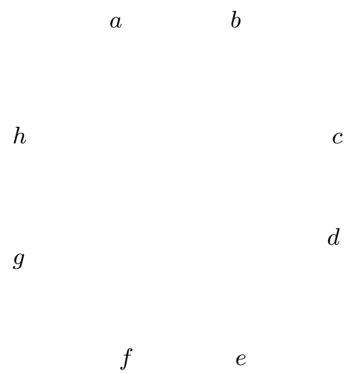


Figure 1.21 A digraph D.

(a) Draw the subdigraphs induced by the vertex sets $\{a, b, c, d, e\}$ and $\{a, d, f, g, h\}$.
(b) Draw the subdigraphs arc-induced by the arc sets $\{ab, cd, ed, hc, ha\}$ and $\{ab, bc, dc, fb, bg\}$.
(c) Let H be the subdigraph of D with vertex set $V(H) = \{a, b, c, d, e, h\}$ and arc set $A(H) = \{ab, bc, dc, ed, eh, ae\}$. List all chords of H in D.
(d) Let H be as above. Is H induced in D? Is it arc-induced?

1.21. **(−)** Let D be the digraph from Figure 1.21. Draw the directed multigraphs $D/\{a, b, c, d, e, h\}$ and $D/\{e, f, h\}$.

1.22. **(−)** Prove that an undirected graph is eulerian if and only if it has an eulerian orientation.

1.23. **(−)** Let D be the digraph from Figure 1.21. Determine the independence number $\alpha(D)$ of D.

1.24. Let D be the digraph in Figure 1.21. Determine the chromatic number of $UG(D)$.

1.25. Let $T = (V, A)$ be a tournament such that every vertex is on a cycle. Prove that for every $a \in A$ the digraph $T - a$ is unilateral.

1.26. Prove that, if a tournament T has a cycle, then it has two hamiltonian paths.

1.27. Let D be a semicomplete multipartite digraph such that every vertex of D is on some cycle. Prove that D is unilateral.

1.28. Let G be an undirected graph. Prove that either G or its complement $\overline{G}$ is connected.

1.29. Prove that every strong tournament T on at least 4 vertices has two distinct vertices x, y such that $T - x$ and $T - y$ are both strong.

1.30. **Strong connectivity is equivalent to cyclic connectivity in digraphs.** A digraph is **cyclically connected** if for every pair x, y of distinct vertices of D there is a sequence of cycles $C_1, \dots, C_k$ such that x is in C_1, y is in C_k and C_i and C_{i+1} have at least one common vertex for every $i \in \{1, 2, \dots, k - 1\}$. Prove that a digraph D is strong if and only if it is cyclically connected.

1.31. (−) Let D be the digraph from Figure 1.21. Find an out-branching with root a in D.

1.32. (−) Prove that a directed multigraph D is strong if and only if it has an out-branching rooted at v for every vertex v of D.

1.33. (+) **Preserving cycle subdigraphs**. Let D be a strong digraph with the property that, for every pair x, y of vertices, the deletion of all arcs between x and y results in a connected digraph. Let $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_t$ be a cycle subdigraph in D such that every cycle C_i has length at least three. Prove that D has a strong spanning oriented subgraph containing $\mathcal{F}$. Hint: use Corollary 1.7.2 (Volkmann [730]).

1.34. Prove Proposition 1.4.6.

1.35. (−) Show that every digraph D contains a path of length at least $\delta^0(D)$.

1.36. Show that every oriented graph D on n vertices and with $\delta^0(D) \geq \lceil (n-1)/4 \rceil$ is strong. Show that this is best possible in terms of $\delta^0(D)$.

1.37. Prove that a connected digraph is strong if and only if every arc is contained in a cycle. Hint: use the result of Exercise 1.30.

1.38. Prove that every edge of a 2-edge-connected graph belongs to a cycle.

1.39. (−) Prove that an undirected tree of order n has $n-1$ edges.

1.40. Prove that every undirected tree has a vertex of degree one.

1.41. Prove that every connected undirected graph G has a spanning tree. Hint: observe that a connected spanning subgraph of G with minimum number of edges is a tree.

1.42. Using the results of the last two exercises, prove that every connected undirected graph G has a vertex x such that $G - x$ is connected.

1.43. An undirected multigraph G is eulerian if it contains a closed trail T such that $A(T) = A(G)$. Prove without using Theorem 1.6.3 that G is eulerian if and only if G is connected and $d(x)$ is even for every vertex x of G.

1.44. Prove using Exercise 1.43 that, if an undirected graph $G = (V, E)$ has no vertex of odd degree, then it has an orientation $D = (V, A)$ such that $d_D^+(v) = d_D^-(v)$ for all $v \in V$.

1.45. Let $G = (V, E)$ be an eulerian graph. Using Exercise 1.43 and Corollary 1.6.4, prove that $d(W)$ is even for every proper subset W of V.

1.46. (−) Prove that a tournament is transitive if and only if it is acyclic. Hint: apply Theorem 1.5.1.

1.47. **Hamiltonian walks in strong digraphs.** Prove that a digraph is strong if and only if it has a Hamilton closed walk.

1.48. (−) Prove that every strong digraph H has an extension $D = H[\overline{K}_{n_1}, \ldots, \overline{K}_{n_h}]$, $h = |V(H)|$, such that D is hamiltonian. Hint: consider a hamiltonian closed walk in H.

1.49. A **transitive triple** in a digraph D is a set of three vertices x, y, z such that xy, xz and yz are arcs of D. Prove that, if a 2-strong digraph D contains a transitive triple, then D has two cycles whose length differ by one.

1.50. List all the acyclic orders of the digraph $SC(D)$ in Figure 1.12.

1.51. **(−) Hamiltonian extensions of cycles.** Characterize extensions of cycles which are hamiltonian.

1.52. Let $D = \vec{C}_r[\overline{K}_{n_1}, \ldots, \overline{K}_{n_r}]$ be an extension of a cycle. Prove that $\kappa(D) = \min\{n_i : \ i = 1, 2, \ldots, r\}$.

1.53. **(+) Traceable semicomplete bipartite digraph characterization.** Prove that a semicomplete bipartite digraph B is traceable if and only if it contains a 1-path-cycle factor $\mathcal{F}$. Hint: demonstrate that, if $\mathcal{F}$ consists of a path and a cycle only, then B is traceable; use it to establish the desired result (Gutin [355]). (See also Chapter 5.)

1.54. (+) Let B be a strong semicomplete bipartite digraph containing a cycle factor consisting of two cycles. Prove that B is hamiltonian (Gutin [353]).

1.55. **(+) Hamiltonian semicomplete bipartite digraph characterization.** Using the result of Exercise 1.54 prove that a semicomplete bipartite digraph B is hamiltonian if and only if B is strong and B contains a cycle factor (Gutin [353]). (See also Chapter 5.)

1.56. (−) Show that every orientation of a quasi-transitive digraph is a quasi-transitive digraph.

1.57. Prove that every strong quasi-transitive digraph of order $n \geq 3$ has a strong orientation, and so does every strong semicomplete bipartite digraph with every partite set of cardinality at least 2. Hint: use Corollary 1.7.2.

1.58. (−) Prove that, if a bipartite tournament has a cycle then it has a 4-cycle.

1.59. (−) Describe an infinite family of strong bipartite tournaments without a 6-cycle.

1.60. Characterize 2-connected undirected graphs for which every cycle has odd length.

1.61. (−) Show that for every undirected graph G on n vertices we have $\chi(G) \geq \lceil n/\alpha(G) \rceil$.

1.62. Show that a digraph D has a cycle factor if and only if its bipartite representation $BG(D)$ contains a perfect matching.

1.63. Describe an infinite family of strong multipartite tournaments, each of which have a cycle factor but is not hamiltonian.

1.64. Describe an infinite family of strong quasi-transitive digraphs, each of which have a cycle factor but is not hamiltonian.

1.65. Give a characterization of hamiltonian complete 3-partite undirected graphs.

1.66. Give an infinite class of strong extended tournaments, none of which is hamiltonian.

1.67. **4-kings in bipartite tournaments.** A vertex v in a digraph D is a k-**king**, if for every $u \in V(D) - \{v\}$ there is a (v, u)-path of length at most k. Prove that a vertex of maximum out-degree in a strong bipartite tournament is a 4-king. For all $s, t \geq 4$ construct strong bipartite tournaments with partite sets of cardinality s and t which do not have 3-kings. (Gutin [356])

1.68. **(+) A special case of the maximum independent set problem**. The maximum independent set problem is as follows. Given an undirected graph G, find an independent set of maximum cardinality in G. The purpose of this exercise is to show that a special case of the maximum independent set problem is equivalent to the 2-satisfiability problem and hence can be solved using any algorithm for 2-SAT.

(a) Let $G = (V, E)$ be a graph on $2k$ vertices and suppose that G has a perfect matching (i.e. a collection $e_1, \ldots, e_k$ of edges with no common end-vertex). Construct an instance $\mathcal{F}$ of 2-SAT which is satisfiable if and only if G has an independent set of k vertices. Hint: fix a perfect matching M of G and let each edge in M correspond to a variable and its negation.

(b) Prove the converse, namely if $\mathcal{F}$ is any instance of 2-satisfiability, then there exists a graph $G = (V, E)$ with a perfect matching such that G has an independent set of size $|V(G)|/2$ if and only if $\mathcal{F}$ is satisfiable.

(c) Prove that it is $\mathcal{NP}$-complete to decide if a given graph has an independent set of size at least ℓ, even if G has a perfect matching. Hint: use a reduction from MAX-2-SAT.

1.69. **Linear time algorithm for finding an acyclic ordering of an acyclic digraph**. Verify that the algorithm given in the proof of Proposition 1.4.3 can be implemented as an $O(n + m)$ algorithm using the adjacency list representation.

1.70. Show how to extend the algorithm MergeHamPathTour (see Subsection 1.9.1) so that it works for tournaments with an arbitrary number of vertices.

1.71. Based on the proof of Theorem 1.5.1, give a polynomial algorithm to find cycles of lengths $3, 4, \ldots, n$ through a given vertex in a strong tournament T. What is the complexity of your algorithm and how do you store information about T and the cycles you find?

1.72. **(+) Fast algorithm for Euler trails.** Demonstrate how to implement the algorithm in the proof of Theorem 1.6.3 as an $O(n + m)$ algorithm. Hint: use adjacency lists along with a suitable data structure to store the trail constructed so far.

1.73. Suppose $\mathcal{S}, \mathcal{T}, \mathcal{K}$ are decision problems such that $\mathcal{S} \leq_{\mathcal{P}} \mathcal{T}$ and $\mathcal{T} \leq_{\mathcal{P}} \mathcal{K}$. Prove that $\mathcal{S} \leq_{\mathcal{P}} \mathcal{K}$.

1.74. The **independent set problem** is as follows: Given a graph $G = (V, E)$ and natural number k, decide whether G has an independent set of size at least k. Show that the independent set problem belongs to the complexity class $\mathcal{NP}$.

1.75. Suppose $\mathcal{W}$ is an $\mathcal{NP}$-complete problem and that $\mathcal{T}$ is a decision problem such that $\mathcal{W} \leq_{\mathcal{P}} \mathcal{T}$. Prove that $\mathcal{T}$ is $\mathcal{NP}$-hard.

1.76. **Finding a cycle of maximum weight in a digraph.** Show that it is an $\mathcal{NP}$-hard problem to find a cycle of maximum weight in a digraph with weights on its arcs. Hint: show how to reduce the Hamilton cycle problem to this problem by a polynomial reduction.

1.77. **The acyclic subdigraph problem.** Let $\mathcal{S}$ be the following decision problem. Given a digraph D and a natural number k, does D contain an induced acyclic subdigraph on at least k vertices? Show that the independent set problem polynomially reduces to $\mathcal{S}$ (the independent set problem is: given a graph G and a number k, does G contain an independent set of size at least k?).

1.78. Show that if a decision problem $\mathcal{S}$ belongs to the complexity class $\mathcal{P}$ then it also belongs to $\mathcal{NP}$.

1.79. Show that $\mathcal{P} \subseteq \mathcal{NP} \cap co-\mathcal{NP}$.

1.80. Show that if there is some decision problem $\mathcal{S}$ which belongs to both of the classes $\mathcal{P}$ and $\mathcal{NPC}$, then $\mathcal{P} = \mathcal{NP}$.

1.81. **(+) Reducing the Hamilton cycle problem to Satisfiability.** Describe a polynomial reduction from the Hamilton cycle problem to the Satisfiability problem. Hint: model different attributes by different sets of clauses. For example you should use one family of clauses to ensure that every vertex is the tail of at least one arc.

1.82. Describe a polynomial reduction from the problem of deciding whether an undirected graph has a matching of size k to the problem MAX-2-SAT.

1.83. **Finding a 1-maximal cycle.** A cycle C in a digraph D is **1-maximal** if D has no cycle C' such that $C - a$ is a subpath of C' for some arc a of C. Describe a polynomial algorithm for finding a 1-maximal cycle in a strong digraph. What is the complexity of your algorithm? Hint: compare it with the proof of Theorem 1.5.1.

1.84. Describe a linear time algorithm to check whether a given acyclic digraph has more than one acyclic ordering. Hint: use the result of Exercise 1.18.

1.85. **Transitive subtournaments in tournaments**. Show that every tournament on 8 vertices contains a transitive tournament on 4 vertices (as an induced subdigraph). Hint: start from a vertex of maximum out-degree. Use the idea above to prove that every tournament on n vertices contains a transitive tournament of size $\Omega(\log n)$.

2. Distances

In this chapter, we study polynomial algorithms which find distances in weighted and unweighted digraphs as well as some related complexity results. We consider bounds on certain distance parameters of a digraph and describe several results on minimizing (and maximizing) the diameter of an orientation of a graph. We study some applications of distances in digraphs to the travelling salesman problem, the one-way street problem and the gossip problem.

Additional terminology and notation are given in Section 2.1. Some basic results on the structure of shortest paths in weighted digraphs are proved in Section 2.2. In Section 2.3 we study algorithms to find shortest paths from a vertex to the rest of the vertices of weighted and unweighted digraphs. We also consider the Floyd-Warshall algorithm to compute distances between all pairs of vertices in a weighted digraph. In Section 2.4 we consider bounds on the following parameters: out-radius, in-radius, radius and diameter of a digraph. The problem of maximizing the diameter of a strong orientation of a bridgeless graph is investigated in Section 2.5. The problem of minimizing the diameter of an orientation of a bridgeless graph, which has applications to the one-way street problem and the gossip problem, is studied extensively in Sections 2.6, 2.7, 2.8 and 2.9. Notice that while both the problem of finding an orientation of minimum diameter and the problem of finding an orientation of maximum diameter are $\mathcal{NP}$-hard, the former is much more complicated from a graph theoretical point of view than the latter.

So-called kings in various classes of digraphs are investigated in Section 2.10. The notion of a king is related to the study of domination in biology and sociology. The last two sections are devoted to applications of distances in digraphs. In Section 2.11 we discuss the one-way street problem and the gossip problem as well as their natural extensions to digraphs. Some recent results on the topics are described. In particular, we state theorems on sharp upper bounds of the minimum diameter orientations of quasi-transitive and semicomplete bipartite digraphs. In Section 2.12 we consider a new approach to compute near optimal solutions to the travelling salesman problem, the exponential neighbourhood local search (ENLS). We show how to utilize the notions and results on distances in the study of ENLS.

2.1 Terminology and Notation on Distances

Let $D = (V, A)$ be a directed pseudo-graph. Recall that, for a set $W \subseteq V$,

$$N_D^+(W) = \bigcup_{w \in W} N^+(w) - W, \; N_D^-(W) = \bigcup_{w \in W} N^-(w) - W.$$

Let $N_D^0(W) = W$, $N_D^{+1}(W) = N_D^+(W)$, $N_D^{-1}(W) = N_D^-(W)$. For every positive integer p, we can define the **pth out-neighbourhood** of W as follows:

$$N_D^{+p}(W) = N_D^+(N_D^{+(p-1)}(W)) - \bigcup_{i=0}^{p-1} N_D^{+i}(W).$$

Similarly, one can define $N_D^{-p}(W)$ for every positive integer p. In particular, $N^{+2}(W) = N^+(N^+(W)) - (W \cup N^+(W))$. Sometimes, $N_D^{+p}(W)$ ($N_D^{-p}(W)$) is called the **open pth out-neighbourhood** (**open pth in-neighbourhood**) of W. We will also use the **closed pth in- and out-neighbourhoods** of a set W of vertices of D which are defined as follows ($p > 0$):

$$N_D^0[W] = W, \quad N_D^{+p}[W] = \bigcup_{i=0}^{p} N_D^{+i}(W), \quad N_D^{-p}[W] = \bigcup_{i=0}^{p} N_D^{-i}(W).$$

To simplify the notation, we set $N_D^+[W] = N_D^{+1}[W]$ and $N_D^-[W] = N_D^{-1}[W]$. See Figure 2.1.

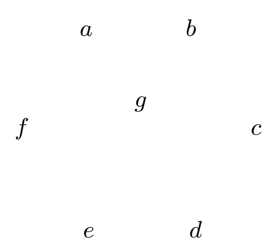


Figure 2.1 A digraph D. The out-neighbourhoods of the set $W = \{a, b\}$ are $N^+(\{a, b\}) = \{f, g\}$, $N^{+2}(\{a, b\}) = \{e\}$, $N^{+3}(\{a, b\}) = \{d\}$, $N^{+4}(\{a, b\}) = \{c\}$. The closed out-neighbourhoods of $W = \{a, b\}$ are $N^+[\{a, b\}] = \{a, b, f, g\}$, $N^{+2}[\{a, b\}] = \{a, b, e, f, g\}$, $N^{+3}[\{a, b\}] = \{a, b, d, e, f, g\}$, $N^{+4}[\{a, b\}] = \{a, b, c, d, e, f, g\}$.

Let $D = (V, A, c)$ be a directed multigraph with a weight function $c : A \to \mathcal{R}$ on its arcs. Recall that the weight of a subdigraph $D' = (V, A')$ of D is given by $c(A') = \sum_{a \in A'} c(a)$. Whenever we speak about the length of a walk we mean the weight of that walk (with respect to c). A **negative cycle** in a weighted digraph $D = (V, A, c)$ is a cycle W whose weight $c(W)$ is negative.

We assume that D has no negative cycle, for otherwise the following definition becomes meaningless. If x and y are vertices of D then the **distance from x to y** in D, denoted $\mathrm{dist}(x, y)$, is the minimum length of a (x, y)-walk, if y is reachable from x, and otherwise $\mathrm{dist}(x, y) = \infty$. Since D has no cycle of negative weight, it follows that $\mathrm{dist}(x, x) = 0$ for every vertex $x \in V$. It follows from Proposition 1.4.1 that there is a shortest (x, y)-walk which is, in fact, a path (if D has no cycle of zero weight either, a shortest walk is always a path). Furthermore, by Proposition 1.4.1, the distance function satisfies the triangle inequality:

$$\mathrm{dist}(x, z) \leq \mathrm{dist}(x, y) + \mathrm{dist}(y, z) \text{ for every triple of vertices } x, y, z. \quad (2.1)$$

The above definitions are applicable to unweighted directed multigraphs as well: simply take the weight of every arc equal to one (then, the length of a walk in the 'weighted' and 'unweighted' cases coincide).

The **distance from a set X to a set Y** of vertices in D is

$$\mathrm{dist}(X, Y) = \max\{\mathrm{dist}(x, y) : \ x \in X, y \in Y\}^1. \quad (2.2)$$

The **diameter** of D is $\mathrm{diam}(D) = \mathrm{dist}(V, V)$. Clearly, D has finite diameter if and only if D is strong. The **out-radius** $\mathrm{rad}^+(D)$ and the **in-radius** $\mathrm{rad}^-(D)$ of D are defined by

$$\mathrm{rad}^+(D) = \min\{\mathrm{dist}(x, V) : \ x \in V\}, \qquad \mathrm{rad}^-(D) = \min\{\mathrm{dist}(V, x) : \ x \in V\}.$$

Because of the obvious duality between out-radius and in-radius, in many cases, we will consider only one of them. The **radius** of D is

$$\mathrm{rad}(D) = \min\{(\mathrm{dist}(x, V) + \mathrm{dist}(V, x))/2 : \ x \in V\}.$$

To illustrate the definitions above, consider the digraph D in Figure 2.1. Here we have $\mathrm{dist}(a, V) = \mathrm{dist}(b, V) = \mathrm{dist}(e, V) = 4$ and $\mathrm{dist}(c, V) = \mathrm{dist}(d, V) = \mathrm{dist}(f, V) = \mathrm{dist}(g, V) = 3$. Furthermore, we have $\mathrm{dist}(V, c) = \mathrm{dist}(V, f) = 4$, $\mathrm{dist}(V, a) = \mathrm{dist}(V, b) = \mathrm{dist}(V, d) = 3$ and $\mathrm{dist}(V, e) = \mathrm{dist}(V, g) = 2$. Now we see that $\mathrm{rad}^+(D) = 3$, $\mathrm{rad}^-(D) = 2$, $\mathrm{rad}(D) = 2.5$ and $\mathrm{diam}(D) = 4$. It is also easy to see that $\mathrm{dist}(\{a, c\}, \{b, f\}) = 3$.

The following proposition gives a characterization of weighted digraphs D of finite out-radius.

Proposition 2.1.1 *A weighted digraph D has a finite out-radius if and only if D has a unique initial strong component.*

Proof: A digraph with two or more initial strong components is obviously of infinite out-radius. If D has only one initial strong component, then D contains an out-branching (by Proposition 1.6.1). Thus, $\mathrm{rad}^+(D) < \infty$. □

[1] This definition may seem somewhat unnatural (with max instead of min), but it simplifies some of the notation in this chapter and also appears quite useful.

This proposition implies that a weighted digraph D has a finite in-radius if and only if D has a unique terminal strong component. Notice that $\mathrm{rad}(D) < \infty$ if and only if D is strong.

For an undirected graph G, we can introduce the notions of distance between pairs of vertices, vertex sets, radius, etc. by considering $\overleftrightarrow{G}$. For an integer r, a vertex v is an ***r*-king** if $\mathrm{dist}(v, V) \leq r$. For example, the vertex c in Figure 2.1 is a 3-king.

2.2 Structure of Shortest Paths

In this section we study elementary, but very important properties of shortest paths in weighted digraphs. We also consider some complexity results on paths in directed and mixed weighted graphs.

We assume that $D = (V, A, c)$ is a weighted digraph with no negative cycle.

Proposition 2.2.1 *If $P = x_1 x_2 \ldots x_k$ is a shortest (x_1, x_k)-path in D, then $P[x_i, x_j]$ is a shortest (x_i, x_j)-path for all $1 \leq i \leq j \leq k$.*

Proof: Suppose that $x_i Q x_j$ is an (x_i, x_j)-path whose length is smaller than that of $P[x_i, x_j]$. Then the weight of the walk $W = P[x_1, x_i] Q P[x_j, x_k]$ is less than the length of P. However, by Proposition 1.4.1, and the fact that D has no negative cycle, W contains an (x_i, x_j)-path R whose length is at most that of W and hence is smaller than that of P, a contradiction. $\square$

Let s be a fixed vertex of D such that $\mathrm{dist}(s, V) < \infty$. Consider spanning subdigraphs of D, each of which contains a shortest path from s to every other vertex in D. The proof of the following theorem shows that given any such subdigraph D' of D, we can construct an out-branching of D rooted at s, which contains a shortest (s, u)-path for every $u \in V - s$.

Theorem 2.2.2 *Let D' and s be as above. There exists an out-branching F_s^+ such that, for every $u \in V$, the unique (s, u)-path in F_s^+ is a shortest (s, u)-path in D.*

Proof: We will give a constructive proof showing how to build F_s^+ from any collection $\{P_v : v \in V - s\}$ of shortest paths from s to the rest of the vertices.

Choose a vertex $u \in V - s$ arbitrarily. Let initially $F_s^+ := P_u$. By Proposition 2.2.1, for every $x \in V(F_s^+)$, the unique (s, x)-path in F_s^+ is a shortest (s, x)-path in D. Hence, if $V(F_s^+) = V$, then we are done. Thus, we may assume that there exists $w \notin V(F_s^+)$. Let z be the last vertex on P_w which belongs to F_s^+. Define H as follows:

$$V(H) := V(F_s^+) \cup V(P_w[z, w]), \; A(H) := A(F_s^+) \cup A(P_w[z, w]).$$

We claim that, for every vertex x in $P_w[z,w]$, the unique (s,x)-path in H is a shortest (s,x)-path in D. By Proposition 2.2.1, $P_w[s,z]$ is a shortest (s,z)-path in D. Since $z \in V(F_s^+)$, the unique (s,z)-path Q in H has the same length as $P_w[s,z]$. Therefore, the length of the path $QP_w[z,x]$ is equal to the length of the path $P_w[s,x]$. Now observe that $QP_w[z,x]$ is the unique (s,x)-path in H. We set $F_s^+ := H$ and use an analogous approach to include all vertices of D and preserve the desired property of F_s^+. □

Our constructive proof above implies the existence of a polynomial algorithm to construct the final out-branching, starting from a collection of shortest paths from s to all other vertices. We call such an out-branching a **shortest path tree from** s. As we will see in Exercises 2.8 and 2.9, the algorithms described in the next section can be easily modified so that they construct a shortest path tree directly, while searching for the shortest paths.

If we allow D to have negative weight cycles, then we obtain the following result for shortest paths (recall that in the presence of negative cycles the length of a shortest walk may not be defined, whereas the length of a shortest path is still well-defined).

Proposition 2.2.3 *It is $\mathcal{NP}$-hard to find a shortest path between a pair of vertices of a given weighted digraph.*

Proof: Let $D = (V, A)$ be an (unweighted) digraph and let $x \neq y$ be vertices of D. Set $c(uv) = -1$ for every arc $uv \in A$. We have obtained a weighted digraph $D' = (V, A, c)$. Clearly, D' has an (x,y)-path of length $1 - n$ if and only if D has a hamiltonian (x,y)-path. Since the problem of finding a hamiltonian (x,y)-path is $\mathcal{NP}$-complete (see Exercise 6.3) and D' can be constructed from D in polynomial time, our claim follows. □

Clearly D' above has a negative cycle if and only if D has any directed cycle. As we will show in Subsection 2.3.2, we can find a longest path in an acyclic digraph in polynomial time, using a reduction to the shortest path problem.

In Section 2.3, we will see that one can check whether a weighted digraph has a negative cycle in polynomial time. However, unless $\mathcal{P} = \mathcal{NP}$, this result cannot be extended to weighted mixed graphs, because of the following theorem by Arkin and Papadimitriou [28].

Theorem 2.2.4 *Given a weighted mixed graph, it is $\mathcal{NP}$-complete to determine whether a negative cycle exists.* □

It follows from Proposition 2.2.3 that it is $\mathcal{NP}$-hard to find a shortest path between a pair of vertices in a weighted mixed graph. More interestingly, Arkin and Papadimitriou showed that the same is true even if we restrict ourselves to weighted mixed graphs without negative cycles [28].

2.3 Algorithms for Finding Distances in Digraphs

In this section we describe well-known algorithms to find distances in weighted and unweighted digraphs. Almost all algorithms which we describe are for finding the distances from a fixed vertex of a digraph to the rest of the vertices. If the given digraph is unweighted then one can use the very simple and fast breadth-first search algorithm, that is introduced in Subsection 2.3.1. If the given digraph D is weighted and acyclic, another fast and simple approach based on dynamic programming is provided in Subsection 2.3.2. When D is an arbitrary digraph, but its weights are non-negative, Dijkstra's algorithm introduced in Subsection 2.3.3 solves the problem. When the weights may be negative, but no negative cycle is allowed, the Bellman-Ford-Moore algorithm given in Subsection 2.3.4 can be applied. This algorithm has the following additional useful property: it can be used to detect a negative cycle (if one exists).

If we are interested in finding the distances between all pairs of vertices of a weighted digraph D, we can apply the Bellman-Ford-Moore algorithm from every vertex of D. However, there is a much faster algorithm, due to Floyd and Warshall. We describe the Floyd-Warshall algorithm in Subsection 2.3.5. The reader can find comprehensive overviews of theoretical and practical issues on the topic in the papers [153] by Cherkassky and Goldberg and [154] by Cherkassky, Goldberg and Radzik.

2.3.1 Breadth-First Search (BFS)

This approach allows one to find quickly the distances from a given vertex s to the rest of the vertices in an unweighted digraph $D = (V, A)$. BFS is based on the following simple idea. Starting at s, we visit each vertex x dominated by s. We set $\text{dist}'(s, x) := 1$ and $s := \text{pred}(x)$ (s is the predecessor of x). Now we visit all vertices y not yet visited and dominated by vertices x of distance 1 from s. We set $\text{dist}'(s, y) := 2$ and $x := \text{pred}(y)$. We continue in this fashion until we have reached all vertices which are reachable from s (this will happen after at most $n-1$ iterations, by Proposition 1.4.1). For the rest of the vertices z (not reachable from s), we set $\text{dist}'(s, z) := \infty$. In other words, we visit the first (open) out-neighbourhood of s, then its second (open) out-neighbourhood, etc. A more formal description of BFS is as follows. At the end of the algorithm, $\text{pred}(v) = \textbf{nil}$ means that either $v = s$ or v is not reachable from s. The correctness of the algorithm is due to the fact that $\text{dist}(s, x) = \text{dist}'(s, x)$ for every $x \in V$. This will be proved below.

BFS
Input: A digraph $D = (V, A)$ and a vertex $s \in V$.
Output: $\text{dist}'(s, v)$ and $\text{pred}(v)$ for all $v \in V$.

1. For each $v \in V$ set $\text{dist}'(s, v) := \infty$ and $\text{pred}(v) := \textbf{nil}$.

2. Set $\mathrm{dist}'(s,s) := 0$. Create a queue Q consisting of s.
3. While Q is not empty do the following. Delete a vertex u, the head of Q, from Q and consider the out-neighbours of u in D one by one. If, for an out-neighbour v of u, $\mathrm{dist}'(s,v) = \infty$, then set $\mathrm{dist}'(s,v) := \mathrm{dist}'(s,u)+1$, $\mathrm{pred}(v) := u$, and put v to the end of Q.

If D is represented by adjacency lists, the complexity of the above algorithm is $O(n+m)$. Indeed, Step 1 requires $O(n)$ time. The time to perform Step 3 is $O(m)$ as the out-neighbours of every vertex are considered only once and $\sum_{x \in V} d^+(x) = m$, by Proposition 1.2.1.

To prove the correctness of BFS, it suffices to prove that $\mathrm{dist}(s,x) = \mathrm{dist}'(s,x)$ for every $x \in V$. By Steps 2 and 3 of the algorithm, $\mathrm{dist}(s,x) \le \mathrm{dist}'(s,x)$. Indeed, $v_1 v_2 \dots v_k$, where $v_1 = s$, $v_k = x$ and $v_i = \mathrm{pred}(v_{i+1})$ for every $i = 1, 2, \dots, k-1$, is an (s,x)-path. By induction on $\mathrm{dist}(s,x)$, we prove that, in fact, the equality holds. If $\mathrm{dist}(s,x) = 0$, then $x = s$ and the result follows. Suppose that $\mathrm{dist}(s,x) = k > 0$ and consider a shortest (s,x)-path P. Let y be the predecessor of x, i.e., $y = x_P^-$. By the induction hypothesis, $\mathrm{dist}'(s,y) = \mathrm{dist}(s,y) = k-1$. Since x is dominated by y , by the algorithm, $\mathrm{dist}'(s,x) \le \mathrm{dist}'(s,y) + 1 = k = \mathrm{dist}(s,x)$. Combining $\mathrm{dist}(s,x) \le \mathrm{dist}'(s,x)$ with $\mathrm{dist}'(s,x) \le \mathrm{dist}(s,x)$, we are done.

The BFS algorithm allows one to compute the radius, out-radius, in-radius and diameter of a digraph in time $O(n^2 + nm)$. Using the array pred one can generate the actual paths. We finish this section with the following two important observations which are stated as propositions. Proposition 2.3.1 follows from the description of BFS. Proposition 2.3.2 has been already proved. In both propositions $D = (V, A)$ is a directed multigraph with a specified vertex s.

Proposition 2.3.1 *Let U be the set of vertices reachable from s. Then (U, B), where $B = \{(\mathrm{pred}(v), v) : v \in U - s\}$ is an out-branching in $D\langle U \rangle$ with root s.* □

We call the out-branching in the above proposition a **BFS tree** of $D\langle U \rangle$ with **root** s, or simply a **BFS tree from** $\boldsymbol{s}$. It is instructive to compare Proposition 2.3.1 with Theorem 2.2.2.

Proposition 2.3.2 *Let $\mathrm{dist}(s, V) < \infty$. For every non-negative integer $p \le \mathrm{dist}(s, V)$, we have $N^{+p}(s) = \{v \in V : \mathrm{dist}(s,v) = p\}$.* □

Given an directed multigraph $D = (V, A)$ and a vertex s we call sets

$$N^0(s), N^+(s), N^{+2}(s), N^{+3}(s), \dots,$$

the **distance classes from** $\boldsymbol{s}$. By the proposition above, $N^{+i}(s)$ consists precisely of those vertices whose distance from s is i. See Figure 2.2 for an illustration of a BFS tree and the corresponding distance classes.

Summarizing the discussion above we obtain the following.

Theorem 2.3.3 *When applied to a directed multigraph D and a vertex s in D, the BFS algorithm correctly determines a BFS tree T from s in D in time $O(n+m)$. Furthermore, the distance classes from s in D are the same as the distance classes from s in T.* □

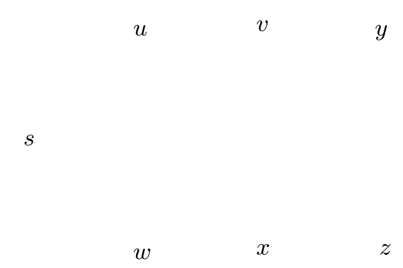


Figure 2.2 A digraph D with a BFS tree indicated by the fat arcs. The distance classes from s are $N^0(s) = s$, $N^+(s) = \{u, w\}$, $N^{+2}(s) = \{v, x, y\}$ and $N^{+3}(s) = \{z\}$.

2.3.2 Acyclic Digraphs

Let $D = (V, A, c)$ be an acyclic weighted digraph. We will show that the distances from a vertex s to the rest of the vertices can be found quite easily, using dynamic programming. Without loss of generality, we may assume that the in-degree of s is zero. Let $\mathcal{L} = v_1, v_2, \ldots, v_n$ be an acyclic ordering of the vertices of D such that $v_1 = s$. Clearly, $\mathrm{dist}(s, v_1) = 0$. For every i, $2 \le i \le n$, we have

$$\mathrm{dist}(s, v_i) = \begin{cases} \min\{\mathrm{dist}(s, v_j) + c(v_j, v_i) : \ v_j \in N^-(v_i)\} & \text{if } N^-(v_i) \neq \emptyset \\ \infty & \text{otherwise.} \end{cases} \tag{2.3}$$

The correctness of this formula can be shown by the following argument. We may assume that v_i is reachable from s. Since the ordering $\mathcal{L}$ is acyclic, the vertices of a shortest path P from s to v_i belong to $\{v_1, v_2, \ldots, v_i\}$. Let v_k be the vertex dominating v_i in P. By induction, $\mathrm{dist}(s, v_k)$ is computed correctly using (2.3). The term $\mathrm{dist}(s, v_k) + c(v_k, v_i)$ is one of the terms in the right-hand side of (2.3). Clearly, it provides the minimum.

The algorithm has two phases: the first finds an acyclic ordering, the second implements Formula (2.3). The complexity of this algorithm is $O(n+m)$ since the first phase runs in time $O(n+m)$ (see Section 4.1) and the second phase requires the same asymptotic time due to the formula $\sum_{x \in V} d^-(x) = m$ in Proposition 1.2.1. Hence we have shown the following.

Theorem 2.3.4 *The shortest paths from a fixed vertex s to all other vertices can be found in time $O(n+m)$ for acyclic digraphs.* □

We can also find the length of longest (s, x)-paths in linear time in any acyclic digraph, by replacing the weight $c(uv)$ of every arc uv with $-c(uv)$. In particular, we can see immediately that the longest path problem for acyclic digraphs is solvable in polynomial time. In fact, a longest path of an acyclic digraph can always be found in linear time:

Theorem 2.3.5 *For acyclic digraphs a longest path can be found in time $O(n + m)$.*

Proof: Exercise 2.6. □

2.3.3 Dijkstra's Algorithm

The next algorithm, due to Dijkstra [192], finds the distances from a given vertex s in a weighted digraph $D = (V, A, c)$ to the rest of the vertices, provided that all the weights of arcs are non-negative.

In the course of the execution of Dijkstra's algorithm, the vertex set of D is partitioned into two sets, P and Q. Moreover, a parameter δ_v is assigned to every vertex $v \in V$. Initially all vertices are in Q. In the process of the algorithm, the vertices reachable from s move from Q to P. While a vertex v is in Q, the corresponding parameter δ_v is an upper bound on $\mathrm{dist}(s, v)$. Once v moves to P, we have $\delta_v = \mathrm{dist}(s, v)$. A formal description of Dijkstra's algorithm follows.

Dijkstra's algorithm

Input: A weighted digraph $D = (V, A, c)$, such that $c(a) \geq 0$ for every $a \in A$, and a vertex $s \in V$.

Output: The parameter δ_v for every $v \in V$ such that $\delta_v = \mathrm{dist}(s, v)$.

1. Set $P := \emptyset$, $Q := V$, $\delta_s := 0$ and $\delta_v := \infty$ for every $v \in V - s$.
2. While Q is not empty do the following.
   Find a vertex $v \in Q$ such that $\delta_v = \min\{\delta_u : \ u \in Q\}$.
   Set $Q := Q - v$, $P := P \cup v$.
   $\delta_u := \min\{\delta_u, \delta_v + c(v, u)\}$ for every $u \in Q \cap N^+(v)$.

To prove the correctness of Dijkstra's algorithm, it suffices to show that the following proposition holds.

Proposition 2.3.6 *At any time during the execution of the algorithm, we have that*

(a) For every $v \in P$, $\delta_v = \mathrm{dist}(s, v)$.
(b) For every $u \in Q$, δ_u is the distance from s to u in the subdigraph of D induced by $P \cup u$.

Proof: When $P = \emptyset$, $\delta_s = \mathrm{dist}(s, s) = 0$ and the estimates $\delta_u = \infty$, $u \in V - s$, are also correct.

Assume that $P = P_0$ and $Q = Q_0$ are such that the statement of this proposition holds. If $Q_0 = \emptyset$, we are done. Otherwise, let v be the next vertex chosen by the algorithm. Since Part (b) follows from Part (a) and the way in which we update δ_u in the algorithm, it suffices to prove Part (a) only. Suppose that (a) does not hold for $P = P_0 \cup v$. This means that $\delta_v > \mathrm{dist}(s, v)$. Let W be a shortest (s, v)-path in D. Since $\delta_v > \mathrm{dist}(s, v)$, W must contain at least one vertex from $Q = Q_0 - v$. Let u be the first vertex on W which is not in P_0. Clearly, $u \neq v$. By Proposition 2.2.1 and the fact that $u \in W$, we have $\mathrm{dist}(s, u) \leq \mathrm{dist}(s, v)$. Furthermore, since the statement of this proposition holds for P_0 and Q_0, we obtain that $\mathrm{dist}(s, u) = \delta_u$. This implies that $\delta_u = \mathrm{dist}(s, u) \leq \mathrm{dist}(s, v) < \delta_v$. In particular, $\delta_u < \delta_v$, which contradicts the choice of v by the algorithm. □

Each time a new vertex v is to be chosen we use $O(n)$ comparisons to find $\min\{\delta_u : \ u \in Q\}$. Updating the parameters takes $O(n)$ time as well. Since Step 2 is performed $n-1$ times, we conclude that the complexity of Dijkstra's algorithm is $O(n^2)$. In fact, Dijkstra's algorithm can be implemented (using so-called Fibonacci heaps) in time $O(n \log n + m)$ (see the paper [278] by Fredman and Tarjan).

Summarizing the discussion above we obtain

Theorem 2.3.7 *Dijkstra's algorithm determines the distances from s to all other vertices in time $O(n \log n + m)$.* □

Figure 2.3 illustrates Dijkstra's algorithm.

It is a challenging open question whether there exists a linear algorithm for calculating the distances from one vertex to all other vertices in a given digraph with no negative cycles. It is easy to see that Dijkstra's algorithm sorts the vertices according to their distances from s. Fredman and Tarjan [278] showed that, if Dijkstra's algorithm can be implemented as a linear time algorithm, then one can sort numbers in linear time. Thorup [715] showed that the opposite claim holds as well: if one can sort numbers in linear time, then Dijkstra's algorithm can be implemented as a linear time algorithm. Currently, no one knows how to sort in linear time[2].

In the case when D is the complete biorientation of an undirected graph G and $c(u, v) = c(v, u)$ holds for every arc uv of D, Thorup [716] recently gave a linear algorithm for calculating shortest paths from a fixed vertex to all other vertices. Thorup's algorithm avoids the sorting bottleneck by building a hierarchical bucketing structure, identifying vertex pairs that may be visited in any order.

[2] Some readers may be confused about this as they may know of a lower bound of $\Omega(n \log n)$ for sorting a set of n numbers. However, this lower bound is only valid for comparison based sorting. There are algorithms for sorting n numbers that are faster than $\Omega(n \log n)$, see e.g. the paper [25] by Anderson.

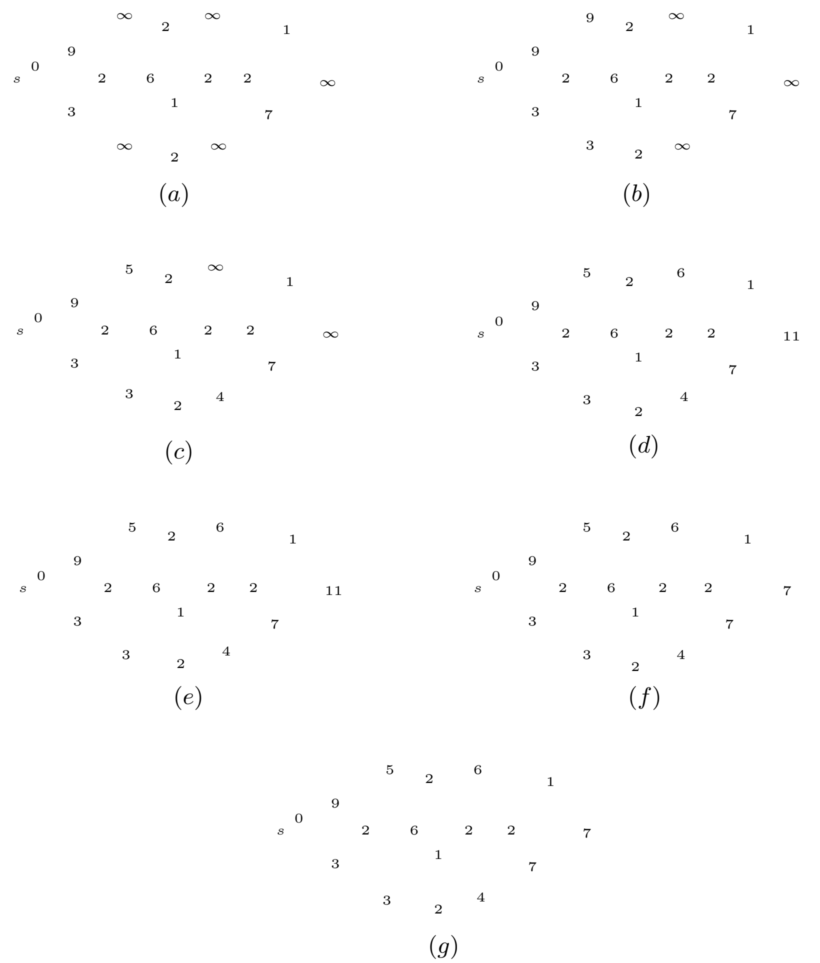


Figure 2.3 Execution of Dijkstra's algorithm. The white vertices are in Q; the black vertices are in P. The number above each vertex is the current value of the parameter δ. (a) The situation after performing the first step of the algorithm. (b)–(g) The situation after each successive iteration of the loop in the second step of the algorithm. The fat arcs indicate the corresponding shortest path tree found by the algorithm if extended as in Exercise 2.8.

2.3.4 The Bellman-Ford-Moore Algorithm

This algorithm originates from the papers [102] by Bellman, [245] by Ford and [572] by Moore. Let $D = (V, A, c)$ be a weighted digraph, possibly with arcs of negative weight. The algorithm described below can be applied to find the distances from a given vertex s in D to the rest of the vertices, provided D has no negative cycle.

Let $\delta(v,m)$ be the length of a shortest (s,v)-path that has at most m arcs. Clearly, $\delta(s,0)=0$ and $\delta(v,0)=\infty$ for every $v \in V-s$. Let $v \in V$. We prove that for every $m \geq 0$,

$$\delta(v,m+1)=\min\{\delta(v,m),\min\{\delta(u,m)+c(u,v):\ u \in N^-(v)\}\}. \qquad (2.4)$$

We show (2.4) by induction on m. For $m=0$, (2.4) trivially holds. For $m \geq 1$, (2.4) is valid due to the following argument. Assume that there is a shortest (s,v)-path P with no more than $m+1$ arcs. If P has at most m arcs, its length is $\delta(v,m)$, otherwise P contains $m+1$ arcs and, by Proposition 2.2.1, consists of a shortest (s,u)-path with m arcs and the arc uv for some $u \in N^-(v)$. If every shortest (s,v)-path has more than $m+1$ arcs, then there is no in-neighbour u of v such that $\delta(u,m)<\infty$. Therefore, Formula (2.4) implies correctly that $\delta(v,m+1)=\infty$.

Since no path has more than $n-1$ arcs, $\delta(v,n-1)=\mathrm{dist}(s,v)$ for every $v \in V-s$. Thus, using (2.4) for $m=0,1,\dots,n-2$, we obtain the distances from s to the vertices of D. This results in the following algorithm.

The Bellman-Ford-Moore algorithm
Input: A weighted digraph $D=(V,A,c)$ with no negative cycle, and a fixed vertex $s \in V$.
Output: The parameter δ_v for every $v \in V$ such that $\delta_v=\mathrm{dist}(s,v)$ for all $v \in V$.

1. Set $\delta_s := 0$ and $\delta_v := \infty$ for every $v \in V-s$.
2. For $i=1$ to $n-1$ do: for each $vu \in A$ update the parameter δ_u by setting $\delta_u := \min\{\delta_u, \delta_v + c(v,u)\}$.

It is easy to verify that the complexity of this algorithm is $O(nm)$. Hence we have

Theorem 2.3.8 *When applied to a weighted directed graph $D=(V,A,c)$ with no negative cycle and a fixed vertex $s \in V$, the Bellman-Ford-Moore algorithm correctly determines the distances from s to all other vertices in D in time $O(nm)$.* □

Figure 2.4 illustrates the execution of the Bellman-Ford-Moore algorithm.

Checking whether D has no negative cycle can be accomplished as follows. Let us assume that D is strong (otherwise, we will consider the strong components of D one by one; an effective algorithm to build the strong components is described in Chapter 4). Let us append the following additional step to the above algorithm:

3. For every arc $vu \in A$ do: if $\delta_u > \delta_v + c(vu)$ then return the message 'the digraph contains a negative cycle'.

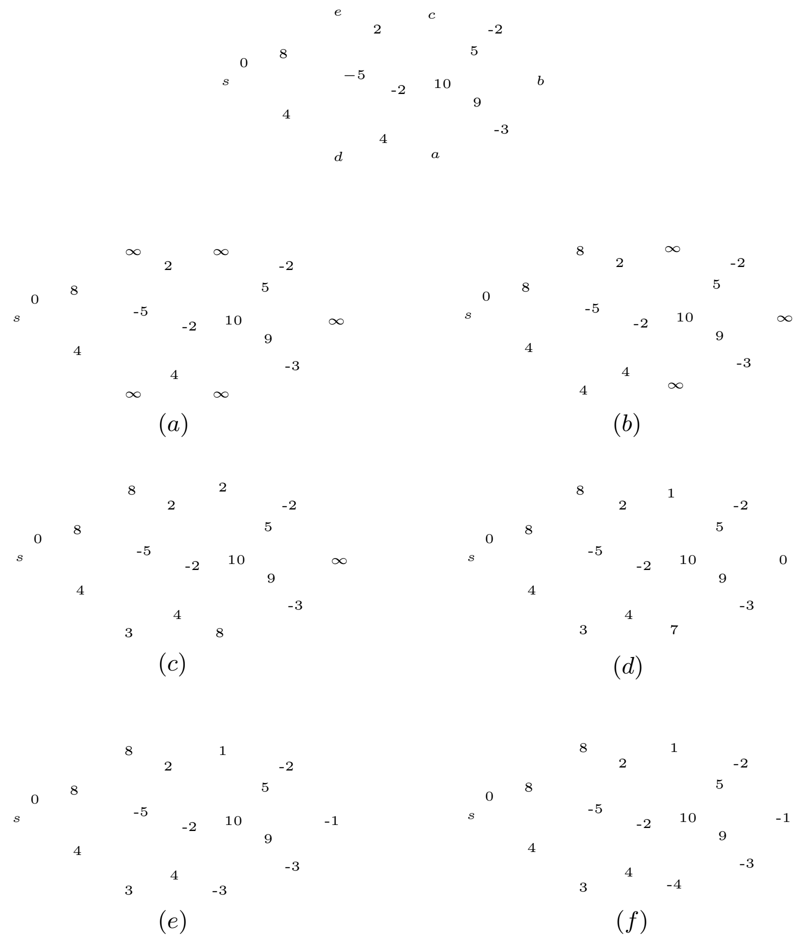


Figure 2.4 Execution of the Bellman-Ford-Moore algorithm. The vertex labellings and arc weights are given in the first digraph. The values of the parameter δ are given near the vertices of the digraphs (a)–(f). In the inner loop of the second step of the algorithm the arcs are considered in the lexicographic order: *ab*, *ac*, *ba*, *bc*, *cb*, *da*, *dc*, *ec*, *ed*, *sd*, *se*. (a) The situation after performing the first step of the algorithm. (b)–(f) The situation after each of the 5 successive executions of the inner loop in the second step of the algorithm.

Theorem 2.3.9 *A strong weighted digraph D has a negative cycle if and only if Step 3 returns its message.*

Proof: Suppose that D has no negative cycle. By the description of Step 2 and Proposition 2.2.1, $\delta_u \leq \delta_v + c(vu)$ for every arc $vu \in A$. Hence, the message will not be returned.

Assume that D has a negative cycle $Z = v_1v_2 \dots v_kv_1$. Assume for the purpose of contradiction that Step 3 of the Bellman-Ford-Moore algorithm does not return the message. Thus, in particular, $\delta_{v_i} \le \delta_{v_{i-1}} + c(v_{i-1}v_i)$ for every $i = 1, 2, \ldots, k$, where $v_0 = v_k$. Hence,

$$\sum_{i=1}^{k} \delta_{v_i} \le \sum_{i=1}^{k} \delta_{v_{i-1}} + \sum_{i=1}^{k} c(v_{i-1}v_i).$$

Since the first two sums in the last inequality are equal, we obtain $0 \le \sum_{i=1}^{k} c(v_{i-1}v_i) = c(Z)$; a contradiction. □

2.3.5 The Floyd-Warshall Algorithm

The above algorithms can be run from all vertices to find *all* pairwise distances between the vertices of a strong digraph D. However, if D has negative weight arcs, but does not contain a negative cycle, we may only use the Bellman-Ford-Moore algorithm n times, which will result in $O(n^2m)$ time (see Exercise 2.19 for a faster method). The Floyd-Warshall algorithm will find the required distances faster, in $O(n^3)$ time. According to Skiena [674], in practice, the algorithm even outperforms Dijkstra's algorithm applied from n vertices (when the weights in D are all non-negative) due to the simplicity of its code (and, thus, smaller hidden constants in the time complexity). The algorithm originates from the papers [243] by Floyd and [734] by Warshall. We assume that we are given a strong weighted digraph $D = (V, A, c)$ that has no negative cycle. In this subsection, it is convenient to assume that $V = \{1, 2, \ldots, n\}$.

Denote by δ_{ij}^m the length of a shortest (i,j)-path in $D\langle\{1, 2, \ldots, m-1\} \cup \{i,j\}\rangle$, for all $1 \le m \le n-1$. In particular, δ_{ij}^1 is the length of the path ij, if it exists. Observe that a shortest (i,j)-path in $D\langle\{1, 2, \ldots, m\} \cup \{i,j\}\rangle$ either does not include the vertex m, in which case $\delta_{ij}^{m+1} = \delta_{ij}^m$, or does include it, in which case $\delta_{ij}^{m+1} = \delta_{im}^m + \delta_{mj}^m$. Therefore,

$$\delta_{ij}^{m+1} = \min\{\delta_{ij}^m, \delta_{im}^m + \delta_{mj}^m\}. \tag{2.5}$$

Observe that $\delta_{ii}^m = 0$ for all $i = 1, 2, \ldots, n$, and, furthermore, for all pairs i, j such that $i \ne j$, $\delta_{ij}^1 = c(i,j)$ if $ij \in A$ and $\delta_{ij}^1 = \infty$, otherwise. Formula (2.5) is also correct when there is no (i,j)-path in $D\langle\{1, 2, \ldots, m\} \cup \{i,j\}\rangle$. Clearly, δ_{ij}^{n+1} is the length of a shortest (i,j)-path (in D). It is also easy to verify that $O(n^3)$ operations are required to compute δ_{ij}^{n+1} for all pairs i, j.

The above assertions can readily be implemented as a formal algorithm (the Floyd-Warshall algorithm, see Exercise 2.14). The Floyd-Warshall algorithm allows one to find the diameter and radius of a weighted digraph without cycles of negative weight in $O(n^3)$ time. Using the algorithm, we may check whether D has no negative cycle. For simplicity let us assume, as above, that D is strong. Then the verification can be based on the following theorem (see, e.g., Lawler's book [509]) whose proof is left to the interested reader as Exercise 2.15.

Theorem 2.3.10 *A weighted digraph D has a negative cycle if and only if $\delta_{ii}^m < 0$ for some $m, i \in \{1, 2, \ldots, n\}$.* □

2.4 Inequalities Between Radius, Out-Radius and Diameter

For a network representing a certain real-world system, it is desirable to have a small diameter as it increases the reliability of the system (see e.g., Fiol, Yebra and Alegre [236]). Small out-radius means that the system has an element that can quickly reach the rest of the elements (for example, by sending a message to them). In-radius and radius have similar interpretations. However, networks representing real-world systems normally do not have many arcs to avoid too costly constructions. The objectives of minimizing the diameter or/and radius (or out-radius) and the size of a digraph clearly contradict each other. Therefore, it is important for a designer to know what kind of trade-off can be achieved. The inequalities of this section give some insight into this problem.

2.4.1 Radius and Diameter of a Strong Digraph

It is well-known that, in a connected undirected graph G, we have $\mathrm{rad}(G) \leq \mathrm{diam}(G) \leq 2\,\mathrm{rad}(G)$. This inequality holds also for strong digraphs (for our definition of radius).

Proposition 2.4.1 *For a strong digraph $D = (V, A)$, we have* $\mathrm{rad}(D) \leq \mathrm{diam}(D) \leq 2\,\mathrm{rad}(D)$.

Proof: Clearly, $\mathrm{rad}(D) \leq \mathrm{diam}(D)$. Let x be a vertex of D such that $(\mathrm{dist}(x, V) + \mathrm{dist}(V, x))/2 = \mathrm{rad}(D)$, and let y, z be vertices of D such that $\mathrm{dist}(y, z) = \mathrm{diam}(D)$. Since $\mathrm{dist}(y, z) \leq \mathrm{dist}(y, x) + \mathrm{dist}(x, z) \leq 2\mathrm{rad}(D)$, we conclude that $\mathrm{diam}(D) \leq 2\,\mathrm{rad}(D)$. □

The following simple bound (called the **Moore bound**) on the order of a strong digraph is important in certain applications [236]. We leave its proof to the reader (Exercise 2.25).

Proposition 2.4.2 *Let n, d and t be the order, the maximum out-degree and the diameter, respectively, of a strong digraph D. Then $n \leq 1+d+d^2+\ldots+d^t$.* □

The Moore bound is attained for $d = 1$ by the cycle $\vec{C}_{t+1}$ and for $t = 1$ by the complete digraph on $d+1$ vertices. However, it is well-known (see Bridges and Toueg [136] and Plesník and Znám [609]) that this bound cannot be attained for $d > 1$ and $t > 1$. Therefore,

$$n < \frac{d^{t+1} - 1}{d - 1}$$

for $d > 1$ and $t > 1$. After simple algebraic transformations, we obtain the following.

Proposition 2.4.3 *Let n, d and t be the order, the maximum out-degree and the diameter, respectively, of a strong digraph D. If $d > 1$ and $t > 1$, then*

$$t \geq \lfloor \log_d(n(d-1)+1) \rfloor.$$

□

The cases $d = 2, 3$ have received special consideration. For $d = 2$, Miller and Fris [566] proved that there is no 2-regular digraph of diameter $t \geq 3$ and order $n = d + d^2 + \ldots + d^t$. Moreover, for most values of k no 2-regular digraphs of order $n = d + d^2 + \ldots + d^t - 1$ exists (see Miller [565]). 3-regular digraphs of order $n = d + d^2 + \ldots + d^t$, with $d = 3$, have been studied by Baskoro, Miller, Plesník and Znám [96].

2.4.2 Extreme Values of Out-Radius and Diameter

In this subsection, we will consider results on the following problems: what is the minimum (maximum) value of the out-radius and diameter of a strong digraph with n vertices and m arcs?

We start with the minimization problem for the out-radius. Theorem 2.4.4 is due to Goldberg [327].

Theorem 2.4.4 *Let D be a strong digraph and let $f(n,m) = \lceil \frac{n-1}{m-n+1} \rceil$. Then $\mathrm{rad}^+(D) \geq f(n,m)$. For all integers $m \geq n \geq 2$, there exists a digraph $D(n,m)$ (which we call the Goldberg digraph $D(n,m)$) of order n and size m whose out-radius is $f(n,m)$.*

Proof: Let v be a vertex of D such that $\mathrm{dist}(v, V) = \mathrm{rad}^+(D)$, and let T be a BFS tree of D with root v. Let also W be the set of vertices $w \in V$ such that $d^+_T(w) = 0$. For a vertex $w \in W$, let $P(w)$ denote the set of vertices, except for v, in the (v,w)-path of T. Then,

$$n - 1 = |\cup_{w \in W} P(w)| \leq \sum_{w \in W} |P(w)| \leq |W| \mathrm{dist}(v, V) = |W| \mathrm{rad}^+(D).$$

Thus,

$$|W| \mathrm{rad}^+(D) \geq n - 1. \tag{2.6}$$

Since D is strong, every vertex $w \in W$ is the tail of an arc in $D - A(T)$. Being a tree, T has $n-1$ arcs (see Exercise 1.39). Hence, $|W| \leq m - (n-1)$. Combining this with (2.6), we obtain that $\mathrm{rad}^+(D) \geq f(n,m)$.

Set $r = n - 1 - (m - n + 1)(f(n, m) - 1)$. It is not difficult to verify that $0 < r \leq m - n + 1$. The digraph $D(n, m)$ is constructed as follows. Take r cycles of length $f(n, m) + 1$ and $m - n + 1 - r$ cycles of length $f(n, m)$, mark a vertex in each cycle by v, and then identify all $m - n + 1$ vertices marked by v. Since $r > 0$, at least one of the cycles in $D(n, m)$ has $f(n, m) + 1$ vertices. Thus, $\mathrm{dist}(v, V(D(n, m))) = f(n, m)$. Hence, $\mathrm{rad}^+(D(n, m)) = f(n, m)$. □

Figure 2.5 depicts $D(10, 14)$. Clearly, $\mathrm{rad}^+(D(10, 14)) = 2$.

Figure 2.5 The Goldberg digraph D(10,14).

Being quite simple, the problem of finding a tight upper bound for the out-radius of a digraph of order n and size m has not been studied in the literature. The following two theorems solve the problems of establishing lower and upper bounds for the diameter of a strong digraph. Theorem 2.4.5 was proved by Goldberg [328]; Theorem 2.4.6 was derived by Ghouila-Houri [314].

Theorem 2.4.5 *Let D be a strong digraph of order n and size m, $m \geq n+1$, and let $g(n, m) = \lceil \frac{2n-2}{m-n+1} \rceil$. Then $\mathrm{diam}(D) \geq g(n, m)$. This bound is the best possible.* □

Theorem 2.4.6 *Let D be a strong digraph of order n and size m. Then $\mathrm{diam}(D) \leq n - 1$, if $n \leq m \leq (n^2 + n - 2)/2$ and $\mathrm{diam}(D) \leq \lfloor n + \frac{1}{2} - \sqrt{2m - n^2 - n + \frac{17}{4}} \rfloor$, otherwise.* □

Oriented graphs of diameter 2 and minimum size (for fixed order n) were discussed by Fűredi, Horak, Pareek and Zhu [285].

2.5 Maximum Finite Diameter of Orientations

For a connected bridgeless multigraph G, let G' denote an orientation of G having maximum finite diameter. Let $\mathrm{lp}(G)$ stand for the length of a longest path of G. The following theorem was obtained by Gutin [366].

Theorem 2.5.1 *Let G be a connected bridgeless graph. Then,* $\mathrm{diam}(G') = \mathrm{lp}(G)$.

Proof: For every strongly connected orientation G_0 of G we obviously have $\mathrm{diam}(G_0) \leq \mathrm{lp}(G)$. Hence, to prove this theorem it suffices to construct some orientation G_1 of G with the property $\mathrm{diam}(G_1) = \mathrm{lp}(G)$.

Let $P = x_1x_2 \dots x_k$ be a longest path of G, and associate each vertex x_i with a label $r(x_i) = i$. Since G has no bridge, the edge $x_{k-1}x_k$ is not a bridge. Consequently, there exists an $(\{x_1, x_2, \dots, x_{k-1}\}, x_k)$-path R_1 which is different from the path $x_{k-1}x_k$. Let x_i be the initial vertex of R_1. Define $r(v) = i$ for all vertices $v \in V(R_1) - \{x_k\}$. Since $x_{i-1}x_i$ is not a bridge there exists an $(\{x_1, x_2, \dots, x_{i-1}\}, \{x_i, x_{i+1}, \dots, x_k\} \cup V(R_1))$-path R_2 which is different from the path $x_{i-1}x_i$. If x_j is the initial vertex of R_2 (observe that $j < i$), then define $r(v) = j$ for all vertices v in R_2 besides the terminal one. Analogously, we can build paths $R_3, R_4, \dots$ and define the label $r(.)$ of the vertices of $R_3, R_4, \dots$ until we obtain a path R_s with the initial vertex x_1 and set $r(v) = 1$ for all vertices v in R_s but the terminal one.

Now, we orient the path P from x_1 to x_k (we obtain the directed path Q), and each path R_i $(i = 1, 2, \dots, s)$ from its end vertex having a bigger label to its other end vertex (we derive the path Q_i). It is easy to check that the oriented graph induced by the arcs of the paths $\cup_{i=1}^s Q_i \cup Q$ is strong. Define

$$X = V(G) - (\cup_{i=1}^s V(R_i) \cup V(P))$$

and suppose that $X \neq \emptyset$ (the case $X = \emptyset$ is easier). Since G has no bridge there exists some vertex $v \in X$ and a pair of paths from v to vertices in $V(G) - X$ with no common vertices (besides v), see Exercise 7.18. We merge these two paths to one (path S_1). Now orient the last path from its end vertex having the bigger label to the one having the smaller label. If the labels of the two end vertices coincide then the orientation is arbitrary. The labels of all other vertices of the path S_1 are the same as the label of terminal vertex of this path.

If $X - V(S_1) \neq \emptyset$ we will continue the construction of paths $S_2, S_3, \dots$ passing over the rest of the vertices of X until $\cup_{i=1}^t V(S_i) = X$, where the orientations and labels are chosen in the same manner. Finally orient each unoriented edge uv from u to v if $r(u) \geq r(v)$ and from v to u otherwise.

Let D denote the obtained oriented graph. The digraph D contains a strongly connected spanning subgraph. Therefore, D is strongly connected. Since all the arcs (u, w) of D, besides those in P, are oriented such that $r(u) \geq r(w)$, there is no path from x_1 to x_k having length less than $k - 1$. Hence, $\mathrm{diam}(D) = k - 1$. □

Since the longest path problem for undirected graphs is $\mathcal{NP}$-hard (see the book [303] by Garey and Johnson), the last theorem implies that the problem to find a maximum finite diameter orientation of a graph is $\mathcal{NP}$-hard as well.

2.6 Minimum Diameter of Orientations of Multigraphs

The same complexity result holds for the following problem: find a minimum diameter orientation of a graph. Indeed, the following assertion holds.

Theorem 2.6.1 (Chvátal and Thomassen) *[164] It is $\mathcal{NP}$-complete to decide whether an undirected graph admits an orientation of diameter 2.*

For a bridgeless multigraph G, let $\mathrm{diam}_{\min}(G)$ denote the minimum diameter of an orientation of G. We will present a minor modification of the original proof of Theorem 2.6.1 by Chvátal and Thomassen [164]. The main difference is in the use of Lemma 2.6.2 (which is applied to two different results in this section). Define a bipartite tournament BT_s, with partite sets U, W, each of cardinality s, as follows. Let $U = \{u_1, u_2, \ldots, u_s\}$ and $W = \{w_1, w_2, \ldots, w_s\}$. The vertex u_i dominates only vertices $w_i, w_{i+1}, \ldots, w_{i+\lfloor s/2 \rfloor -1}$ (the subscripts are taken modulo s) for every $i = 1, 2, \ldots, s$.

Lemma 2.6.2 *Let $s \geq 2$. The diameter $\mathrm{diam}(BT_s)$ equals 3. In particular, $\mathrm{dist}(U,U) = \mathrm{dist}(W,W) = 2$.*

Proof: Clearly, it suffices to show that $\mathrm{dist}(U,U) = \mathrm{dist}(W,W) = 2$. This follows from the fact that, for every $i \neq j$, we have $N^+(u_i) - N^+(u_j) \neq \emptyset$ and, hence, there is a vertex $w \in W$ such that $u_i \to w \to u_j$. □

Lovász [520] proved that it is $\mathcal{NP}$-hard to decide whether a hypergraph of rank[3] 3 is 2-colourable. By the result of Lovász, Theorem 2.6.1 follows from the next theorem.

Theorem 2.6.3 *Given a hypergraph H of rank 3 and order n, one can construct in polynomial time (in n) a graph G such that $\mathrm{diam}_{\min}(G) = 2$ if and only if H is 2-colourable.*

Proof: Let k be the integer satisfying $8 \leq k \leq 11$ and $n + k$ is divisible by 4. Let H_0 be a hypergraph obtained from H by adding k new vertices $v_1, \ldots, v_k$. Moreover, append three new edges $\{\{v_i, v_{i+1}\} : i = 1, 2, 3\}$ to H_0 if H has an odd number of edges, and add four new edges $\{\{v_i, v_{i+1}\} : i = 1, 2, 3, 4\}$ to H_0 otherwise. Observe that H_0 has an even number of edges, which is at least four. To construct G, take disjoint sets R and Q such that the elements of R (Q) are in a one-to-one correspondence with the vertices (the edges) of H_0. Let $G\langle R \rangle$ and $G\langle Q \rangle$ be complete graphs, and $p \in R$ and $q \in Q$ be adjacent if and only if the vertex corresponding to p belongs to the edge corresponding to q (in H_0).

Append four new vertices w_1, w_2, w_3, w_4 and join each of them to all the vertices in $R \cup Q$. Finally, add a new vertex x and join it to all the vertices

[3] Recall that the rank of a hypergraph is the cardinality of its largest edge.

in R. We show that the obtained graph G has the desired property. (Clearly, G can be constructed in polynomial time.)

Assume that G admits an orientation G^* of diameter 2. For a vertex $u \in R$, set $f(u) = 0$ if and only if $x \to u$ in G^*; otherwise, $f(u) = 1$. Since $\mathrm{dist}_{G^*}(x, q) = 2$ ($\mathrm{dist}_{G^*}(q, x) = 2$, respectively) for each $q \in Q$, every edge e of H contains a vertex y such that $f(y) = 0$ ($f(y) = 1$, respectively). Thus H is 2-colourable.

Now assume that H is 2-colourable. Then H_0 admits a 2-colouring which generates a partition $R = R_1 \cup R_2$ such that every edge of H_0 has a vertex corresponding to an element from R_i and $|R_i| \geq 4$ (for every $i = 1, 2$). An orientation G' of G of diameter 2 is defined as follows. Orient the edges in each complete graph $G\langle L\rangle \in \{G\langle R_1\rangle, G\langle R_2\rangle, G\langle Q\rangle\}$ such that the resulting tournament contains the bipartite tournament $BT_{|L|}$. Let A_i, B_i be the partite sets of the bipartite tournaments in $G\langle R_i\rangle$ ($i = 1, 2$) and let A, B be the partite sets of the bipartite tournament in $G\langle Q\rangle$. The rest of the edges in G are oriented as follows:

$$\begin{array}{ll} x \to R_1 \to R_2 \to x, & R_1 \to Q \to R_2, \\ (A_1 \cup A_2) \to w_1 \to A, & B \to w_1 \to (B_1 \cup B_2), \\ (A_1 \cup A_2) \to w_2 \to B, & A \to w_2 \to (B_1 \cup B_2), \\ (B_1 \cup B_2) \to w_3 \to A, & B \to w_3 \to (A_1 \cup A_2), \\ (B_1 \cup B_2) \to w_4 \to B, & A \to w_4 \to (A_1 \cup A_2). \end{array}$$

Using Lemma 2.6.2, it is not difficult to verify that $\mathrm{diam}(G') = 2$. For example, to show that $\mathrm{dist}_{G'}(A_1, V(G')) \leq 2$ and $\mathrm{dist}_{G'}(V(G'), A_1) \leq 2$, it suffices to observe that $\mathrm{dist}_{G'}(A_1, A_1) = 2$ and

$$\begin{aligned} B_1 \cup R_2 \cup Q \cup \{w_1, w_2\} &\subseteq N^+(A_1), \\ \{x, w_3, w_4\} &\subseteq N^+(B_1 \cup R_2 \cup Q \cup \{w_1, w_2\}) \\ B_1 \cup \{x, w_3, w_4\} &\subseteq N^-(A_1), \\ N^-(B_1 \cup \{x, w_3, w_4\}) &\subseteq R_2 \cup Q \cup \{w_1, w_2\}. \end{aligned}$$

□

Chvátal and Thomassen [164] dealt with the following parameter which we call the **strong radius**. The strong radius of a strongly connected digraph $D = (V, A)$, $\mathrm{srad}(D)$, is equal to

$$\min\{\max\{\mathrm{dist}(v, V), \mathrm{dist}(V, v)\} : v \in V\}.$$

Chvátal and Thomassen [164] showed that it is $\mathcal{NP}$-hard to decide whether a graph admits a strongly connected orientation of strong radius 2. The strong radius is of interest because, in particular, $\mathrm{srad}(D) \leq \mathrm{diam}(D) \leq 2\,\mathrm{srad}(D)$ for every strongly connected digraph D (this follows from the fact that $\mathrm{rad}(D) \leq \mathrm{srad}(D)$ for every strong digraph D and Proposition 2.4.1).

Following [164], we prove a sharp upper bound for the value of the strong radius of a strong orientation of a bridgeless connected multigraph. The first part of the proof of Theorem 2.6.4 is illustrated in Figure 2.6.

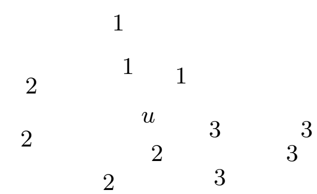


Figure 2.6 Constructing the orientation D of H in the proof of Theorem 2.6.4. The integers on arcs indicate the step number in the process of obtaining D.

Theorem 2.6.4 *[164] Every bridgeless connected multigraph* $G = (V, E)$ *admits an orientation of strong radius at most* $(\mathrm{rad}(G))^2 + \mathrm{rad}(G)$.

Proof: We will show a slightly more general result. Let $u \in V$ be arbitrary and let $\mathrm{dist}_G(u, V) = r$, then there is an orientation L of G such that $\mathrm{dist}_L(u, V) \leq r^2 + r$ and $\mathrm{dist}_L(V, u) \leq r^2 + r$.

Since G is bridgeless, every edge uv is contained in some undirected cycle; let $k(v)$ denote the length of a shortest cycle through uv. It is not difficult to prove (see Exercise 2.28) that, for every $v \in N(u)$,

$$k(v) \leq 2r + 1. \tag{2.7}$$

We claim that there is a subgraph H of G and an orientation D of H with the following properties:

(a) $N_G(u) \subseteq V(H)$.
(b) For each $v \in N(u)$, D has a cycle C_v of length $k(v)$ containing either uv or vu.
(c) D is the union of the cycles C_v.

Observe that by this claim and (2.7), we have

$$\max\{\mathrm{dist}_D(u, V(D)), \mathrm{dist}_D(V(D), u)\} \leq 2r. \tag{2.8}$$

We demonstrate the above claim by constructing H and D step by step. Let uv be an edge in G and let Z_v be an undirected cycle of length $k(v)$ through uv. Orient Z_v arbitrarily as a directed cycle and let C_v denote the cycle obtained this way. Set $H := Z_v,\ D := C_v$. This completes the first step. At step $i(\geq 2)$, we choose an edge uw such that $w \notin V(H)$ and an undirected cycle $Z = w_1w_2 \dots w_kw_1$ in G such that $w_1 = u$, $w_2 = w$, and $k = k(w)$. If no vertex in $Z_w - u$ belongs to H, then append the directed cycle $C_w = w_1w_2 \dots w_kw_1$ to D and the cycle Z to H. Go to the next step.

Otherwise, there is a vertex w_i $(2 \le i \le k)$ such that $w_i \in V(H)$ (and hence $w_i \in V(D)$). Suppose that w_i has the least possible subscript with this property. Since $w_i \in V(D)$, there is some neighbour v of u such that $w_i \in C_v$. (Recall that C_v is a directed cycle.) Let $C_v = v_1v_2 \ldots v_tv_1$, where $u = v_1$, $v \in \{v_2, v_t\}$, and $w_i = v_j$ for some j. By considering the converse of D, if necessary, we may assume, without loss of generality, that $v = v_2$. Now we consider two cases.

Case 1: $w_k \neq v$. In this case, define the directed cycle $C_w = uw_2w_3 \ldots w_iC_v[v_{j+1}, u]$ and observe that C_w has length $k(w)$. (Indeed, if C_w had more than $k(w)$ arcs, the path $C_w[w_i, u]$ would be longer than the path $P_2 = w_iw_{i+1} \ldots w_ku$. In that case, the walk $Z_v[u, v_j]P_2[w_{i+1}, u]$ containing uv would be of length less than $k(v)$; a contradiction.) Let $Z_w := UG(C_w)$. Add C_w to D and Z_w to H. Go to the next step.

Case 2: $w_k = v$. In this case, define the directed cycle C_w as follows: $C_w = C_v[u, v_j]w_{i-1}w_{i-2} \ldots w_2u$ and observe that C_w has length $k(w)$ (the proof of the last fact is similar to the one given in Case 1). Let $Z_w := UG(C_w)$. Add C_w to D and Z_w to H. Go to the next step.

Since $V(G)$ is finite and we add at least one new vertex to H at each step, this process will terminate with the desired subgraph H and its orientation D. Thus, the claim is proved.

Consider the directed multigraph D. In G, contract all the vertices of D into a new vertex u^* (the operation of contraction for undirected multigraphs is similar to that for directed multigraphs) and call the resulting multigraph G^*. Note that G^* is bridgeless and that by the property (a) of the above claim, we obtain $\mathrm{dist}_{G^*}(u^*, V(G^*)) \le r - 1$. By the induction hypothesis, there is an orientation L^* of G^* such that

$$\mathrm{dist}_{L^*}(u^*, V(L^*)) \le r^2 - r \text{ and } \mathrm{dist}_{L^*}(V(L^*), u^*) \le r^2 - r. \tag{2.9}$$

Consider an orientation L of G obtained by combining L^* with D and orienting the rest of the edges in G arbitrarily. By (2.8) and (2.9), we have

$$\mathrm{dist}_L(u, V(L)) \le r^2 + r \text{ and } \mathrm{dist}_L(V(L), u) \le r^2 + r.$$

□

The sharpness of the bound in Theorem 2.6.4 is proved in [164]. Theorem 2.6.4 immediately implies the following.

Corollary 2.6.5 *For every bridgeless connected multigraph G of radius r,* $\mathrm{diam}_{\min}(G) \le 2r^2 + 2r$. □

Plesník [607] generalized Theorem 2.6.4 and Corollary 2.6.5 to orientations of weighted multigraphs.

Theorem 2.6.6 *Let G be a bridgeless connected multigraph in which every edge has weight between 1 and W. If the radius of G is r, then G admits an orientation of strong radius at most $r^2 + rW$ and of diameter at most $2r^2 + 2rW$.* □

Plesník [607] showed that the result of the previous theorem regarding the strong radius is sharp.

Chung, Garey and Tarjan [157] generalized Corollary 2.6.5 to mixed graphs. They proved the following.

Theorem 2.6.7 *Every bridgeless connected mixed graph G of radius r admits an orientation of diameter at most $8r^2 + 8r$. Such an orientation can be found in time $O(r^2(n + m))$.* □

2.7 Minimum Diameter Orientations of Complete Multipartite Graphs

Many authors consider the following parameter $\rho(G)$ of a bridgeless graph G: $\rho(G) := \mathrm{diam}_{\min}(G) - \mathrm{diam}(G)$. It turns out that, for many interesting graphs G, $\rho(G) = 0$, 1 or 2 (a result which is quite different from the 'pessimistic' bound proved in Theorem 2.6.4). In this section, we discuss results on minimum diameter orientations of complete multipartite graphs.

Šoltés [676] obtained the following result for complete bipartite graphs.

Theorem 2.7.1 *If $n_1 \geq n_2 \geq 2$, then $\rho(K_{n_1,n_2}) = 1$ for $n_1 \leq \binom{n_2}{\lfloor n_2/2 \rfloor}$, and $\rho(K_{n_1,n_2}) = 2$, otherwise.*

The original proof of Theorem 2.7.1 is rather long. A shorter proof of this result using the well-known Sperner's lemma[4] is given by Gutin [361]. We present below an adapted version of the proof in [361]. We start from Sperner's lemma. (We call a family $\mathcal{F}$ of subsets of $\{1, 2, \ldots, n\}$ an **antichain** if no set in $\mathcal{F}$ is contained in another.)

Lemma 2.7.2 *Let $\mathcal{F}$ be an antichain on $\{1, \ldots, n\}$. Then*

$$|\mathcal{F}| \leq \binom{n}{\lfloor n/2 \rfloor}.$$

The bound is attained by taking $\mathcal{F}$ to be the family of all subsets of size $\lfloor n/2 \rfloor$. □

[4] For an elegant probabilistic proof of Sperner's lemma, see Alon and Spencer [14].

Proof of Theorem 2.7.1: Let $n_1 \geq n_2 \geq 2$. Let $\mathcal{O}(K)$ be the set of strongly connected orientations of a complete bipartite graph $K = K_{n_1,n_2}$. It is easy to see that no digraph in $\mathcal{O}(K)$ has diameter 2. Thus, it suffices to show that there is an orientation $D \in \mathcal{O}(K)$ of diameter 3 when $n_1 \leq \binom{n_2}{\lfloor n_2/2 \rfloor}$, and that there is an orientation $D \in \mathcal{O}(K)$ of diameter 4 but no orientation of diameter 3 when $n_1 > \binom{n_2}{\lfloor n_2/2 \rfloor}$.

Let us first assume that $n_1 \leq \binom{n_2}{\lfloor n_2/2 \rfloor}$. If $n_1 = n_2$, then the bipartite tournament BT_{n_1} defined just before Lemma 2.6.2 provides the required orientation (see Lemma 2.6.2). Now, consider the case when $n_1 > n_2$. Let V_1 and V_2 be the partite sets of K, $|V_i| = n_i$. Let U be a subset of V_1 of cardinality n_2. Orient the edges between U and V_2 in such a way that the resulting digraph D' is isomorphic to BT_{n_2} and $d^+(v) = \lfloor n_2/2 \rfloor$ for every $v \in U$. Clearly, $\{N^+(v) : \ v \in U\}$ is an antichain on V_2 (see Lemma 2.7.2). This antichain is formed by some subsets of V_2 of cardinality $\lfloor n_2/2 \rfloor$. Since $|V_1| \leq \binom{n_2}{\lfloor n_2/2 \rfloor}$ and there are $\binom{n_2}{\lfloor n_2/2 \rfloor}$ subsets of V_2 (each of cardinality $\lfloor n_2/2 \rfloor$) forming a (maximum) antichain, the out-neighbourhoods of vertices in $V_1 - U$ can be chosen in such a way that the family $\mathcal{F} = \{N^+(v) : \ v \in V_1\}$ is an antichain. The family $\mathcal{F}$ determines an orientation of K which we denote by D. By Lemma 2.6.2, $\mathrm{dist}_{D'}(V_2, V_2) = 2$ and, thus, $\mathrm{dist}_D(V_2, V_2) = 2$. Since the out-neighbourhoods of every pair of vertices in V_1 are not contained in each other, $\mathrm{dist}_D(V_1, V_1) = 2$. Thus, $\mathrm{diam}(D) = 3$ as every vertex in D is dominated by another vertex.

Now let us assume that $n_1 > \binom{n_2}{\lfloor n_2/2 \rfloor}$. Let $H \in \mathcal{O}(K)$ and V_1, V_2 be the partite sets of K such that $n_i = |V_i|$. By Lemma 2.7.2, there is a pair of vertices $x, y \in V_1$ such that $N_H^+(x) \subseteq N_H^+(y)$. Therefore, $\mathrm{dist}_H(x, y) > 2$. Hence, by the obvious parity reason, $\mathrm{dist}_H(x, y) \geq 4$. Thus, there is no orientation of K of diameter 3. To present an orientation H of K of diameter 4, choose a set $W \subset V_1$ of cardinality $\binom{n_2}{\lfloor n_2/2 \rfloor}$. Orient the edges of $K\langle W \cup V_2 \rangle$ such that the resulting digraph H' is isomorphic to the digraph D defined above. Let w be a fixed vertex of W. For a vertex $v \in W \cup V_2$, set $N_H^+(v) = N_{H'}^+(v)$, and for a vertex $v \in V_1 - W$ set $N_H^+(v) = N_H^+(w)$. We have proved that $\mathrm{diam}(H') = 3$. It remains to show that $\mathrm{dist}(V_1 - W, V(H)) \leq 4$ and $\mathrm{dist}(V(H), V_1 - W) \leq 4$. Actually, by the definition of H, it suffices to demonstrate that $\mathrm{dist}(w, w') = 4$, where $w' \in V_1 - W$. The last fact follows from $\mathrm{dist}_H(w, V_2) \leq 3$ and $N_H^-(w') \cap V_2 \neq \emptyset$. $\square$

Let $f(n_1, \ldots, n_k)$ be the minimum possible diameter of a k-partite tournament with partite sets of sizes $n_1, \ldots, n_k$. For $k = 2$ the value of this function was determined in Theorem 2.7.1 (if $\min\{n_1, n_2\} = 1$, then $f(n_1, n_2) = \infty$). For $k \geq 3$ the problem to determine the function $f(n_1, \ldots, n_k)$ was posed independently by Gutin [366] and Plesník [607]. It is easy to show that $2 \leq f(n_1, \ldots, n_k) \leq 3$ for every $k \geq 3$ and all positive integers $n_1, \ldots, n_k$ (see Proposition 2.7.4 below). Thus, it suffices to find out when $f(n_1, \ldots, n_k) = 2$. In [366, 487, 607], it was shown that $f(n_1, \ldots, n_k) = 2$ if $n_1 = n_2 = \ldots = n_k$

except for $k = 4$, $n_1 = n_2 = n_3 = n_4 = 1$ (it is easy to see that $f(1,1,1,1) = 3$). This result was extended by Koh and Tan [488] as follows.

An ordered pair p, q of integers is called a **co-pair** if $1 \leq p \leq q \leq \binom{p}{\lfloor p/2 \rfloor}$. An ordered triple p, q, r of positive integers is called a **co-triple** if p, q and p, r are co-pairs.

Theorem 2.7.3 *If $m_1, \ldots, m_k$ can be partitioned into co-pairs when k is even and into co-pairs and a co-triple when k is odd, then $f(m_1, \ldots, m_k) = 2$.* □

Since even this theorem falls short to provide a complete solution to the above-mentioned problem, we give only a proof of the most basic result on $f(n_1, \ldots, n_k)$ obtained independently by Plesník [607] and Gutin [361].

Proposition 2.7.4 *For every $k \geq 3$ and all positive integers $n_1, \ldots, n_k$, we have $2 \leq f(n_1, \ldots, n_k) \leq 3$.*

Proof: Obviously, $f(n_1, \ldots, n_k) \geq 2$.

If k is odd, let $R(n_1, n_2, \ldots, n_k)$ stand for a multipartite tournament with partite sets $V_1, \ldots, V_k$ of cardinalities $n_1, \ldots, n_k$ such that $V_i \rightarrow V_j$ if and only if $j - i \equiv 1, 2, \ldots, \lfloor k/2 \rfloor \pmod{k}$. If k is even, then $R(n_1, n_2, \ldots, n_k)$ is determined as follows:

$$R(n_1, n_2, \ldots, n_k) - V_k \cong R(n_1, n_2, \ldots, n_{k-1}) \ ,$$

$$V_k \rightarrow V_i \ (i = 1, 3, 5, \ldots, k-1), \ V_j \rightarrow V_k \ (j = 2, 4, 6, \ldots, k-2) \ .$$

We show that $\mathrm{diam} R(n_1, n_2, \ldots, n_k) \leq 3$ for every $k \geq 3$.

Case 1: k is odd, $k \geq 3$. It is sufficient to prove that $\mathrm{dist}(V_1, V_i) \leq 3$ for all $i = 1, 2, \ldots, k$. If $1 < j \leq \lfloor k/2 \rfloor + 1$, then $V_1 \rightarrow V_j$ by the definition. If $\lfloor \frac{k}{2} \rfloor + 1 < j \leq k$, then $V_{\lfloor k/2 \rfloor + 1} \rightarrow V_j$, hence $\mathrm{dist}(V_1, V_j) = 2$. Since $V_1 \rightarrow V_{\lfloor k/2 \rfloor + 1} \rightarrow V_{\lfloor k/2 \rfloor + 2} \rightarrow V_1$, we have $\mathrm{dist}(V_1, V_1) \leq 3$.

Case 2: k is even, $k \geq 4$. Since $R(n_1, \ldots, n_k) - V_k \cong R(n_2, \ldots, n_{k-1})$, we have $\mathrm{dist}(V_i, V_j) \leq 3$ for all $1 \leq i, \ j \leq k-1$. Moreover, $V_k \rightarrow V_i \rightarrow V_{i+1}$ for $i = 1, 3, 5, \ldots, k-3$ and $V_k \rightarrow V_{k-1}$. Therefore $\mathrm{dist}(V_k, V_t) \leq 2$ for $t = 1, 2, \ldots, k-1$. Analogously, $V_i \rightarrow V_{i+1} \rightarrow V_k$ for $i = 1, 3, 5, \ldots, k-3$ and $V_{k-1} \rightarrow V_1 \rightarrow V_2 \rightarrow V_k$. Hence $\mathrm{dist}(V_t, V_k) \leq 3$ for $t = 1, 2, \ldots, k-1$. Finally, $V_k \rightarrow V_1 \rightarrow V_2 \rightarrow V_k$. Therefore $\mathrm{dist}(V_k, V_k) \leq 3$. □

2.8 Minimum Diameter Orientations of Extensions of Graphs

Proposition 2.7.4 was generalized by Koh and Tay [496, 691] to extensions of graphs. We recall the notion of an extension of a graph introduced in Chapter 1. Let H be a graph with vertex set $\{1, \ldots, h\}$ and let $n_1, \ldots, n_h$

be positive integers. Then $G = H[\overline{K}_{n_1}, \ldots, \overline{K}_{n_h}]$ is the graph with vertex set $\{(p_i, i) : \ 1 \le i \le h, \ 1 \le p_i \le n_i\}$ such that vertices (p_i, i) and (p_j, j) are adjacent in G if and only if $ij \in E(H)$. (We call G an extension of H.)

Theorem 2.8.1 (Koh and Tay) *[496] Let H be a connected graph of order $h \ge 3$. Let $G = H[\overline{K}_{n_1}, \ldots, \overline{K}_{n_h}]$ with $n_i \ge 2, \ 1 \le i \le h$. Then,* $\mathrm{diam}(H) \le \mathrm{diam}_{\min}(G) \le \mathrm{diam}(H) + 2$.

Figure 2.7 An orientation F of $G = P_3[\overline{K}_3, \overline{K}_2, \overline{K}_2]$. Observe that $\mathrm{diam}(G) = 2$ and $\mathrm{diam}(F) = 4$.

This theorem is illustrated by Figure 2.7. The requirement $h \ge 3$ is important as one can see from Theorem 2.7.1 ($\mathrm{diam}(K_2) = 1$, but $\mathrm{diam}_{\min}(K_2[\overline{K}_{n_1}, \overline{K}_2]) = 4$ for $n_1 \ge 3$). Clearly, $\mathrm{diam}(H) \le \mathrm{diam}(D)$ for every orientation D of G. To prove the more difficult part of the inequality in Theorem 2.8.1, we will use the following lemma.

Lemma 2.8.2 *[496] Let t_i, n_i be integers such that $2 \le t_i \le n_i$ for $1 \le i \le h$. If the graph $G' = H[\overline{K}_{t_1}, \ldots, \overline{K}_{t_h}]$ admits an orientation F' in which every vertex v lies on a cycle C_v of length not exceeding s, then $G = H[\overline{K}_{n_1}, \ldots, \overline{K}_{n_h}]$ has an orientation F whose diameter is at most* $\max\{s, \mathrm{diam}(F')\}$.

Proof: Given an orientation F' of G', we define an orientation F of G as follows. We have $(p, i)\rightarrow(q, j)$ in F if and only if one of the following holds:

(a) $p < t_i$, $q < t_j$ and $(p, i)\rightarrow(q, j)$ in F'.
(b) $p < t_i$, $q \ge t_j$ and $(p, i)\rightarrow(t_j, j)$ in F'.
(c) $p \ge t_i$, $q < t_j$ and $(t_i, i)\rightarrow(q, j)$ in F'.
(d) $p \ge t_i$ and $q \ge t_j$ and $(t_i, i)\rightarrow(t_j, j)$ in F'.

Let $u = (p, i)$ and $v = (q, j)$ be a pair of distinct vertices in F. If $i \ne j$, then it is clear that $\mathrm{dist}_F(u, v) \le \mathrm{diam}(F')$ (we can use obvious modifications of the corresponding paths in F'). We have the same result if $i = j$ but $p < t_i$ or $q < t_i$. If $i = j$, $p, q \ge t_i$, then using the cycle C_u we conclude that $\mathrm{dist}_F(u, v) \le s$. □

Proof of Theorem 2.8.1: We prove that there exists an orientation D of G such that $\mathrm{diam}(D) \le \mathrm{diam}(H) + 2$. If $\mathrm{diam}(H) = 1$, then this claim follows from Proposition 2.7.4. Thus, we may assume that $\mathrm{diam}(H) \ge 2$.

Define an orientation F' of $H[T_1, \dots, T_h]$, where every $T_i = \overline{K}_2$, as follows:

$$(1,i)\to(1,j)\to(2,i)\to(2,j)\to(1,i) \text{ if and only if } i < j. \tag{2.10}$$

Let $u = (p,i)$ and $v = (q,j)$ be a pair of distinct vertices in F'. We show that $\mathrm{dist}_{F'}(u,v) \le \mathrm{diam}(H)+2$. Suppose that $ik_1k_2 \dots k_sj$ is a path of length $s+1 = \mathrm{dist}_H(i,j)$ in H. Then the path $Q = (p,i), (k_1^*, k_1), (k_2^*, k_2), \dots, (k_s^*, k_s), (j^*, j)$, where $x^* = 1$ or 2, is of length $\mathrm{dist}_H(i,j)$ in F'. If $j^* = q$, then the last inequality follows. Otherwise, i.e. $j^* \ne q$, the path $Q, (3-k_s^*, k_s), (q,j)$ is of length $\mathrm{dist}_H(i,j) + 2$ in F'. Thus, $\mathrm{dist}_{F'}(u,v) \le \mathrm{diam}(H) + 2$. Thus, $\mathrm{diam}(F') \le \mathrm{diam}(H)$. By (2.10), every vertex of F' belongs to a cycle of length 4. Now this theorem follows from Lemma 2.8.2. □

Thus, totally non-trivial extensions (i.e., with at least two vertices in every independent set used for the extension) of bridgeless undirected graphs G can be divided into three classes according to the difference between the minimum diameter of an orientation of the extension (with at least two vertices in every independent set used for the extension) and $\mathrm{diam}(G)$. Some wide subclasses of these three classes have been constructed in [496, 691]. These constructions indicate that perhaps the following conjecture is true.

Conjecture 2.8.3 *[496] If H in Theorem 2.8.1 is of diameter at least 3, then the upper bound on* $\mathrm{diam}_{\min}(G)$ *there can be replaced by* $\mathrm{diam}(H) + 1$.

2.9 Minimum Diameter Orientations of Cartesian Products of Graphs

The **Cartesian product** of a family of undirected graphs $G_1, G_2, \dots, G_n$, denoted by $G = G_1 \times G_2 \times \dots \times G_n$ or $\prod_{i=1}^n G_i$, where $n \ge 2$, is the graph G having $V(G) = V(G_1) \times V(G_2) \times \dots \times V(G_n) = \{(w_1, w_2, \dots, w_n) : w_i \in V(G_i), i = 1, 2, \dots, n\}$ and a pair of vertices $(u_1, u_2, \dots, u_n)$ and $(v_1, v_2, \dots, v_n)$ of G are adjacent if and only if there exists an $r \in \{1, 2, \dots, n\}$ such that $u_rv_r \in E(G_r)$ and $u_i = v_i$ for all $i \in \{1, 2, \dots, n\} - \{r\}$. Let P_n (C_n, K_n) be the (undirected) path (cycle, complete graph) of order n and let T_n stand for a tree of order n. Roberts and Xu [638, 639, 640, 641] and Koh and Tan [484] evaluated the quantity $\rho(P_k \times P_s)$. (We remark that Roberts and Xu [638, 639, 640, 641] considered objective functions other than ρ for orientations of the Cartesian products of undirected paths.) Koh and Tay [491] proved that most of those results can be extended as follows.

Theorem 2.9.1 *For $n \ge 2$, $k_1 \ge 3$, $k_2 \ge 6$ and $(k_1, k_2) \ne (3, 6)$, we have*

$$\rho(\prod_{i=1}^{n} P_{k_i}) = 0.$$

$\square$

This, in particular, generalizes the main result of McCanna [558] on n-cubes, i.e. the graphs $\prod_{i=1}^{n} P_2$. Koh and Tay [490] have obtained the values of $q(r,k) = \rho(C_{2r} \times P_k)$ for $r, k \geq 2$:

(a) $q(r,k) = 0$ if $k \geq 4$.
(b) $q(r,k) = 2$ if $k = 2$ and r is even.
(c) $q(r,k) = 1$, otherwise.

They have also evaluated $\rho(K_m \times P_k)$, $\rho(K_m \times C_{2r+1})$ and $\rho(K_m \times K_n)$ [492], $\rho(K_m \times C_{2r})$ [495] and $\rho(T_m \times T_n)$ [493]. König, Krumme and Lazard [500] studied the Cartesian products of cycles. They proved the following interesting result.

Theorem 2.9.2 *Let p, q be integers with $p, q \geq 6$. If at least one of these two integers is even, then $\rho(C_p \times C_q) = 0$. If both p and q are odd, then $\rho(C_p \times C_q) = 1$.* $\square$

König, Krumme and Lazard [500] evaluated $\rho(C_p \times C_q)$ in most cases when the minimum of p and q is smaller than 6. They also extended the $\rho(C_p \times C_q) = 0$ part of Theorem 2.9.2 to the Cartesian products of three or more cycles. These results are described in more detail in [691]. Some of the above results were extended by Koh and Tay [491], where the following theorem was proved.

Theorem 2.9.3 *For $m \geq 2$, $r \geq 0$, $k_1 \geq 3$, $k_2 \geq 6$ and $(k_1, k_2) \neq (3, 6)$, we have $\rho(\prod_{i=1}^{m} P_{k_i} \times \prod_{i=1}^{r} C_{n_i}) = 0$.* $\square$

This result was further extended by Koh and Tay in [494]. The rest of this subsection is based on [494].

Let $\mathcal{G}$ be the set of all bipartite graphs G such that $\mathrm{diam}(G) \geq 3$ and G admits an orientation (called a $\mathcal{G}$-orientation) of diameter $\mathrm{diam}(G)$, in which every vertex is contained in a cycle of length at most $\mathrm{diam}(G)$. Let $\mathcal{G}^*$ be the set of all bipartite graphs G such that $\mathrm{diam}(G) \geq 3$ and G admits an orientation F (called a $\mathcal{G}^*$-orientation) of diameter $\mathrm{diam}(G)$ with the following further properties: every vertex is contained in a cycle of F of length at most $\mathrm{diam}(G)$ and if $u \rightarrow v$ in F then there exists a (u, v)-walk of length at least three and at most $\mathrm{diam}(G)$.

Let $\mathcal{S}$ be the set of all graphs in which every graph G admits an orientation H (called an $\mathcal{S}$-orientation) such that for all vertices $u, v \in V(H)$ at least one of the following holds:

(a) $\min\{\mathrm{dist}_H(u,v), \mathrm{dist}_H(v,u)\} \leq \mathrm{diam}(G)$.

(b) There are vertices y and z such that

$$\max\{\mathrm{dist}_H(u,y)+\mathrm{dist}_H(v,y),\mathrm{dist}_H(z,u)+\mathrm{dist}_H(z,v)\}\le \mathrm{diam}(G).$$

Let $\mathcal{S}^*$ be the set of all graphs in which every graph G admits an orientation H (called an $\mathcal{S}^*$-orientation) such that for all vertices $u,v\in V(H)$ at least one of the following holds:

(a) $\min\{\mathrm{dist}_H(u,v),\mathrm{dist}_H(v,u)\}\le \mathrm{diam}(G)$.
(b) There is a vertex y such that $\mathrm{dist}_H(u,y)+\mathrm{dist}_H(v,y)\le \mathrm{diam}(G)$.
(c) There is a vertex z such that $\mathrm{dist}_H(z,u)+\mathrm{dist}_H(z,v)\le \mathrm{diam}(G)$.

Clearly, $\mathcal{G}^*\subseteq\mathcal{G}$ and $\mathcal{S}\subseteq\mathcal{S}^*$. Koh and Tay [494] showed the following:

(a) For $m\ge 2$ and $k\ge 4$, $C_{2m}\times P_k\in\mathcal{G}^*$.
(b) $C_4\times C_4\in\mathcal{G}$.
(c) For $m\ge 2$ and $n\ge 3$, $C_{2m}\times C_{2n}\in\mathcal{G}^*$.
(d) If T' and T'' are trees of diameter at least four, then $T'\times T''\in\mathcal{G}^*$.
(e) $\{P_j:\ j\ge 2\}\cup\{C_j:\ j\ge 3\}\cup\{K_j:\ j\ge 1\}\cup\{G:\ \rho(G)=0\}\subset\mathcal{S}$, also $\{K_{p,q}:\ 2\le p\le q\}\subset\mathcal{S}$.
(f) If T is a tree which is not a path, then $T\in\mathcal{S}^*$.
(g) If $\mathcal{G}_2$ is the set of all graphs of diameter two, then $\mathcal{G}_2\subset\mathcal{S}^*$.

Due to the fact that the families $\mathcal{G},\mathcal{G}^*,\mathcal{S},\mathcal{S}^*$ of graphs are quite large, the following results proved by Koh and Tay [494] are undoubtedly interesting.

Theorem 2.9.4 *If $G\in\mathcal{G}$ and $A_i\in\mathcal{S}$, $i=1,2,\dots,n$, then* $\rho(G\times\prod_{i=1}^n A_i)=0$.

Theorem 2.9.5 *If $G\in\mathcal{G}^*$ and $A_i\in\mathcal{S}^*$, $i=1,2,\dots,n$, then* $\rho(G\times\prod_{i=1}^n A_i)=0$. □

We will prove only Theorem 2.9.4 since the proof of Theorem 2.9.5 is similar and is left as Exercise 2.32.

Proof of Theorem 2.9.4: Let $\mathrm{diam}(G)=k$ and let U and W be the partite sets of G. Let F (H_i) be a $\mathcal{G}$-orientation (an $\mathcal{S}$-orientation) of G (A_i, $i=1,2,\dots,n$). We will orient $G\times\prod_{i=1}^n A_i$ inductively as follows:

1. Orient G as F and A_1 as H_1. In $G\times A_1$, orient an edge $\{(x,i),(x,j)\}$ from (x,i) to (x,j) if and only if either $x\in U$ and $ij\in A(H_1)$ or $x\in W$ and $ji\in A(H_1)$; orient an edge $\{(x,i),(y,i)\}$ from (x,i) to (y,i) if and only if $xy\in A(F)$.
2. Suppose that $G\times\prod_{i=1}^r A_i$, where $1\le r\le n-1$, has been oriented. Orient A_{r+1} as H_{r+1}. Orient $G\times\prod_{i=1}^{r+1}A_i$ so that the orientation of $G\times\prod_{i=1}^r A_i\times\{j\}$ is isomorphic to that of $G\times\prod_{i=1}^r A_i$ for each $j\in V(A_{r+1})$ and orient an edge $\{(x,a_1,\dots,a_r,i),(x,a_1,\dots,a_r,j)\}$ from $(x,a_1,\dots,a_r,i)$ to $(x,a_1,\dots,a_r,j)$ if and only if either $x\in U$ and $ij\in A(H_{r+1})$ or $x\in W$ and $ji\in A(H_{r+1})$.

Let F^* be the resulting orientation of $G \times \prod_{i=1}^n A_i$. Define the following sets

$$\begin{aligned} R_{1i} &= \{(u,v) \in V(H_i \times H_i) : \ \mathrm{dist}_{H_i}(u,v) \le \mathrm{diam}(A_i)\}, \\ R_{2i} &= \{(u,v) \in V(H_i \times H_i) : \ (u,v) \notin R_{1i}, \ \mathrm{dist}_{H_i}(v,u) \le \mathrm{diam}(A_i)\}, \\ R_{3i} &= \{(u,v) \in V(H_i \times H_i) : \ (u,v) \notin R_{1i} \cup R_{2i}, \ \exists y,z \in V(H_i) \\ \max &\{\mathrm{dist}_{H_i}(u,y) + \mathrm{dist}_{H_i}(v,y), \mathrm{dist}_{H_i}(z,u) + \mathrm{dist}_{H_i}(z,v)\} \le \mathrm{diam}(A_i)\}. \end{aligned}$$

Observe that R_{1i}, R_{2i}, R_{3i} form a partition of $V(H_i \times H_i)$.

Let $(x, a_1, \ldots, a_n)$ and $(y, b_1, \ldots, b_n)$ be a pair of distinct vertices of F^*. We will construct, in F^*, a path $P_1P_2P_3P_4$ from $(x, a_1, \ldots, a_n)$ to $(y, b_1, \ldots, b_n)$ of length at most $\mathrm{diam}(G \times \prod_{i=1}^n A_i) = k + \sum_{i=1}^n \mathrm{diam}(A_i)$. (See Exercise 2.29.)

Without loss of generality, assume that $x \in U$ (the case of $x \in W$ can be treated similarly). Let x' be the successor of x either on a shortest (x,y)-path in F if $x \neq y$ or on a shortest cycle through x if $x = y$. Clearly, $x' \in W$.

The path P_1 is a shortest path from $(x, a_1, \ldots, a_n)$ to $(x, c_1, \ldots, c_n)$, where c_i, $i = 1, \ldots, n$, is defined as follows:

(a) $c_i = b_i$ if $(a_i, b_i) \in R_{1i}$.
(b) $c_i = a_i$ if $(a_i, b_i) \in R_{2i}$.
(c) If $(a_i, b_i) \in R_{3i}$, we set $c_i = y_i$, where y_i is a vertex satisfying

$$\mathrm{dist}_{H_i}(a_i, y_i) + \mathrm{dist}_{H_i}(b_i, y_i) \le \mathrm{diam}(A_i).$$

The path P_2 is a shortest path from $(x, c_1, \ldots, c_n)$ to $(x', c_1, \ldots, c_n)$. The path P_3 is a shortest path from $(x', c_1, \ldots, c_n)$ to $(x', b_1, \ldots, b_n)$ and the path P_4 is a shortest path from $(x', b_1, \ldots, b_n)$ to $(y, b_1, \ldots, b_n)$. Observe that the total length of P_1 and P_3 does not exceed $\sum_{i=1}^n \mathrm{diam}(A_i)$ and the total length of P_2 and P_4 is at most k. □

2.10 Kings in Digraphs

In this section, we study r-kings in tournaments, semicomplete multipartite digraphs and other generalizations of tournaments. The main emphasis is on 4-kings in semicomplete multipartite digraphs. The notion of a 2-king and some results on 2-kings in tournaments will be generalized in Section 12.3.2.

2.10.1 2-Kings in Tournaments

Studying dominance in certain animal societies, the mathematical sociologist Landau [508] observed that every tournament has a 2-king. In fact, in every tournament T, each vertex x of maximum out-degree is a 2-king. Indeed, for a vertex $y \in T$, $y \neq x$, either $x \to y$ or there is an out-neighbour of x which is an in-neighbour of y. In both cases, $\mathrm{dist}(x, y) \leq 2$. Observe that if a tournament T has a vertex of in-degree zero, this vertex is the only r-king in T for every positive integer r. Moon [569] proved the following.

Theorem 2.10.1 *Every tournament with no vertex of in-degree zero has at least three 2-kings.*

Proof: Exercise 2.35. □

The following example shows that this bound on the number of 2-kings by Moon is sharp. Let T_n be a tournament with vertex set $\{x_1, x_2, \ldots, x_n\}$ and arc set $A = X \cup Y \cup \{x_{n-2}x_n\}$, where

$$X = \{x_i x_{i+1} : \ i = 1, 2, \ldots, n-1\},$$
$$Y = \{x_j x_i : \ 1 \leq i < j - 1 \leq n - 1, (j, i) \neq (n, n-2)\}.$$

It is easy to verify that, for $n \geq 5$, $x_{n-3}, x_{n-2}, x_{n-1}$ are the only 2-kings in T_n (Exercise 2.37), see Figure 2.8.

Figure 2.8 An example of a tournament with exactly three 2-kings. The arcs which are not shown are oriented from right to left.

Since the converse of a tournament is a tournament, the above two results can be reformulated for 2-serfs. (A vertex x is a **2 serf** if $\mathrm{dist}(V, x) \leq 2$.) The concepts of 2-kings and 2-serfs in tournaments were extensively investigated by both mathematicians and political scientists (the latter have studied so-called majority preferences). The interested reader is referred to Reid [630] for a comprehensive recent survey on the topic.

2.10.2 Kings in Semicomplete Multipartite Digraphs

It is easy to see that Proposition 2.1.1 implies that a multipartite tournament T has a finite out-radius if and only if T contains at most one vertex of in-degree zero (Exercise 2.38). Moreover, the following somewhat surprising assertion holds. If a multipartite tournament has finite out-radius, the out-radius is at most four. In other words, every multipartite tournament with at most one vertex of in-degree zero contains a 4-king. (Similar results hold for quasi-transitive digraphs and a certain class of digraphs that includes multipartite tournaments, see Subsection 2.10.3.) This result was proved independently by Gutin [356] and Petrovic and Thomassen [605]. The bound is sharp as there exist infinitely many p-partite tournaments without 3-kings for every $p \geq 2$ [356]. Indeed, bipartite tournaments $\vec{C}_4[\overline{K}_q, \overline{K}_q, \overline{K}_q, \overline{K}_q]$ for $q \geq 2$ do not have 3-kings ($\mathrm{dist}(u, v) = 4$ for distinct vertices u, v from the same $\overline{K}_q$). It is clear that every multipartite tournament, for which the initial strong component is some $\vec{C}_4[\overline{K}_q, \overline{K}_q, \overline{K}_q, \overline{K}_q]$ ($q \geq 2$), has no 3-king either.

Thus, 4-kings are of particular interest in multipartite tournaments. In a number of papers (see, e.g., Gutin [361], Koh and Tan [485, 486, 489], Petrović [604] and the survey paper [630] by Reid) the authors investigate the minimum number of 4-kings in multipartite tournaments without vertices of in-degree zero. (If a multipartite tournament has exactly one vertex of in-degree zero, it contains exactly one 4-king, so this case is trivial.) In our view, the most interesting result in this direction was obtained by Koh and Tan in [485].

Theorem 2.10.2 *Let T be a k-partite tournament with no vertex of in-degree zero. If $k = 2$, T contains at least four 4-kings; it has exactly four 4-kings if its initial strong component consists of a cycle of length four. If $k \geq 3$, T contains at least three 4-kings; it has exactly three 4-kings if its initial strong component consists of a cycle of length three.* □

This theorem can be considered as a characterization of bipartite (p-partite, $p \geq 3$) tournaments with exactly k 4-kings for $k \in \{1, 2, 3, 4\}$ ($k \in \{1, 2, 3\}$). The next theorem by Gutin and Yeo [376] goes further with respect to both exact number of 4-kings and the class of digraphs under consideration.

Theorem 2.10.3 *Let $D = (V, A)$ be a semicomplete multipartite digraph and let k be the number of 4-kings in D. Then*

1. *$k = 1$ if and only if D has exactly one vertex of in-degree zero.*
2. *$k = 2, 3$ or 4 if and only if the initial strong component of D has k vertices.*
3. *$k = 5$ if and only if either the initial strong component Q of D has five vertices or Q contains at least six vertices and possesses a path $P = p_0p_1p_2p_3p_4$ such that $\mathrm{dist}(p_0, p_4) = 4$ and $\{p_1, p_2, p_3, p_4\} \Rightarrow V - V(P)$.*

□

We have seen that a vertex of maximum out-degree in a tournament is a 2-king. It is slightly more difficult to show that a vertex of maximum out-degree in a bipartite tournament is a 4-king (Exercise 1.67). With 4-kings in k-partite tournaments for $k \geq 3$, the situation is more complicated as can be seen from the next theorem by Goddard, Kubicki, Oellermann and Tian [321].

Theorem 2.10.4 *Let T be a strongly connected 3-partite tournament of order $n \geq 8$. If v is a vertex of maximum out-degree in T, then* $\mathrm{dist}(v, V(T)) \leq \lfloor n/2 \rfloor$ *and this bound is best possible.* □

In the rest of this subsection, we will prove the following theorem using an argument adapted from [376].

Theorem 2.10.5 *Every semicomplete multipartite digraph with at most one vertex of in-degree zero has a 4-king.*

For the proof we need the following lemmas:

Lemma 2.10.6 *If $P = p_0 p_1 \ldots p_\ell$ is a shortest path from p_0 to p_ℓ in a semicomplete multipartite digraph D, and $\ell \geq 3$, then there is a (p_ℓ, p_0)-path of length at most 4 in $D\langle V(P)\rangle$.*

Proof: Since $\ell \geq 3$ and P is a shortest path we have $(\{p_0, p_1\}, p_\ell) = \emptyset$. If $p_\ell \rightarrow p_0$ we are done, so assume that p_ℓ and p_0 belong to the same partite set of D. This implies that $p_\ell \rightarrow p_1$. Analogously, $(p_0, \{p_2, p_3\}) = \emptyset$, which implies that either $p_\ell p_1 p_2 p_3 p_0$ or $p_\ell p_1 p_2 p_0$ is a (p_ℓ, p_0)-path of length at most 4 in $D\langle V(P)\rangle$. □

Lemma 2.10.7 *Let D be a semicomplete multipartite digraph and let Q be an initial strong component of D. If Q has at least two vertices, then D has only one initial strong component. Every vertex in Q, which is a 4-king in Q, is a 4-king in D.*

Proof: Assume that $|V(Q)| \geq 2$, but D has another initial strong component Q'. Since Q contains adjacent vertices, there is an arc between Q and Q', a contradiction.

Let x be a 4-king in Q and let $y \in V(D) - V(Q)$ be arbitrary. If x and y are adjacent, then clearly $x \rightarrow y$. Assume that x and y are not adjacent. Since Q is strong, it contains a vertex z dominated by x. Clearly, $x \rightarrow z \rightarrow y$. Hence $\mathrm{dist}(x, y) \leq 2$ and x is a 4-king in D. □

Lemma 2.10.8 *Let D be a strong semicomplete multipartite digraph and let w be a vertex in D. For $i \geq 3$, if $N^{+i}(w) \neq \emptyset$, then* $\mathrm{dist}(N^{+i}(w), N^{+i}[w]) \leq 4$.

Proof: Let $z \in N^{+i}(w)$ be arbitrary. Since a shortest path from w to z is of length $i \geq 3$, by Lemma 2.10.6, $\mathrm{dist}(z, w) \leq 4$. Let $q \in N^{+i}[w] - \{w, z\}$ and let $r_0 r_1 \ldots r_j$ be a shortest (w, q)-path in D. If $1 \leq j \leq 3$ then, since z dominates at least one of the vertices r_0, r_1, either $z r_0 r_1 \ldots r_j$ or $z r_1 \ldots r_j$ is a (z, q)-path in D of length at most 4. If $j \geq 4$ then, since z dominates at least one of the vertices r_{j-3}, r_{j-2}, either $z r_{j-3} r_{j-2} r_{j-1} r_j$ or $z r_{j-2} r_{j-1} r_j$ is a (z, q)-path in D of length at most 4. □

Proof of Theorem 2.10.5: Let D be a semicomplete multipartite digraph with at most one vertex of in-degree zero. If D has a vertex x of in-degree zero, then clearly x is a 2-king in D. Thus, assume that D has no vertex of in-degree zero. Then, every initial strong component Q of D has at least two vertices. By Lemma 2.10.7, Q is unique and every 4-king in Q is a 4-king in D. It remains to show that Q has a 4-king. If every vertex in Q is a 4-king, then we are done. Otherwise, let w be a vertex in Q which is not a 4-king of Q. Then, $r = \mathrm{dist}_Q(w, V(Q)) \geq 5$. By Lemma 2.10.8, $\mathrm{dist}_Q(N_Q^{+r}(w), N_Q^{+r}[w]) \leq 4$, i.e., every vertex in $N_Q^{+r}(w)$ is a 4-king in Q (since $N_Q^{+r}[w] = V(Q)$). □

2.10.3 Kings in Generalizations of Tournaments

Bang-Jensen and Huang [80] considered kings in quasi-transitive digraphs. The main result of [80] is the following.

Theorem 2.10.9 *Let D be a quasi-transitive digraph. Then we have*

(1) D has a 3-king if and only if it has a finite out-radius[5]*.*
(2) If D has a 3-king, then the following holds:

(a) Every vertex in D of maximum out-degree is a 3-king.
(b) If D has no vertex of in-degree zero, then D has at least two 3-kings.
(c) If the unique initial strong component of D contains at least three vertices, then D has at least three 3-kings.

□

In the following family of quasi-transitive digraphs, every digraph has a 3-king but no 2-king: $\vec{C}_3[\overline{K}_{k_1}, \overline{K}_{k_2}, \overline{K}_{k_3}]$ for every $k_1, k_2, k_3 \geq 2$.

In [605], Petrovic and Thomassen obtained the following.

Theorem 2.10.10 *Let G be an undirected graph whose complement is the disjoint union of complete graphs, paths and cycles. Then every orientation of G with at most one vertex of in-degree zero has a 6-king.* □

2.11 Application: The One-Way Street and the Gossip Problems

In this section, we show how (some extensions of) the one-way and gossip problems lead one to consider minimum diameter orientations of digraphs. Recall that an orientation of a digraph D is a subdigraph of D obtained from D by deleting exactly one arc between x and y for every pair $x \neq y$ of vertices such that both xy and yx are in D. Some results are given on minimum diameter orientations of digraphs from well-known classes, semicomplete bipartite digraphs and quasi-transitive digraphs.

[5] See Proposition 2.1.1.

2.11.1 The One-Way Street Problem and Orientations of Digraphs

Graph theoretical modelling of the one-way problem can be traced back to the classical paper of Robbins [637]. It is well-known that introduction of one-way streets usually decreases the number of car accidents and allows one to simplify the traffic control. By Robbins' theorem (see Theorem 1.6.2) a connected graph G has a strongly connected orientation if and only if G has no bridge. This theorem shows when the one-way street system can be introduced. One reason why one-way streets are not used everywhere is that the travelling distances after such arrangements will increase. To minimize this disadvantage of the one-way traffic system, we may choose certain assignments of directions that minimize some disadvantage criterion. Three such criteria are discussed by Roberts and Xu [638, 639, 640, 641] Most other papers on the topic deal only with one criterion: the minimization of the longest path that has to be travelled, i.e. the diameter of an orientation of the undirected graph representing the street configuration. We restrict ourselves to this objective function.

Practically all papers on the topic consider orientations of undirected graphs. This corresponds to converting all streets, which were initially two-way, into one-way streets (see, e.g., Koh and Tay [492, 493, 495], König, Krumme and Lazard [500] and Plesník [608]). This model is quite restricted: certain streets may already be one-way. To take such streets into consideration, one has to study orientations of directed rather than undirected graphs. While there are a few papers (see, e.g., Boesch and Tindell [120], Chung, Garey and Tarjan [157], and Volkmann [730]) dealing with finite diameter orientations of digraphs, we are aware of only one paper [378] devoted to minimizing the diameter of an orientation of a digraph. In particular, the following results are proved by Gutin and Yeo [378]. For a digraph D, as in the case of undirected graphs, let $\mathrm{diam}_{\min}(D)$ denote the minimum diameter of an orientation of D.

Theorem 2.11.1 *If D is a strong quasi-transitive digraph of order $n \geq 3$, then*

$$\mathrm{diam}_{\min}(D) \leq \max\{3, \mathrm{diam}(D)\}.$$

There is an infinite family $\mathcal{Q}$ of strong quasi-transitive digraphs such that for every $Q \in \mathcal{Q}$, $\mathrm{diam}(Q) = 2$ but no orientation of Q is of diameter[6] *2.* □

Theorem 2.11.2 *If D is a strong semicomplete bipartite digraph of order $n \geq 4$ such that $D \neq \overleftrightarrow{K}_{1,n-1}$, then $\mathrm{diam}_{\min}(D) \leq \max\{5, \mathrm{diam}(D)\}$. There is an infinite family $\mathcal{B}$ of strong semicomplete bipartite digraphs such that for every $B \in \mathcal{B}$, $\mathrm{diam}(B) = 4$ but $\mathrm{diam}_{\min}(B) = 5$.* □

[6] Observe that by Exercise 1.57 every strong quasi-transitive digraph of order $n \geq 3$ has a strong orientation. So does every strong semicomplete bipartite digraph with every partite set of cardinality at least 2. On the other hand, $\overleftrightarrow{K}_{1,n-1}$, $n \geq 2$, has no strong orientation.

The sharpness of the upper bounds of these theorems can be seen from the following examples. Let T_k, $k \geq 3$, be a (transitive) tournament with vertices $x_1, x_2, \ldots, x_k$ and arcs $x_i x_j$ for every $1 \leq i < j \leq k$. Let y be a vertex not in T_k, which dominates all vertices of T_k but x_k and is dominated by all vertices of T_k but x_1. (See Figure 2.9.) The resulting semicomplete digraph D_{k+1} has diameter 2. However, the deletion of any arc of D_{k+1} between y and the set $\{x_2, x_3, \ldots, x_{k-1}\}$ leaves a digraph with diameter 3. Indeed, if we delete yx_i ($x_i y$), $2 \leq i \leq k-1$, then a shortest (x_k, x_i)-path ((x_i, x_1)-path) becomes of length 3.

Figure 2.9 A semicomplete digraph of diameter 2 with no orientation of diameter 2.

Let H be a strong semicomplete bipartite digraph with the following partite sets V_1 and V_2 and arc set A: $V_1 = \{x_1, x_2, x_3\}$, $V_2 = \{y_1, y_2, y_3\}$, and

$$A = \{x_1y_1, y_1x_1, x_1y_2, y_3x_1, x_2y_1, y_2x_2, y_3x_2, y_1x_3, x_3y_3, x_3y_2\}.$$

Let $H' = H - x_1y_1$ and $H'' = H - y_1x_1$. It is easy to verify that $\mathrm{diam}(H) = 4$ (in particular, $\mathrm{dist}(y_2, y_3) = 4$) and that $\mathrm{diam}(H') = \mathrm{diam}(H'') = 5$ (a shortest (x_1, y_3)-path in H' and a shortest (y_2, x_1)-path in H'' are of length 5). The digraph H can be used to generate an infinite family of semicomplete bipartite digraphs with the above property: replace x_3, say, by a set of independent vertices.

2.11.2 The Gossip Problem

'There are n ladies, and each one of them knows an item of scandal which is not known to any of the others. They communicate by telephone, and whenever two ladies make a call, they pass on to each other, as much scandals as they know at the time. How many calls are needed before all ladies know every scandal?' This is the way the so-called gossip problem (apparently due to A. Boyd) was stated by Hajnal, Milner and Szemerédi [392] in 1972. Since then numerous research papers on the topic have been published (see e.g.

surveys Fraigniaud and Lazard [248], Hedetniemi, Hedetniemi and Liestman [409], Hromkovič, Klasing, Monien and Peine [433]). The main reason of this popularity is a high applicability of the gossip problem, especially in computer networks.

Actually the above quotation captures only a special case of the gossip problem. In a more general setting, this problem can be formulated as follows. Let G be a connected graph of order n. Every vertex v of G holds initially an item $I(v)$ (different from the items of other vertices). A vertex v can pass all items it currently has to all or some of its neighbours at one step. The aim is to calculate the minimum number of steps required to pass to every vertex u the set $\{I(v) : v \in V(G)\}$ of all items.

The problem can be specified by allowing only one-way communications (like in radio communications over one frequency or email) when at every given step, for every pair u, v of adjacent vertices, either u can pass all items it holds to v, or v can pass all items it holds to u, but not both [248]. This specification is often called **half-duplex**. The half-duplex gossip problem is $\mathcal{NP}$-hard [248]. On the other hand, this problem is normally of interest, from the applications point of view, only for some special families of graphs such as the Cartesian products of cycles used in practice to build the Intel Δ-prototype (see Rattner [622]) and many transputer-based machines (see May [557]). Several important families of graphs are discussed by Fraigniaud and Lazard [248]. The solutions obtained for them are based on an upper bound that includes, as the main term, the minimum diameter of an orientation of a given undirected graph [248].

In the half-duplex gossip problem, we may consider symmetric digraphs $\overleftrightarrow{G}$ instead of undirected graphs G. The half-duplex model can be extended from symmetric to arbitrary digraphs D, where a vertex v can pass all its items only to vertices u such that vu is an arc in D. The use of arbitrary digraphs may well be of interest when security concerns dictate that some of the directions of communications are forbidden.

We consider only the half-duplex model for a strong digraph D. Let $s(G)$ stand for the minimum number of steps for gossiping in this model. Since the minimum number of steps to pass all items of vertex u to another vertex v is $\mathrm{dist}(u, v)$, we have $s(D) \geq \mathrm{diam}(D)$.

Gutin and Yeo [378] proved the following simple upper bound on $s(D)$, which is an improvement on the similar upper bound in [248] even in the case of symmetric digraphs.

Theorem 2.11.3 *Let $D = (V, A)$ be a strong digraph. Then*

$$s(D) \leq \min\{2\,\mathrm{rad}(D), \mathrm{diam}_{\min}(D)\}.$$

Proof: Let H be an orientation of D of minimum diameter. Let every vertex in D pass its items to all out-neighbours in H. Repeat this iteration till every vertex holds all items. Clearly, the number of iterations required is the length of the longest path in H, i.e. $s(D) \leq \mathrm{diam}(H) = \mathrm{diam}_{min}(D)$.

Let x be a vertex of D such that $\mathrm{rad}(D) = (\mathrm{dist}(x, V) + \mathrm{dist}(V, x))/2$. Let F_x^+ (F_x^-) be a BFS tree of D rooted at x (the converse of a BFS tree of the converse of D rooted at x). In the first $\mathrm{dist}(V, x)$ steps pass items from vertices to their out-neighbours along arcs of F_x^-. Thus, in the end, x holds all items. During the next $\mathrm{dist}(x, V)$ steps pass items from vertices to their out-neighbours along arcs of F_x^+. Hence, in the end, every vertex holds all items. Thus, $s(D) \leq 2\,\mathrm{rad}(D)$. □

The bound of Theorem 2.11.3 is of special interest when D satisfies $\mathrm{diam}(D) = \mathrm{diam}_{\min}(D)$. In this case, a minimum diameter orientation of D provides an optimal solution to the gossip problem. Thus, an orientation H of diameter possibly exceeding $\mathrm{diam}(D)$ by a small constant leads to a good approximate solution for the gossip problem (given H, the upper bound $\min\{2\mathrm{rad}(D), \mathrm{diam}(H)\}$ for $g(D)$ can be computed in polynomial time). In the previous subsection, we saw that slight modifications of $\mathrm{diam}(D) = \mathrm{diam}_{\min}(D)$ hold for some important families of digraphs.

2.12 Application: Exponential Neighbourhood Local Search for the TSP

The aim of this section is to introduce a new approach to obtain near optimal solutions for the travelling salesman problem (TSP). The main idea is to find, in polynomial time, a best solution in a specially constructed set of solutions of exponential cardinality. This idea can be applied not only to the TSP, but also to other $\mathcal{NP}$-hard combinatorial optimization problems. This general idea was used already in the papers by Sarvanov and Doroshko [651, 652] and Gutin [354].

2.12.1 Local Search for the TSP

The TSP is stated as follows. Given a weighted complete digraph $(\overleftrightarrow{K}_n, c)$, find a hamiltonian cycle in $\overleftrightarrow{K}_n$ of minimum cost. In this section and some others where the TSP is considered, we will often call a hamiltonian cycle in $\overleftrightarrow{K}_n$ a **tour**; it is also assumed that $V(\overleftrightarrow{K}_n) = \{1, 2, \ldots, n\}$. The TSP is a well-studied $\mathcal{NP}$-hard problem with numerous applications (see, e.g., the books by Cook, Cunningham, Pulleyblank and Schrijver [166], Lawler, Lenstra, Rinooykan and Shmoys [511], Reinelt [632] and the paper [466] by Johnson and McGeoch). Since the TSP is $\mathcal{NP}$-hard, no polynomial time exact algorithms to solve the problem are known. However, there is a well-tested approach (see, e.g., Johnson and McGeoch [466]) that provides near optimal solutions (which is sufficient in most applications) in reasonable time for large-scale instances of the TSP. The approach consists of two phases. In the first phase, a construction heuristic quickly produces a solution which is normally far from optimal but is much better than a random solution[7]. (Some construction heuristics for the TSP are described later in this book.) In the second phase, a local search heuristic is used. During every iteration of this heuristic, a neighbourhood of a current best solution is considered and a better solution (in certain cases, the best solution in the neighbourhood) is found. When no better solution in the neighbourhood exists the heuristic terminates. (There are several variations of the above description [466].)

In many cases, so-called 3-Opt is applied. In 3-Opt, the neighbourhood of a hamiltonian cycle C consists of all tours in $\overleftrightarrow{K}_n$ obtained from C by deleting three arcs and then adding three arcs. (This notion can be easily generalized to k-Opt for every fixed $k \geq 3$.) The cardinality of this neighbourhood is certainly $\Theta(n^3)$. Also, $O(n^3)$ time is required to completely search this neighbourhood (i.e., to find the best hamiltonian cycle) if we look at the tours of the neighbourhood one by one. Certainly, the cubic time is unacceptable for large-scale instances of the TSP. However, 3-Opt is widely used in practice since usually only a small fraction of the neighbourhood is searched before a better solution is found. Despite the fact that 3-Opt allows one to find quite good solutions to large-scale instances of the TSP, the way of looking at the solutions one by one seems rather inefficient.

Therefore, in the 1980's, Sarvanov and Doroshko [651, 652] and Gutin [354] introduced independently some neighbourhoods of exponential size where the best solution can be obtained in polynomial time. Recently, various neighbourhoods of exponential size for the TSP were suggested and investigated (see, e.g., Balas and Simonetti [37], Burkard, Deineko and Woeginger [137], Glover [318], Glover and Punnen [320], Potts and Velde [611] and Punnen [616]). The paper [188] by Deineko and Woeginger is an excellent survey on the topic. Balas and Simonetti [37] and Carlier and Villon [448] constructed and implemented local search algorithms which use exponential neighbourhoods. Their results are very encouraging. They also show the necessity of further theoretical research on the topic.

There are different types of neighbourhoods for the TSP; many of them can be found in [188, 466]. The following definition of a neighbourhood structure for the TSP is due to Deineko and Woeginger [188]. In this definition, we assume that every tour $T = \pi(1)\pi(2)\dots\pi(n)\pi(1)$ starts from the vertex 1, i.e., $\pi(1) = 1$. Therefore, we will identify T with the permutation $\pi(1)\pi(2)\dots\pi(n)$. A **neighbourhood structure** consists of a neighbourhood $N(T)$ for every tour T such that the neighbourhood $N(\pi(1)\pi(2)\dots\pi(n)) = \pi * N(12\dots n)$, where $\pi(1) = 1$ and $*$ stands for the permutation product (applied from right to left). This definition is somewhat restrictive (for example, it requires the cardinality of neighbourhoods to be the same) but reflects the very important 'shifting' property of neighbourhoods which distinguish them

[7] For certain families of instances of the TSP, some construction heuristics produce near optimal tours by themselves; see, e.g., Glover, Gutin, Yeo and Zverovich [319]. In such cases local search is perhaps not required.

from arbitrary sets of tours. Another important property of neighbourhood $N(T)$ of a tour T, which is usually imposed, is that the best tour of $N(T)$ can be computed in time polynomial in n. This is necessary to guarantee an efficient local search. Neighbourhoods satisfying this property are called **polynomially searchable**.

The largest known polynomially searchable neighbourhoods are those of size $2^{\Theta(n \log n)}$ (note that there are $(n-1)!$ tours in $\overleftrightarrow{K}_n$ and $(n-1)! = 2^{\Theta(n \log n)}$ as well). Such neighbourhoods were introduced independently in [354, 616, 652]. Punnen's neighbourhoods [616] are the most general among them. We will consider a special family of these neighbourhoods, which is a slight generalization of neighbourhoods in [354, 652]. We call those neighbourhoods the **assignment neighbourhoods**. (See Subsection 2.12.3 for the definition of these neighbourhoods.) Some features of these neighbourhoods were investigated in [369]. Gutin [369] proved that, for every $\beta > 0$, there is a neighbourhood of cardinality $2^{\Theta(n \log n)}$ that can be searched in time $O(n^{1+\beta})$. Deineko and Woeginger [188] demonstrated that to search a neighbourhood of cardinality $2^{\Theta(n \log n)}$ one needs time $\Omega(n^{1+\beta})$, where $\beta > 0$.

Since the diameter of neighbourhood structure digraphs (defined later) is of certain importance for local search, this parameter has also been studied. We present some recent results on the topic in Subsection 2.12.4.

2.12.2 Linear Time Searchable Exponential Neighbourhoods for the TSP

In this section, we demonstrate how to use the algorithm from Subsection 2.3.2 to search some exponential neighbourhoods. We introduce a neighbourhood of exponential size based on one of the approaches described by Glover and Punnen [320]. Assume that n, the order of $(\overleftrightarrow{K}_n, c)$, equals one modulo three (it is easy to see how to modify our approach when n does not equal one modulo three). Let $C = vv_1^0v_1^1v_1^2v_2^0v_2^1v_2^2 \dots v_t^0v_t^1v_t^2v$ be a hamiltonian cycle of $\overleftrightarrow{K}_n$. Define a neighbourhood of C as follows:

$$NB(C) = \{vv_1^{s_1}v_1^{s_1+1}v_1^{s_1+2} \dots v_t^{s_t}v_t^{s_t+1}v_t^{s_t+2}v : \; s_i \in \{0,1,2\}, i = 1,2,\dots,t\},$$

where all superscripts are taken modulo three. Clearly, $|NB(C)| = 3^{\lfloor n/3 \rfloor}$.

We show how to find the best hamiltonian cycle in $NB(C)$ in time $O(n)$. Construct an auxiliary weighted digraph $D = (V, A, w)$ as follows:

$$V = \{p, q, u_i^0, u_i^1, u_i^2 : \; i \in \{1,2,\dots,t\}\},$$

$$A = \{pu_1^j, u_t^k q, u_i^j u_{i+1}^k : \; j,k \in \{0,1,2\}, \; i \in \{1,2,\dots,t-1\}\}.$$

Moreover, $w(p, u_1^j) = c(v, v_1^j) + c(v_1^j, v_1^{j+1}) + c(v_1^{j+1}, v_1^{j+2})$ for every $j \in \{0,1,2\}$,

$$w(u_i^j, u_{i+1}^k) = c(v_i^{j+2}, v_{i+1}^k) + c(v_{i+1}^k, v_{i+1}^{k+1}) + c(v_{i+1}^{k+1}, v_{i+1}^{k+2})$$

for every $i \in \{1, 2, \ldots, t-1\}$, $j, k \in \{0, 1, 2\}$, and $w(u_t^k, q) = c(v_t^{k+2}, v)$ for every $k \in \{0, 1, 2\}$.

All (p, q)-paths in D are of the form $pu_1^{s_1} u_2^{s_2} \ldots u_t^{s_t} q$, where $s_i \in \{0, 1, 2\}$, $i = 1, 2, \ldots, t$. Therefore, the mapping

$$\phi: \; pu_1^{s_1} u_2^{s_2} \ldots u_t^{s_t} q \rightarrow vv_1^{s_1} v_1^{s_1+1} v_1^{s_1+2} \ldots v_t^{s_t} v_t^{s_t+1} v_t^{s_t+2} v$$

is a bijection from the set of (p, q)-paths in D into $NB(C)$. Moreover, for every (p, q)-path R in D, we have $w(R) = c(\phi(R))$. Hence, to find a minimum weight hamiltonian cycle in $NB(C)$, it suffices to compute a shortest (p, q)-path in D. This can be done in time $O(|A|) = O(n)$ by the algorithm described in Subsection 2.3.2. Moreover, since we can readily give an acyclic ordering of vertices in D, we do not need the first phase of the algorithm in Subsection 2.3.2.

2.12.3 The Assignment Neighbourhoods

The purpose of this subsection is to introduce the assignment neighbourhoods.

Let $C = x_1 x_2 \ldots x_k x_1$ be a cycle in $\overset{\leftrightarrow}{K}_n$. The operation of **removal** of a vertex x_i ($1 \leq i \leq k$) results in the cycle $x_1 x_2 \ldots x_{i-1} x_{i+1} \ldots x_k x_1$ (thus, removal of x_i is not deletion of x_i from C). Let y be a vertex of $\overset{\leftrightarrow}{K}_n$ not in C. The operation of **insertion** of a vertex y into an arc $x_i x_{i+1}$ results in the cycle $x_1 x_2 \ldots x_i y x_{i+1} \ldots x_k x_1$. An **insertion** of y into C is an insertion of y into $x_i x_{i+1}$ for some $1 \leq i \leq k$. For a set $Z = \{z_1, \ldots, z_s\}$ ($s \leq k$) of vertices not in C, an **insertion** of Z into C means an insertion of z_1 in C followed by an insertion of z_2 into the obtained cycle, etc. Furthermore, we require that, in the cycle obtained after insertion of all vertices of Z into C, *no pair of vertices from Z is adjacent*.

Let $T = x_1 x_2 \ldots x_n x_1$ be a tour in $\overset{\leftrightarrow}{K}_n$ and let $Z = \{x_{i_1}, x_{i_2}, \ldots, x_{i_s}\}$ be a set of pairwise non-adjacent vertices of T. The **assignment neighbourhood** $N(T, Z)$ of T with respect to Z consists of the tours that can be obtained from T by removal of the vertices from Z one by one followed by an insertion of Z into the cycle derived after the removal. For example, for $H = x_1 x_2 x_3 x_4 x_5 x_1$, $N(H, \{x_1, x_3\}) = \{x_2 x_i x_4 x_j x_5 x_2, x_2 x_i x_4 x_5 x_j x_2, x_2 x_4 x_i x_5 x_j x_2 : \; \{i, j\} = \{1, 3\}\}$. Let $T = x_1 x_2 \ldots x_n x_1$ and $s = |Z|$; then it is easy to verify that

$$|N(T, Z)| = (n-s)!/(n-2s)!$$

(clearly, we have $n - s \geq s$).

We show that the best tour in $N(T, Z)$ can be found in time $O(n^3)$ [369, 616]. Let $C = y_1 y_2 \ldots y_{n-s} y_1$ be the cycle obtained from T after removal of Z and let $Z = \{z_1, z_2, \ldots, z_s\}$. Let ϕ be an injective mapping from Z to $Y = \{y_1, y_2, \ldots, y_{n-s}\}$. If we insert some z_i into an arc $y_j y_{j+1}$, then the weight of C will be increased by $c(y_j z_i) + c(z_i y_{j+1}) - c(y_j y_{j+1})$. Therefore, if we insert every z_i, $i = 1, 2, \ldots, s$, into $y_{\phi(i)}y_{\phi(i)+1}$, the weight of C will be increased by

$$f(\phi) = \sum_{i=1}^{s}(c(y_{\phi(i)}z_i) + c(z_i y_{\phi(i)+1}) - c(y_{\phi(i)}y_{\phi(i)+1})).$$

Clearly, to find a tour of $N(T, Z)$ of minimum weight, it suffices to minimize $f(\phi)$ on the set of all injections ϕ from Z to Y. This can be done using the following weighted complete bipartite graph B. The partite sets of B are Z and Y, and the weight of an edge $z_i y_j$ is set to be $c(y_j z_i) + c(z_i y_{j+1}) - c(y_j y_{j+1})$.

By the definition of B, every maximum matching M of B corresponds to an injection ϕ_M from Z to Y. Moreover, the weights of M and ϕ_M coincide. A minimum weight maximum matching in B can be found by solving the assignment problem (see Section 3.12). Therefore, in $O(n^3)$ time, we can find the best tour in $N(T, Z)$.

2.12.4 Diameters of Neighbourhood Structure Digraphs for the TSP

Given a neighbourhood $N(T)$ for every tour T in $\overset{\leftrightarrow}{K}_n$ (i.e., some **neighbourhood structure**), the corresponding **neighbourhood digraph** (of order $(n-1)!$) is a directed graph with vertex set consisting of all tours in $\overset{\leftrightarrow}{K}_n$ and arc set of pairs (T', T'') such that $T'' \in N(T')$. When all neighbourhoods $N(T)$ are polynomially searchable, the corresponding digraph is **polynomially searchable**. The diameter of the neighbourhood digraph is one of the most important characteristics of the neighbourhood structure and the corresponding local search scheme [188, 318, 448]. Clearly, a neighbourhood structure with a neighbourhood digraph of smaller diameter seems to be more useful than that with a neighbourhood digraph of larger diameter, let alone a neighbourhood structure whose digraph has infinite diameter (in the last case, some tours are not 'reachable' from the initial tour during local search procedure).

For example, the neighbourhood digraph for polynomially searchable 'pyramidal' neighbourhoods introduced by Carlier and Villon [448] has diameter $d_n = \Theta(\log n)$. (In [448], it was proved that $d_n \le \log n$, the lower bound $d_n = \Omega(\log n)$ follows from the facts that the cardinality of pyramidal neighbourhoods is $2^{\Theta(n)}$ [448] and the total number of tours is $2^{\Theta(n \log n)}$.)

In this subsection, using the assignment neighbourhoods, we construct certain polynomially searchable 'compound' neighbourhoods whose digraphs have diameter bounded by a small constant. We follow the presentation of Gutin and Yeo [375].

For a positive integer $k \le n/2$, the neighbourhood digraph $\Gamma(n, k)$ has a vertex set formed by all tours in $\overset{\leftrightarrow}{K}_n$. In $\Gamma(n, k)$, a tour T dominates a tour R if there exists a set Z of k non-adjacent vertices of T such that $R \in N(T, Z)$. Clearly, T dominates R if and only if R dominates T, i.e., $\Gamma(n, k)$ is symmetric. We denote by $\mathrm{dist}_k(T, R)$ the distance from T to R in $\Gamma(n, k)$.

For a tour T in $\overleftrightarrow{K}_n$, let $\mathcal{I}_{nk}$ denote the family of all sets of k non-adjacent vertices in T. Clearly, the neighbourhood $N_k(T)$ of a tour T in $\Gamma(n, k)$ equals

$$\cup_{Z \in \mathcal{I}_{nk}} N(T, Z).$$

Thus if, for some k, $i(n, k) = |\mathcal{I}_{nk}|$ is polynomial in n, then since $N(T, Z)$ is polynomially searchable $\Gamma(n, k)$ is polynomially searchable. Otherwise, $\Gamma(n, k)$ may be non-polynomially searchable. Since polynomially searchable $\Gamma(n, k)$ are our main interest, we start by evaluating $i(n, k)$ in Theorem 2.12.1. It follows from Theorem 2.12.1 that, for fixed k, $i(n, k)$ and $i(n, n-k)$ are polynomial.

Theorem 2.12.1 *[375]* $i(n, k) = \binom{n-k}{k} + \binom{n-k-1}{k-1}$. □

Corollary 2.12.2 *If p is a non-negative fixed integer ($p < \lfloor n/2 \rfloor$), then $\Gamma(n, p+1)$ and $\Gamma(n, \lfloor (n-p)/2 \rfloor)$ are polynomially searchable.*

Proof: This follows from Theorem 2.12.1 and $\binom{m}{k} = \binom{m}{m-k}$. □

It can be shown (Exercise 2.47) that for n is even $\Gamma(n, n/2)$ consists of an exponential number of strongly connected components and, thus, its diameter is infinite (for example, $x_1x_2 \dots x_nx_1$ and $x_1 \dots x_{n-2}x_nx_{n-1}x_1$ belong to different strong components of this digraph). Therefore, below we consider $\Gamma(n, k)$ for $k < n/2$ only.

Theorem 2.12.3 (Gutin and Yeo) *[375]* $\mathrm{diam}(\Gamma(n, \lfloor (n-1)/2 \rfloor)) \leq 4$.

Proof: We assume that $n \geq 5$, as for $2 \leq n \leq 4$ this claim can be verified directly. Let $C = x_1x_2 \dots x_nx_1$ and $T = y_1y_2 \dots y_ny_1$ be a pair of distinct tours in $\overleftrightarrow{K}_n$. Put $k = \lfloor (n-1)/2 \rfloor$. We will prove that $\mathrm{dist}_k(T, C) \leq 4$, thus showing that $\mathrm{diam}(\Gamma(n, k)) \leq 4$.

We call a vertex v **even** (**odd**) with respect to C if $v = x_j$, where $1 \leq j \leq n$ and j is even (odd). For a set of vertices X of $\overleftrightarrow{K}_n$, let X_{odd} (X_{even}) be the set of odd (even) vertices in X.

First we consider the case of even n, i.e. $k = n/2 - 1$. The proof in this case consists of two steps. At the first step, we show that there exists a tour T'' whose vertices alternate in parity and such that $\mathrm{dist}_k(T, T'') \leq 2$. Moreover, T'' has a pair of consecutive vertices which are also consecutive in C. At the second step, we will see that $\mathrm{dist}_k(T'', C) \leq 2$ as the odd and even vertices of T'' (except for the vertices of the above pair) can be separately reordered to form C. Thus, we will conclude that $\mathrm{dist}_k(T, C) \leq 4$. Now, we proceed to the proof.

Clearly, T has a pair y_j, y_{j+1} such that y_{j+1} is odd and y_j is even. Let

$$Z = \{y_{j+2}, y_{j+4}, \dots, y_{j+2k}\}$$

and let $|Z_{odd}| = s$. Remove the vertices of Z from T and then insert the s odd vertices of Z into the arcs $y_{j+1}y_{j+3}, \dots, y_{j+2s-1}y_{j+2s+1}$ and $k-s$ even vertices of Z into the arcs $y_{j+2s+1}y_{j+2s+3}, y_{j+2s+3}y_{j+2s+5}, \dots, y_{j+2k-1}y_{j+2k+1}$. We have obtained a tour

$$T' = y_j y_{j+1} v_{j+2} y_{j+3} v_{j+4} y_{j+5} \dots y_{j+2k-1} v_{j+2k} y_{j+2k+1} y_j,$$

where $\{v_{j+2}, \dots, v_{j+2k}\} = Z$.

Let $Z' = \{y_{j+3}, y_{j+5}, \dots, y_{j+2k+1}\}$ and let $|Z'_{even}| = t$. Since the number of odd vertices in $V(\overleftrightarrow{K}_n) - \{y_j, y_{j+1}\}$ is equal to $k = |Z_{odd}| + |Z'_{odd}| = s+k-t$, we obtain that $s = t$. Remove Z' from T' and insert the t even vertices of Z' into the arcs $y_{j+1}v_{j+2}, v_{j+2}v_{j+4}, v_{j+6}v_{j+8}, \dots, v_{j+2s-2}v_{j+2s}$ and the $k-s$ odd vertices of Z' into the arcs $v_{j+2s+2}v_{j+2s+4}, \dots, v_{j+2k-2}v_{j+2k}, v_{j+2k}y_j$. We have derived a tour $T'' = u_1u_2 \dots u_nu_1$. Clearly, the vertices of T'' alternate in parity, i.e., for every m, if u_m is odd, then u_{m+1} is even.

Now we prove that the processes of insertion of Z and Z' can be performed in such a way that T'' contains a pair of consecutive vertices which are also consecutive in C (i.e. there exist indices p and q such that $u_p = x_q$ and $u_{p+1} = x_{q+1}$). Since $1 < |Z'| < n$, there exists a pair of distinct indices i, m such that $x_i, x_m \in Z'$ and $x_{i+1}, x_{m-1} \notin Z'$. Without loss of generality, we assume that i is odd. We consider two cases.

Case 1: $|Z'_{odd}| \geq 2$. We prove that we may choose index $q = i$. Since $x_{i+1} \notin Z'$ and $i+1$ is even, either $y_j = x_{i+1}$ or $x_{i+1} \in Z_{even}$. If $x_{i+1} \in Z_{even}$, in the process of insertion of Z, we insert x_{i+1} into $y_{j+2k-1}y_{j+2k+1}$, i.e. $x_{i+1} = v_{j+2k}$. In the process of insertion of Z', we insert x_i into $v_{j+2k}y_j$ if $x_{i+1} = y_j$ or into $v_{j+2k-2}v_{j+2k}$, otherwise (i.e. $x_{i+1} = v_{j+2k}$).

Case 2: $|Z'_{odd}| = 1$. Thus, m is even. Since $n \geq 6$, it follows that $|Z'_{even}| \geq 2$. Analogously to Case 1, one may take $q = m - 1$.

Therefore, without loss of generality, we assume that $u_{n-1} = x_i,\ u_n = x_{i+1}$. Since $\{u_2, u_4, \dots, u_{2k}, x_{i+1}\} = C_{even}$, we can delete $\{u_2, u_4, \dots, u_{2k}\}$ from T'' and insert it into the obtained cycle to get the tour C' given by $C' = u_1x_{i+3}u_3x_{i+5}u_5 \dots u_{2k-1}x_{i-1}u_{n-1}x_{i+1}u_1$. Analogously, we can delete $\{u_1, u_3, \dots, u_{2k-1}\}$ from C' and insert it into the obtained cycle to get C. We conclude that $\text{dist}_k(T, C) \leq 4$.

Now let n be odd; then $k = (n-1)/2$. Notice that, without loss of generality, we may assume that $x_n = y_n$ (to fix the initial labellings of T and C). Consider tours $X = x_1x_2 \dots x_nx_{n+1}x_1$ and $Y = y_1y_2 \dots y_{n-1}y_ny_{n+1}y_1$ in $\overleftrightarrow{K}_{n+1}$, where $y_n = x_n,\ y_{n+1} = x_{n+1}$. If we assume that $j = n$, $j+1 = n+1$, we can obtain, analogously to the case of even n, a tour Y'' such that the vertices of Y'' alternate in parity (with respect to their indices in X), x_{n+1} follows x_n in Y'' and $\text{dist}_k(Y, Y'') \leq 2$. Now if $i = n$ and $i+1 = n+1$, then we can show, similarly to the case of even n, that $\text{dist}_k(Y'', X) \leq 2$ and, thus, $\text{dist}_k(Y, X) \leq 4$. Notice that, in the whole process of constructing X from Y, we have never removed x_n and x_{n+1} or inserted any vertex into the arc $x_n x_{n+1}$. Thus, we could contract the arc $x_n x_{n+1}$ to x_n and obtain C from T in four 'steps'. This shows that $\text{dist}_k(T, C) \leq 4$. □

We can extend Theorem 2.12.3 using the following.

Theorem 2.12.4 *[375] Let* $\text{dist}_k(T, C) = 1$ *for tours* T *and* C *and let* m *be any integer smaller than* k*. Then,* $\text{dist}_m(T, C) \leq \lceil k/m \rceil$.

Corollary 2.12.5 *For every positive* $m \leq \lfloor (n-1)/2 \rfloor$*,*

$$\text{diam}(\Gamma(n, m)) \leq 4\lceil \lfloor (n-1)/2 \rfloor / m \rceil.$$

In particular, if p *is a positive fixed integer, then* $\text{diam}(\Gamma(n, \lfloor (n-p)/2 \rfloor)) \leq 8$ *provided* $n \geq 2p+1$.

Proof: The first inequality follows directly from the above two theorems and the triangle inequality for distances in graphs. It also implies the second one. Indeed, $n \geq 2p+1$ infers

$$\frac{(n-1)/2}{(n-p-1)/2} \leq 2 \text{ and so } Q = \frac{\lfloor (n-1)/2 \rfloor}{\lfloor (n-p)/2 \rfloor} \leq 2.$$

Hence, $\lceil Q \rceil \leq 2$. □

2.13 Exercises

2.1. Formulate the shortest (s, t)-path problem as a linear programming problem with integer variables. Hint: use a variable for each arc.

2.2. **(−)** Show by an example that a minimum weight out-branching with root s may not be a shortest path tree from s.

2.3. **(−)** Illustrate the shortest path algorithm for acyclic digraphs (Subsection 2.3.2) on the acyclic digraph in Figure 2.10.

2.4. **Finding the longest paths from a fixed vertex to all other vertices in a weighted acyclic digraph**. Develop a polynomial algorithm for finding the longest paths from a fixed vertex s to all other vertices in an arbitrary weighted acyclic digraph. Preferably your algorithm should run in linear time.

2.5. Find the longest paths from s to all other vertices in the acyclic digraph in Figure 2.10, e.g. using the algorithm that you designed in Exercise 2.4.

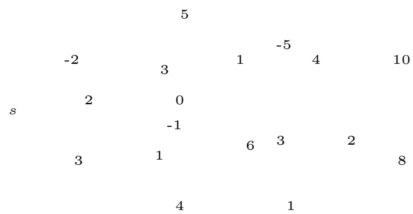


Figure 2.10 A weighted acyclic digraph.

2.6. **Finding a longest path in a weighted acyclic digraph in linear time**. Show how to find a longest path in a weighted acyclic digraph D in linear time. Hint: use a variant of the dynamic programming approach taken in (2.3), or construct a superdigraph D' of D such that one can read out a longest path in D from a shortest path tree from some vertex s in D'.

2.7. **(−)** Execute Dijkstra's algorithm on the digraph in Figure 2.11.

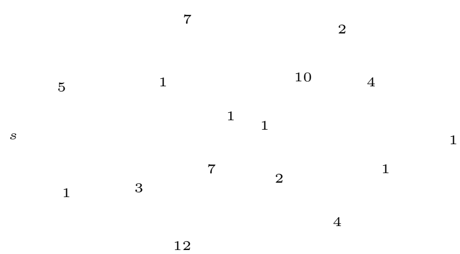


Figure 2.11 A digraph with non-negative weights on the arcs.

2.8. Complete the description of Dijkstra's algorithm in Subsection 2.3.3 such that not only the distances from s to the vertices of D are computed, but also the actual shortest paths are found.

2.9. Complete the description of the Bellman-Ford-Moore algorithm in Subsection 2.3.4 such that not only the distances from s to the vertices of D are computed, but also the actual shortest paths are found.

2.10. **(−)** Execute the Bellman-Ford-Moore algorithm on the digraph in Figure 2.12. Perform the scanning of arcs in lexicographic order.

2.11. **Negative cycle detection using the Bellman-Ford-Moore algorithm.** Prove Theorem 2.3.10.

2.12. Show how to detect a negative cycle in the digraph in Figure 2.13 using the extension of the Bellman-Ford-Moore algorithm.

2.13. Show by an example that Dijkstra's algorithm may not find the correct distances if it is applied to a weighted directed graph D where some arcs have negative weights, even if there is no negative cycle in D.

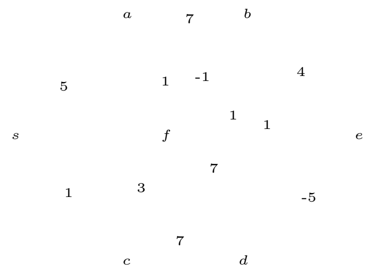


Figure 2.12 A digraph with weights on the arcs and no negative cycles.

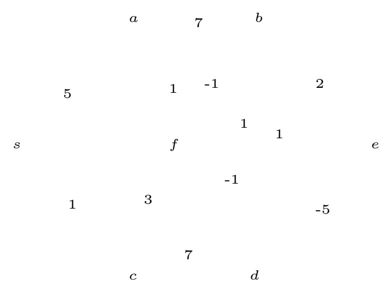


Figure 2.13 A weighted digraph with a negative cycle.

2.14. **(−)** Show how to implement the Floyd-Warshall algorithm so that it runs in time $O(n^3)$.

2.15. Prove Theorem 2.3.10.

2.16. **Re-weighting the arcs of a digraph**. Let $D = (V, A, c)$ be a weighted digraph and let $\pi : V \to \mathcal{R}$ be a function on the vertices of D. Define a new weight function c^* by $c^*(u, v) = c(u, v) + \pi(u) - \pi(v)$ for all $v \in V$. Let dist* be the distance function with respect to $D^* = (V, A, c^*)$, and let P be an (x, y)-path in D. Prove that P is a shortest (x, y)-path in D (with respect to c) if and only if P is a shortest (x, y)-path in D^* (with respect to c^*). Hint: consider what happens to the length of a path after the transformation above.

2.17. **(−)** Consider the weights introduced in Exercise 2.16. Show that the weight of a cycle in D is unchanged under the transformation from $D = (V, A, c)$ to $D^* = (V, A, c^*)$.

2.18. **Getting rid of negative weight arcs by re-weighting.** Let $D = (V, A, c)$ be a weighted digraph with some arcs of negative weight, but with no negative cycle. Let $D' = (V, A', c')$ be obtained from D by adding a new vertex s and all arcs of the form sv, $v \in V$, and setting $c'(s, v) = 0$ for all $v \in V$ and $c'(u, v) = c(u, v)$ for all $u, v \in V$. Let $\pi(v) = \mathrm{dist}_{D'}(s, v)$ for all $v \in V$. Define c^* by $c^*(u, v) = c(u, v) + \pi(u) - \pi(v)$ for all $u, v \in V$. Prove that $c^*(u, v) \geq 0$ for all $u, v \in V$.

2.19. **Johnson's algorithm for shortest paths.** Show that by combining the observations of Exercises 2.16-2.18, one can obtain an $O(n^2 \log n + nm)$ algorithm for the all pairs shortest path problem in digraphs with no negative cycles (Johnson [463]).

2.20. Let $M = [m_{ij}]$ be the adjacency matrix of a digraph $D = (V, A)$ with $V = \{1, 2, \ldots, n\}$ and let k be a natural number. Prove that there is an (i, j)-walk of length k in D if and only if the (i, j) entry of the kth power of M is positive.

2.21. Show how to compute the kth power of the adjacency matrix of a digraph of order n in time $O(P(n) \log n)$, where $P(n)$ is the time required to compute the product of two $n \times n$ matrices.

2.22. **Finding a shortest cycle in a digraph**. Describe a polynomial algorithm to find the shortest cycle in a digraph. Hint: use Exercise 2.20.

2.23. **(+) The generalized triangle-inequality.** An arc-weighted digraph $D = (V, A, c)$ satisfies the **generalized triangle-inequality** if, whenever P and Q are (x, y)-paths for some $x, y \in V(D)$ we have that $|A(P)| \leq |A(Q)|$ implies that $c(P) \leq c(Q)$. Describe a polynomial algorithm to check whether a given arc-weighted digraph satisfies the generalized triangle-inequality.

2.24. The generalized triangle-inequality was defined above. Show that one can find the shortest path from a given vertex to all other vertices in $O(n + m)$ time in a weighted digraph which satisfies the generalized triangle-inequality.

2.25. Prove Proposition 2.4.2.

2.26. **(−)** Draw the Goldberg digraph $D(12, 15)$ (see the proof of Theorem 2.4.4).

2.27. **(−)** Derive a formula for the maximum diameter of an orientation of the complete k-partite graph $K_{n_1,n_2,\ldots,n_k}$. Hint: apply Theorem 2.5.1.

2.28. **Short cycles through an edge**. Let $G = (V, E)$ be a 2-edge-connected graph and let $uv \in E$. Prove that G has a cycle of length at most $2\mathrm{dist}(u, V) + 1$ through the edge uv. Hint: use the (undirected) distance classes from u and v as well as the fact that uv is not a bridge.

2.29. **(−)** Let $G_1, G_2, \ldots, G_p$ be connected undirected graphs. Prove that

$$\mathrm{diam}(\Pi_{i=1}^{p} G_i) = \sum_{i=1}^{p} \mathrm{diam}(G_i).$$

2.30. Prove that $\rho(C_p \times C_q) > 0$ when both p and q are odd ($p, q \geq 3$) (West, see [500]).

2.31. Construct orientations of $P_3 \times P_6$ and $P_3 \times P_7$ of diameter 8.

2.32. Prove Theorem 2.9.5.

2.33. **(−)** For every odd number $n \geq 3$, give an example of a tournament T of order n, in which all vertices are 2-kings.

2.34. **(−)** Let T be a tournament on 4 vertices. Show that T contains a vertex which is not a 2-king.

2.35. Prove Theorem 2.10.1 (Moon [571]).

2.36. **(−)** Describe an infinite family of semicomplete digraphs, in which every member has exactly two 2-kings.

2.37. Prove that the tournament T_n in Subsection 2.10.1 has only three 2-kings for $n \geq 5$.

2.38. Prove that a multipartite tournament T has a finite out-radius if and only if T contains at most one vertex of in-degree zero. Hint: use Proposition 2.1.1.

2.39. **(−)** Characterize 2-kings in multipartite tournaments.

2.40. **3-kings in quasi-transitive digraphs**. Show that every quasi-transitive digraph of finite radius has a 3-king (Bang-Jensen and Huang [80]).

2.41. Prove Theorem 2.9.5.

2.42. Prove Theorem 2.12.1.

2.43. Prove Theorem 2.12.4.

2.44. Prove that, in the half-duplex model of gossiping (see Section 2.11), $s(G) \leq \mathrm{diam}(G) + 1$ for every connected bipartite graph G (Krumme, Cybenko and Venkataraman [504]).

2.45. Using the upper bound of the previous exercise, prove that $s(C_{2k}) = k + 1$ for every integer $k \geq 2$.

2.46. **(−)** Evaluate the cardinality of a neighbourhood in k-Opt for the TSP ($k \geq 3$).

2.47. **(−) Poor quality exponential neighbourhoods.** Show that, if n is even, then $\Gamma(n, n/2)$ (see Subsection 2.12.4) consists of an exponential number of strongly connected components and, thus, its diameter is infinite.

2.48. **(−)** Find the cardinality of the assignment neighbourhood $N(T, Z)$ for the TSP with n vertices and $k = |Z|$ (Gutin [369]).

2.49. **Maximizing exponential neighbourhoods.** Find the value of $k = |Z|$ for which the cardinality of the assignment neighbourhood $N(T, Z)$ for the TSP with n vertices is maximum (Gutin [369]).

3. Flows in Networks

The purpose of this chapter is to describe basic elements of the theory and applications of network flows. This topic is probably the most important single tool for applications of digraphs and perhaps even of graphs as a whole. At the same time, from a theoretical point of view, flow problems constitute a beautiful common generalization of shortest path problems and problems such as finding internally (arc)-disjoint paths from a given vertex to another. The theory of flows is well understood and fairly simple. This, combined with the enormous applicability to real-life problems, makes flows a very attractive topic to study. From a theoretical point of view, flows are well understood as far as the basic questions, such as finding a maximum flow from a given source to a given sink or characterizing the size of such a flow, are concerned. However, the topic is still a very active research field and there are challenging open problems such as deciding whether an $O(nm)$ algorithm[1] exists for the general maximum flow problem.

Several books deal almost exclusively with flows see e.g. the books [7] by Ahuja, Magnanti and Orlin, [199] by Dolan and Aldous, the classical text [246] by Ford and Fulkerson and [578] by Murty. In particular, [7] and [578] contain a wealth of applications of flows. In this chapter we can only cover a very small part of the theory and applications of network flows, but we will try to illustrate the diversity of the topic and show several applications of a practical as well as theoretical nature. Many of the results given in this chapter will be used in several other chapters such as those on connectivity and hamiltonian cycles.

3.1 Definitions and Basic Properties

A **network** is a directed graph $D = (V, A)$ associated with the following functions on $V \times V$: a **lower bound** $l_{ij} \geq 0$, a **capacity** $u_{ij} \geq l_{ij}$ and a **cost** c_{ij} for each $(i, j) \in V \times V$. These parameters satisfy the following requirement:

[1] Here and everywhere in this chapter n is the number of vertices and m the number of arcs in the network under consideration.

$$\text{For every } (i,j) \in V \times V, \text{ if } ij \notin A, \text{ then } l_{ij} = u_{ij} = 0. \tag{3.1}$$

In order to simplify notation in this chapter we also make the assumption that

$$c_{ij} = -c_{ji} \qquad \forall (i,j) \in V \times V. \tag{3.2}$$

This assumption may seem restrictive but it is purely a technical convention to make some of the following definitions simpler (in particular, the definition of costs in the residual network in Subsection 3.1.2). When it comes to implementing algorithms for various flow problems involving costs, this assumption can easily be avoided (Exercise 3.2). Finally we assume that if there is no arc between i and j (in any direction) then $c_{ij} = 0$.

In some cases we also have a function $b : V \to \mathcal{R}$ called a **balance vector** which associates a real number with each vertex of D. We will always assume that

$$\sum_{v \in V} b(v) = 0. \tag{3.3}$$

We use the shorthand notation $\mathcal{N} = (V, A, l, u, b, c)$ to denote a network with corresponding digraph $D = (V, A)$ and parameters l, u, b, c. If there are no costs specified, or there is no prescribed balance vector, then we omit the relevant letters from the notation. Note that whenever we consider a network $\mathcal{N} = (V, A, l, u, b, c)$ we also have a digraph, namely the digraph $D = (V, A)$ that we obtain from $\mathcal{N}$ by omitting all the functions l, u, b, c.

For a given pair of not necessarily disjoint subsets U, W of the vertex set of a network $\mathcal{N} = (V, A, l, u)$ and a function f on $V \times V$ we use the notation $f(U, W)$ as follows (here f_{ij} denotes the value of f on the pair (i, j)):

$$f(U, W) = \sum_{i \in U, j \in W} f_{ij}. \tag{3.4}$$

We will always make the realistic assumption that $n = O(m)$ which holds for all interesting networks. In fact, almost always, the networks on which we work will be connected as digraphs.

3.1.1 Flows and Their Balance Vectors

A **flow** in a network $\mathcal{N}$ is a function $x : A \to \mathcal{R}_0$ on the arc set of $\mathcal{N}$. We denote the value of x on the arc ij by x_{ij}. For convenience, we will sometimes think of x as a function of $V \times V$ and require that $x_{ij} = 0$ if $ij \notin A$ (see e.g. the definition of residual capacity in (3.7)). An **integer flow** in $\mathcal{N}$ is a flow x such that $x_{ij} \in \mathcal{Z}_0$ for every arc ij. For a given flow x in $\mathcal{N}$ the **balance vector of** x is the following function b_x on the vertices:

$$b_x(v) = \sum_{vw \in A} x_{vw} - \sum_{uv \in A} x_{uv} \qquad \forall v \in V. \tag{3.5}$$

That is, $b_x(v)$ is the difference between the flow on arcs with tail v and the flow on arcs with head v. We classify vertices according to their balance values (with respect to x). A vertex v is a **source** if $b_x(v) > 0$, a **sink** if $b_x(v) < 0$ and otherwise v is **balanced** ($b_x(v) = 0$). When there is no confusion possible (in particular when there is only one flow in question) we may drop the index x on b and say that b is the balance vector of x.

A flow x in $\mathcal{N} = (V, A, l, u, b, c)$ is **feasible** if $l_{ij} \leq x_{ij} \leq u_{ij}$ for all $ij \in A$ and $b_x(v) = b(v)$ for all $v \in V$. If no balance vector is specified for the network, then a feasible flow x is only required to satisfy $l_{ij} \leq x_{ij} \leq u_{ij}$ for all $(i, j) \in A$.

The **cost** of a flow x in $\mathcal{N} = (V, A, l, u, c)$ is given by

$$c^T x = \sum_{ij \in A} c_{ij} x_{ij}. \tag{3.6}$$

See Figure 3.1 for an example of a feasible flow.

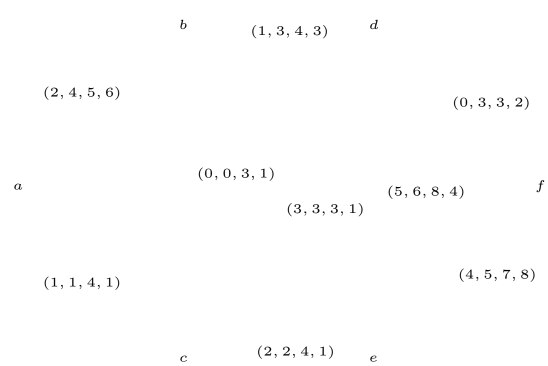


Figure 3.1 A network $\mathcal{N} = (V, A, l, u, c)$ with a feasible flow x specified. The specification on each arc ij is $(l_{ij}, x_{ij}, u_{ij}, c_{ij})$. The cost of the flow is 109.

We point out that whenever the lower bounds are all zero (an assumption that is not a restriction of the modeling power of flows as we shall see in Section 3.2) we will always assume that if iji is a 2-cycle of a network $\mathcal{N}$ and x is a flow in $\mathcal{N}$, then at least one of x_{ij}, x_{ji} is equal to zero. We call such a flow a **netto flow** in $\mathcal{N}$. The practical motivation for this restriction is that very often one uses flows to model items (water, electricity, telephone messages, etc.) that move from one place to another in time. Here it makes perfect sense to say that sending 3 units from i to j and 2 units from j to i is the same as sending 1 unit from i to j and nothing from j to i (we say that 2 of the units **cancel out**). In some of the definitions below it is easier to work with netto flows.

The notion of flows generalize that of paths in directed graphs. Indeed, if P is an (s,t)-path in a digraph $D=(V,A)$, then we can describe a feasible flow x in the network $\mathcal{N}=(V,A,l\equiv 0,u\equiv 1)$ by taking $x_{ij}=1$ if ij is an arc of P and $x_{ij}=0$ otherwise. This flow has balance vector

$$b_x(v)=\begin{cases}1 & \text{if } v=s\\ -1 & \text{if } v=t\\ 0 & \text{otherwise.}\end{cases}$$

We can also see that if there are weights on the arcs of D and we let $\mathcal{N}$ inherit these weights as costs on the arcs, then the cost of the flow defined above is equal to the length (weight) of P. Hence the shortest path problem is a special case of the minimum cost flow problem (which is studied in Section 3.10) with respect to the balance vector described above (here we implicitly used Theorem 3.3.1 for the other direction of going from a flow to an (s,t)-path in D.) In a very similar way we can also see that flows generalize cycles in digraphs. It is an important and very useful fact about flows that in some sense one can also go the other way. As we shall see in Theorem 3.3.1, every flow in a network with n vertices and m arcs can be decomposed into no more than $n+m$ flows along simple paths and cycles. Furthermore, paths and cycles play a fundamental role in several algorithms for finding optimal flows where the optimality is with respect to measures we define later.

3.1.2 The Residual Network

The concept of a residual network was implicitly introduced by Ford and Fulkerson [246].

For a given flow x in a network $\mathcal{N}=(V,A,l,u,c)$, define the **residual capacity** r_{ij} from i to j as follows:

$$r_{ij}=(u_{ij}-x_{ij})+(x_{ji}-l_{ji}). \tag{3.7}$$

The **residual network** $\mathcal{N}(x)$ with respect to x is defined as $\mathcal{N}(x)=(V,A(x),\tilde{l}\equiv 0,r,c)$, where $A(x)=\{ij : r_{ij}>0\}$. That is, the cost function is the same[2] as for $\mathcal{N}$ and all lower bounds are zero. See Figure 3.2 for an illustration.

The arcs of the residual network have a natural interpretation. If $ij\in A$ and $x_{ij}=5<7=u_{ij}$, then we may increase x by up to two units on the arc ij at the cost of c_{ij} per unit. Furthermore, if we also have $l_{ij}=2$ then we can also choose to decrease x by up to 3 units along the arc ij. The cost of this decrease is exactly $c_{ji}=-c_{ij}$ per unit. Note that a decrease of flow along the

[2] Note that this differs from definitions in other texts such as [7], but we can do this since we made the assumption (3.2).

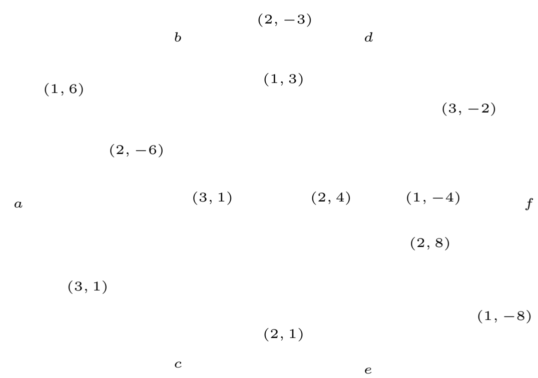


Figure 3.2 The residual network $\mathcal{N}(x)$ corresponding to the flow in Figure 3.1. The data on each arc is (r, c).

arc ij may also be thought of as sending flow in the opposite direction along the residual arc ji and then canceling out.

3.2 Reductions Among Different Flow Models

The purpose of this section is to show that one can restrict the general definition of a flow network considerably and still retain its modeling generality. We also show that one can model networks with lower bounds, capacities and costs on the vertices by networks, where all these numbers are on arcs only.

3.2.1 Eliminating Lower Bounds

We start with the following easy observation which shows that within the general model the assumption that all lower bounds are zero does not limit the model.

Lemma 3.2.1 *Let $\mathcal{N} = (V, A, l, u, b, c)$ be a network.*

(a) Suppose that the arc $ij \in A$ has $l_{ij} > 0$. Let $\mathcal{N}'$ be obtained from $\mathcal{N}$ by making the following changes: $b(j) := b(j) + l_{ij}$, $b(i) := b(i) - l_{ij}$, $u_{ij} := u_{ij} - l_{ij}$, $l_{ij} := 0$. Then every feasible flow x in $\mathcal{N}$ corresponds to a feasible flow x' in $\mathcal{N}'$ and vice versa. Furthermore, the costs of these two flows are related by $c^T x = c^T x' + l_{ij}c_{ij}$.

(b) There exists a network $\mathcal{N}_{l\equiv 0}$ in which all lower bounds are zero such that every feasible flow x in $\mathcal{N}$ corresponds to a feasible flow x' in $\mathcal{N}_{l\equiv 0}$ and vice versa. Furthermore, the costs of these two flows are related by $c^T x = c^T x' + \sum_{ij \in A} l_{ij}c_{ij}$.

Proof: Part (a) is left to the reader as Exercise 3.3. Since we may eliminate lower bounds one arc at the time, (b) follows from (a) by induction on the number of arcs. □

It is also useful to observe that we can construct $\mathcal{N}'$ from $\mathcal{N}$ in time $O(n+m)$ and reconstruct the flow x from x' in time $O(m)$. Hence the time for eliminating lower bounds and reconstructing a flow in the original network is negligible since all algorithms on networks need $O(n+m)$ time just to input the network.

3.2.2 Flows with one Source and one Sink

Let s, t be distinct vertices of a network $\mathcal{N} = (V, A, l \equiv 0, u, c)$. An **$(s,t)$-flow** is a flow x satisfying the following for some $k \in \mathcal{R}_0$:

$$b_x(v) = \begin{cases} k & \text{if } v = s \\ -k & \text{if } v = t \\ 0 & \text{otherwise.} \end{cases}$$

The **value** of an (s,t)-flow x is denoted by $|x|$ and is defined by

$$|x| = b_x(s). \tag{3.8}$$

The next lemma combined with Lemma 3.2.1 shows that using only (s,t)-flows, one can model everything which can be modeled via flows in the general network model.

Lemma 3.2.2 *Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network. Let*[3] $M = \sum_{\{v:b(v)>0\}} b(v)$ *and let $\mathcal{N}_{st}$ be the network defined as follows: $\mathcal{N}_{st} = (V \cup \{s,t\}, A', l' \equiv 0, u', b', c')$, where*

(a) $A' = A \cup \{sr : b(r) > 0\} \cup \{rt : b(r) < 0\}$,
(b) $u'_{ij} = u_{ij}$ *for all* $ij \in A$, $u_{sr} = b(r)$ *for all* r *such that* $b(r) > 0$ *and* $u_{qt} = -b(q)$ *for all* q *such that* $b(q) < 0$,
(c) $c'_{ij} = c_{ij}$ *for all* $ij \in A$ *and* $c' = 0$ *for all arcs leaving* s *or entering* t,
(d) $b'(v) = 0$ *for all* $v \in V$, $b'(s) = M$, $b'(t) = -M$.

Then every feasible flow x in $\mathcal{N}$ corresponds to a feasible flow x' in $\mathcal{N}_{st}$ and vice versa. Furthermore, the costs of x and x' are related by $c^T x = c'^T x'$. See Figure 3.3.

Proof: Exercise 3.4. □

It follows from Lemma 3.2.2 that given any network $\mathcal{N}$ in which all lower bounds are zero, we can check the existence of a feasible flow in $\mathcal{N}$ by constructing the corresponding network $\mathcal{N}_{st}$ and check whether this network has

[3] Recall that we also have $M = -\sum_{\{v:b(v)<0\}} b(v)$ by (3.3).

an (s,t)-flow x such that $|x| = M$ where M is defined in Lemma 3.2.2. This latter task is precisely the problem of finding the maximum value of a feasible (s,t)-flow in $\mathcal{N}_{st}$, a problem which we study extensively in Sections 3.5-3.7. See also Theorem 3.8.3.

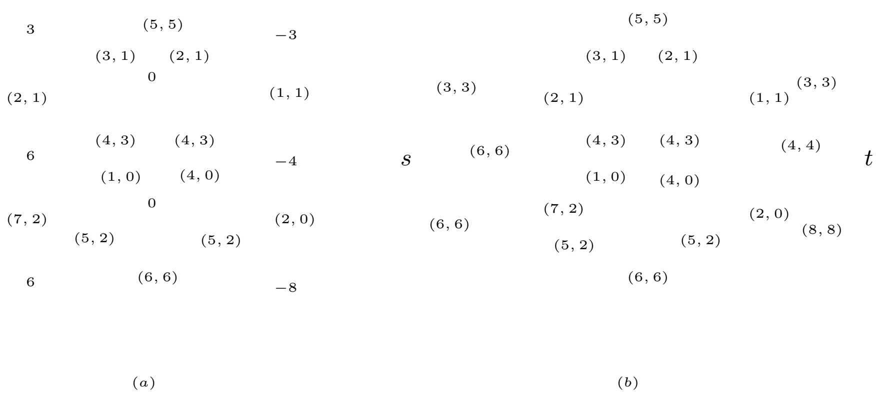


Figure 3.3 Part (a) shows a network $\mathcal{N}$ with a feasible flow with respect to the balance vector specified at each vertex. The numbers on each arc are (capacity, flow). Costs are omitted for clarity. Part (b) shows the network $\mathcal{N}_{st}$ as defined in Lemma 3.2.2 and a feasible flow x' in $\mathcal{N}_{st}$.

3.2.3 Circulations

A **circulation** is a flow x with $b_x(v) = 0$ for all $v \in V$. Combining our next result with Lemma 3.2.1 and Lemma 3.2.2 shows that one can also model everything that can be modeled in the general (flow) network model by the seemingly much more restricted circulations. Note that we cannot completely exclude lower bounds in this reduction (see Exercise 3.5).

Lemma 3.2.3 *Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network with distinct vertices s, t and let the balance vector of $\mathcal{N}$ satisfy $b(v) = 0$ for all $v \in V - \{s, t\}$, $b(s) = M$, $b(t) = -M$, for some $M \in \mathcal{R}_0$. Let $\mathcal{N}^* = (V, A \cup \{ts\}, l'', u'', c'')$ be the network obtained from $\mathcal{N}$ by adding a new arc ts with lower bound $l_{ts} = M$, capacity $u_{ts} = M$ and cost $c''_{ts} = 0$, keeping the lower bound, capacity and cost of each original arc and posing no restriction on the balance vector of $\mathcal{N}^*$. Then every feasible (s,t)-flow x in $\mathcal{N}$ corresponds to a feasible circulation x'' in $\mathcal{N}^*$ and vice versa. Furthermore, the costs of x and x' are related by $c^T x = c''^T x''$.*

Proof: Exercise 3.5. □

The concept of a circulation is a very useful tool for applications to questions concerning sub(di)graphs of (di)graphs as we show in Section 3.11.

3.2.4 Networks with Bounds and Costs on the Vertices

In some applications of flows one is not interested in imposing lower bounds and capacities on arcs, but rather on vertices. One such example is when one is looking for a cycle subdigraph that contains all vertices of a certain subset X and possibly other vertices (see Section 3.11). Another example is when one is looking for a path factor which covers all vertices of a digraph (see Section 5.3). We show below how to model networks with lower bounds, capacities and costs on vertices (and possibly also on arcs) by standard networks where all functions, other than the balance vectors, are on the arcs. First we introduce a useful transformation of any digraph to a bipartite digraph which we will use not only for the problem above but also several other places in the book.

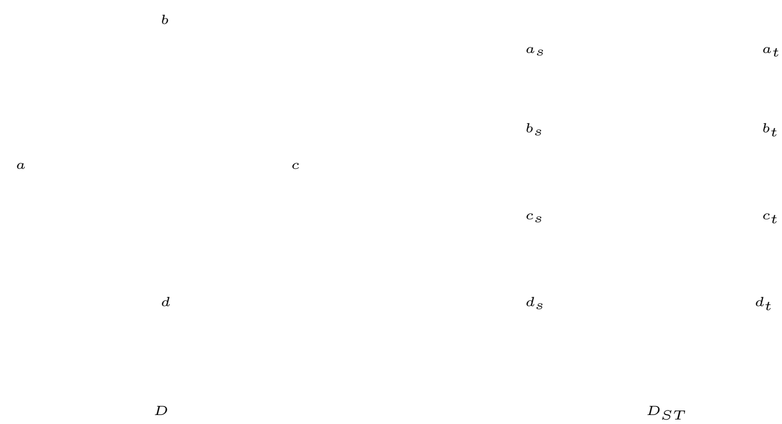


Figure 3.4 The vertex splitting procedure.

Given a digraph $D = (V, A)$, construct a new digraph D_{ST} as follows. For each vertex $v \in V$, D_{ST} contains two new vertices v_s, v_t and the arc $v_t v_s$. For each arc $xy \in A(D)$, $A(D_{ST})$ contains the arc $x_s y_t$. See Figure 3.4. We say that the digraph D_{ST} is obtained from D by the **vertex splitting procedure**.

Now suppose that $\mathcal{N} = (V, A, l, u, b, c, l^*, u^*, c^*)$ is a network with a prescribed balance vector b, lower bounds, capacities and costs l, u, c on the arcs (the case when there are no such specifications can easily be modeled by taking $l \equiv 0, u \equiv \infty, c \equiv 0$) and lower bounds, capacities and costs l^*, u^*, c^* on the vertices. To be precise we have to define the meaning of these new parameters. There is some freedom in such a definition, but for the applications we will need, it suffices to use the definition that $l^*(v)$ is the minimum and $u^*(v)$ the maximum amount of flow that may pass through v and the cost of sending one such unit through v is $c^*(v)$. By 'passing through' we mean the obvious thing when $b(v) = 0$ and if $b(v) > 0$ ($b(v) < 0$) we think of $l^*(v), u^*(v), c^*(v)$ as bounds and costs per unit on the total amount of flow out of (in to) v.

Let D_{ST} be the digraph obtained from $D = (V, A)$ by performing the vertex splitting procedure. Define a new network based on the digraph D_{ST} by adding lower bounds, capacities and costs as follows:

(a) For every arc $i_s j_t$ (corresponding to an arc ij of A) we let $h'(i_s j_t) = h(ij)$, where $h \in \{l, u, c\}$.
(b) For every arc $i_t i_s$ (corresponding to a vertex i of V) we let $h'(i_t i_s) = h^*(i)$, where $h^* \in \{l^*, u^*, c^*\}$.

Finally we define the function b' as follows:

If $b(i) = 0$ then $b'(i_s) = b'(i_t) = 0$;
If $b(i) > 0$ then $b'(i_t) = b(i)$ and $b'(i_s) = 0$;
If $b(i) < 0$ then $b'(i_t) = 0$ and $b'(i_s) = b(i)$.

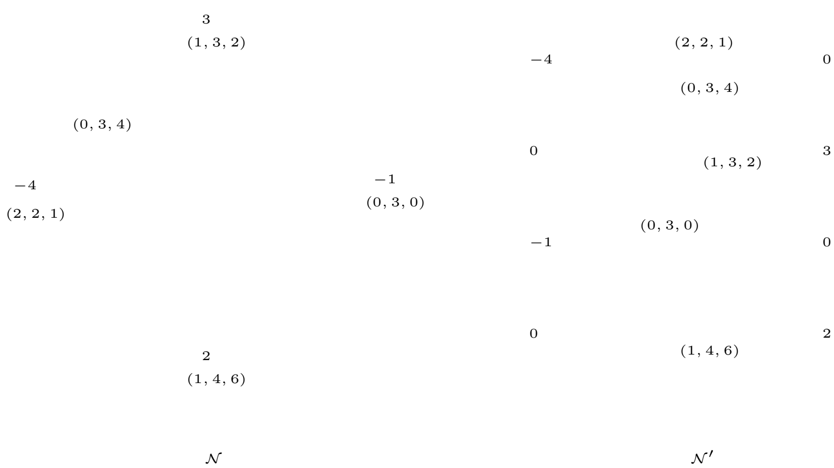


Figure 3.5 The construction of $\mathcal{N}'$ from $\mathcal{N}$. The specification is the balance vector and (l, u, c). For clarity only one arc of $\mathcal{N}$ has a description of bounds and cost.

See Figure 3.5 for an example of the construction. It is not difficult to show the following result.

Lemma 3.2.4 *Let $\mathcal{N}$ and $\mathcal{N}'$ be as described above. Then every feasible flow in $\mathcal{N}$ corresponds to a feasible flow in $\mathcal{N}' = (V(D_{ST}), A(D_{ST}), l', u', b', c')$ and vice versa. Furthermore, the costs of these flows are the same.*

Proof: Exercise 3.6. □

3.3 Flow Decompositions

In this section we consider a network $\mathcal{N} = (V, A, l \equiv 0, u)$ and denote by $D = (V, A)$ the underlying digraph of $\mathcal{N}$. By a path or cycle in $\mathcal{N}$ we mean a directed path or cycle in D. We will show that every flow in a network can be decomposed into a small number of very simple flows in the same network. Besides being a nice elementary mathematical result, this also has very important algorithmic consequences as will be clear from the succeeding sections.

A **path flow** $f(P)$ along a path P in $\mathcal{N}$ is a flow with the property that there is some number $k \in \mathcal{R}_0$ such that $f(P)_{ij} = k$ if ij is an arc of P and otherwise $f(P)_{ij} = 0$. Analogously, we can define a **cycle** flow $f(W)$ for any cycle W in D. The **arc sum** of two flows x, x', denoted $x + x'$, is simply the flow obtained by adding the two flows arc-wise.

Theorem 3.3.1 *Every flow x in $\mathcal{N}$ can be represented as the arc sum of some path and cycle flows $f(P_1), f(P_2), \ldots, f(P_\alpha), f(C_1), \ldots, f(C_\beta)$ with the following two properties:*

(a) Every directed path P_i, $1 \leq i \leq \alpha$ with positive flow connects a source vertex to a sink vertex.
(b) $\alpha + \beta \leq n + m$ and $\beta \leq m$.

Proof: Let x be a non-zero flow in $\mathcal{N}$. Suppose first that $b_x(i_0) > 0$ for some $i_0 \in V$. Since $b_x(i_0) > 0$ it follows from (3.5) that there is some arc $i_0 i_1$ leaving i_0 with $x_{i_0 i_1} > 0$. If $b(i_1) < 0$ then we have found a path from i_0 to the sink i_1. Otherwise $b(i_1) \geq 0$ and it follows from (3.5) and the fact that $x_{i_0 i_1} > 0$ that i_1 has some arc $i_1 i_2$ leaving it with $x_{i_1 i_2} > 0$. Continuing this way, we either find a path P from i_0 to a sink vertex i_k such that x is positive on all arcs on P, or eventually some vertex that was examined previously must be reached for the second time. In the later case we have detected a cycle $C = i_r i_{r+1} \ldots i_{p-1} i_p i_r$ such that x is positive on all arcs of C. Now we change the flow x as follows:

(i) If we detected a path P from i_0 to a sink i_k then let $\delta = \min\{x_{i_q i_{q+1}} : i_q i_{q+1} \in A(P)\}$ and define μ by $\mu = \min\{b_x(i_0), -b_x(i_k), \delta\}$. Let $f(P)$ be the path flow of value μ along P. Decrease x by μ units along P.
(ii) Otherwise we have detected a cycle C. Let $\mu = \min\{x_{i_q i_{q+1}} : i_q i_{q+1} \in A(C)\}$ and let $f(C)$ be a cycle flow of value μ along C. Decrease x by μ units along C.

If no arc carries positive flow after the changes made above we are done. Otherwise we repeat the process above. If every vertex v becomes balanced with respect to the current flow x (i.e. $b_x(v) = 0$) before x is identically zero, then just start from a vertex i_0 which has an arc $i_0 i_1$ with positive flow. From now on only cycle flows will be extracted in the subroutine described above.

Since each of these iterations either results in a vertex becoming balanced with respect to the current flow, or in an arc ij loosing all its flow, i.e., x_{ij} becomes zero, the total number of iterations, extracting either a path flow or a cycle flow from the current flow, is at most $n + m$. It follows from the description above that (a) and the first part of (b) holds. The second part of (b) follows from the fact that each time we extract a cycle flow at least one arc loses all its flow. □

The proof above immediately implies an algorithm for finding such a decomposition in time $O(m^2)$ if one uses DFS to find the next path or cycle flow to extract. However if we use an appropriate data structure and a little care, this complexity can be improved.

Lemma 3.3.2 *Given an arbitrary flow x in $\mathcal{N}$ one can find a decomposition of x into at most $n + m$ path and cycle flows, at most m of which are cycle flows, in time $O(nm)$.*

Proof: Exercise 3.7. □

The following useful fact is an easy consequence of Theorem 3.3.1.

Corollary 3.3.3 *Let $\mathcal{N}$ be a network. Every circulation in $\mathcal{N}$ can be decomposed into no more than m cycle flows.* □

3.4 Working with the Residual Network

Suppose $\mathcal{N}$ is a network and x, x' are feasible flows in $\mathcal{N}$. What can we say about the relation between x and x'? Clearly one can be obtained from the other by changing the flow along each arc appropriately, but we can reveal much more interesting relations as we shall see below. In fact it turns out that if x is feasible in $\mathcal{N}$ and x' is any other feasible flow in $\mathcal{N}$, then x' can be expressed in terms of x and some feasible flow in the residual network $\mathcal{N}(x)$. The other direction holds as well: if x is feasible in $\mathcal{N}$ and y is feasible in $\mathcal{N}(x)$ then we can 'add' y to x and obtain a new feasible flow in $\mathcal{N}$. These two properties imply that in order to study flows in a network $\mathcal{N}$ it suffices to find one feasible flow x and then work in the residual network $\mathcal{N}(x)$. We assume below that all lower bounds are zero. Recall that due to the results in Section 3.2 this restriction does not limit our modeling power.

The first lemma shows that if x is a feasible flow in $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ and $\tilde{x}$ is a feasible flow in $\mathcal{N}(x)$ then one can **'add'** $\tilde{x}$ to x and obtain a new feasible flow in $\mathcal{N}$. Here 'adding' is arc-wise and should be interpreted as defined below. Recall that we may assume we are dealing with netto flows.

Definition 3.4.1 *Let x be a feasible flow in $\mathcal{N} = (V, A, l \equiv 0, u, c)$ and let $\tilde{x}$ be a feasible flow in $\mathcal{N}(x)$. Define the flow $x^* = x \oplus \tilde{x}$ as follows: Start by letting $x^*_{ij} := x_{ij}$ for every $ij \in A$ and then for every arc ij in $\mathcal{N}(x)$ such that $\tilde{x}_{ij} > 0$ we modify x^* as follows (see Figure 3.6).*

(a) If $x_{ji} = 0$ *then* $x^*_{ij} := x_{ij} + \tilde{x}_{ij}$.
(b) If $x_{ij} = 0$ *and* $x_{ji} < \tilde{x}_{ij}$ *then* $x^*_{ij} := \tilde{x}_{ij} - x_{ji}$ *and* $x^*_{ji} := 0$.
(c) If $x_{ji} \geq \tilde{x}_{ij}$ *then* $x^*_{ji} := x_{ji} - \tilde{x}_{ij}$.

Note that by (3.7), if $0 < x_{ji} < \tilde{x}_{ij}$ then $ij \in A$. Using that x is a netto flow it is easy to check that the resulting flow x^* is also a netto flow.

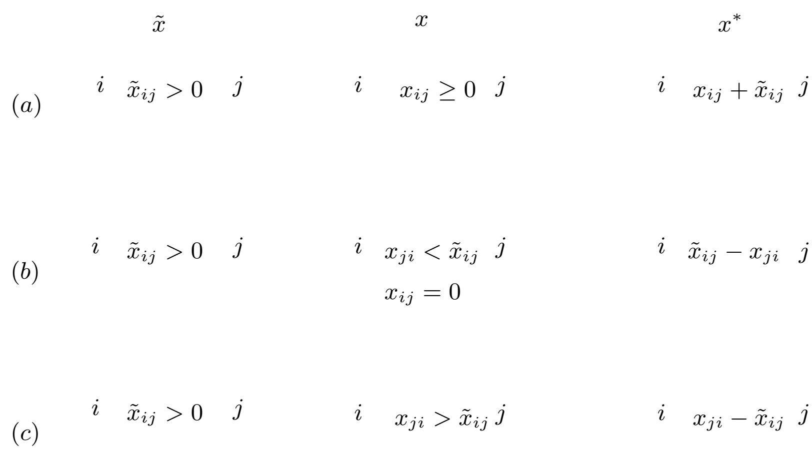


Figure 3.6 The three different cases in Definition 3.4.1. The three columns shows the flows $\tilde{x}$, x and x^*, respectively. An arc between i and j is shown unless the corresponding flow on that arc is zero.

Theorem 3.4.2 *Let* x *be a feasible flow in* $\mathcal{N} = (V, A, l \equiv 0, u, c)$ *with balance vector* b_x *and* $\tilde{x}$ *is a feasible flow in* $\mathcal{N}(x) = (V, A(x), r, c)$ *with balance vector* $b_{\tilde{x}}$. *Then* $x^* = x \oplus \tilde{x}$ *is a feasible flow in* $\mathcal{N}$ *with balance vector* $b_x + b_{\tilde{x}}$ *and the cost of* x^* *is given by* $c^T x^* = c^T x + c^T \tilde{x}$.

Proof: Let us first show that $0 \leq x^*_{ij} \leq u_{ij}$ for every $ij \in A$. We started the construction of x^* by letting $x^*_{ij} := x_{ij}$ for every arc. Hence it suffices to consider pairs (i, j) for which $\tilde{x}_{ij} > 0$. We consider the three possible cases (a)-(c) in Definition 3.4.1. In Case (a) we have $x^*_{ji} = 0$ and

$$\begin{aligned} 0 < x^*_{ij} = x_{ij} + \tilde{x}_{ij} &\leq x_{ij} + r_{ij} \\ &= x_{ij} + (u_{ij} - x_{ij} + x_{ji}) \\ &= u_{ij}, \end{aligned}$$

since we have $x_{ji} = 0$ in Case (a). In Case (b) we will have $x^*_{ji} = 0$ and

$$\begin{aligned} 0 \le x^*_{ij} = \tilde{x}_{ij} - x_{ji} &\le r_{ij} - x_{ji} \\ &= (u_{ij} - x_{ij} + x_{ji}) - x_{ji} \\ &= u_{ij}, \end{aligned}$$

since we have $x_{ij} = 0$ in Case (b). In Case (c) it is easy to see that we get $x^*_{ij} = 0$ and that $0 \le x^*_{ji} < u_{ji}$.

Consider the balance vector of the resulting flow. We wish to prove that x^* has balance vector $b_x + b_{\tilde{x}}$, that is, for every $i \in V$,

$$b_{x^*}(i) = \sum_{ij \in A} x^*_{ij} - \sum_{ji \in A} x^*_{ji} = b_x(i) + b_{\tilde{x}}(i). \tag{3.9}$$

This can be proved directly from the definitions of the balance expressions for x and $\tilde{x}$. However this approach is rather tedious and there is a simple inductive proof using Theorem 3.3.1. If $\tilde{x}$ is just a cycle flow in $\mathcal{N}(x)$, then it is easy to see (Exercise 3.12) that the balance vector of x^* equals that of x. Similarly, if $\tilde{x}$ is just a path flow of value δ along a (p,q)-path, for some distinct vertices $p, q \in V$, then $b_{x^*}(v) = b_x(v)$ for vertices v which are either internal vertices on P or not on P and $b_{x^*}(p) = b_x(p) + \delta$, $b_{x^*}(q) = b_x(q) - \delta$. In the general case, when $\tilde{x}$ is neither a path flow nor a cycle flow in $\mathcal{N}(x)$ we consider a decomposition of $\tilde{x}$ into path and cycle flows in $\mathcal{N}(x)$ according to Theorem 3.3.1. Using the observation above and Theorem 3.3.1 (implying that when adding all balance vectors of the paths and cycles in a decomposition, we obtain the balance vector of $\tilde{x}$) it is easy to prove by induction on the number of paths and cycles in the decomposition that (3.9) holds.

We leave it to the reader to prove using the same approach as above that the cost of x^* is given by $c^T x^* = c^T x + c^T \tilde{x}$ (see Exercise 3.12). □

The next theorem shows that the difference between any two feasible flows in a network can be expressed as a feasible flow in the residual network with respect to any of those flows.

Theorem 3.4.3 *Let $\mathcal{N} = (V, A, l \equiv 0, u, c)$ be a network and let x and x' be feasible netto flows in $\mathcal{N}$ with balance vectors b_x and $b_{x'}$. There exists a feasible flow $\bar{x}$ in $\mathcal{N}(x)$ with balance vector $b_{\bar{x}} = b_{x'} - b_x$ such that $x' = x \oplus \bar{x}$. Furthermore, the costs of these flows satisfy $c^T \bar{x} = c^T x' - c^T x$.*

Proof: Let x, x' be feasible netto flows in $\mathcal{N} = (V, A, l \equiv 0, u, c)$ and define a flow in $\mathcal{N}(x)$ as follows. For every arc $pq \in \mathcal{N}(x)$ we let $\bar{x}_{pq} := 0$ and then for every arc $ij \in A$ such that either $x_{ij} > 0$ or $x'_{ij} > 0$ holds, we modify $\bar{x}$ as follows:

(a) If $x_{ij} > x'_{ij}$ then $\bar{x}_{ji} := x_{ij} - x'_{ij} + x'_{ji}$.
(b) If $x'_{ij} > x_{ij}$ then $\bar{x}_{ij} := x'_{ij} - x_{ij} + x_{ji}$.

Using that x and x' are feasible netto flows in $\mathcal{N}$, one can verify that $\bar{x}$ is a feasible netto flow in $\mathcal{N}(x)$ (Exercise 3.13). It also follows easily from Definition 3.4.1 that $x' = x \oplus \bar{x}$. Now the last two claims regarding balance vector and cost follow from Theorem 3.4.2. □

The following immediate corollary of Theorem 3.4.3 and Corollary 3.3.3 will be useful when we study minimum cost flows in Section 3.10.

Corollary 3.4.4 *If x and x' are feasible flows in the network $\mathcal{N} = (V, A, l \equiv 0, u, c)$ such that $b_x = b_{x'}$, then there exist a collection of at most m cycles $W_1, W_2, \dots, W_k$ in $\mathcal{N}(x)$ and cycle flows $f(W_1), \dots, f(W_k)$ in $\mathcal{N}(x)$ such that the following holds:*

(a) $x' = x \oplus (f(W_1) + \dots + f(W_k)) = (\dots((x \oplus f(W_1)) \oplus f(W_2)) \oplus \dots) \oplus f(W_k)$;
(b) $c^T x' = c^T x + \sum_{i=1}^{k} c^T f(W_i)$. □

3.5 The Maximum Flow Problem

In this and the next section we study (s,t)-flows in networks with all lower bounds equal to zero. That is we consider networks of the type $\mathcal{N} = (V, A, l \equiv 0, u)$ where $s, t \in V$ are special vertices and we are only interested in flows x which satisfy $b_x(s) = -b_x(t)$ and $b_x(v) = 0$ for all other vertices. We call s the **source** and t the **sink** of $\mathcal{N}$. By Theorem 3.3.1, every (s,t)-flow x can be decomposed into a number of path flows along (s,t)-paths and some cycle flows whose values do not affect the value of the flow x. Based on this observation we also say that x is a flow **from s to t**.

Recall from (3.8) that the value $|x|$ of an (s,t)-flow is $|x| = b_x(s)$. We are interested in determining the maximum value k for which $\mathcal{N}$ has a feasible (s,t)-flow of value[4] k. Such a flow is called a **maximum flow** in $\mathcal{N}$. The problem of finding a maximum flow from s to t in a network with a specified source s and sink t is known as the **maximum flow problem** [246].

An **(s,t)-cut** is a set of arcs of the form $(S, \overline{S})$ where $S, \overline{S}$ form a partition of V such that $s \in S, t \in \overline{S}$. The **capacity** of an (s,t)-cut $(S, \overline{S})$ is the number $u(S, \overline{S})$, that is, the sum of the capacities of arcs with tail in S and head in $\overline{S}$ (recall (3.4)). Cuts of this kind are interesting in relation to the maximum flow problem as we shall see below.

Lemma 3.5.1 *For every (s,t)-cut $(S, \overline{S})$ and every (s,t)-flow x, we have*

$$|x| = x(S, \overline{S}) - x(\overline{S}, S). \tag{3.10}$$

Proof: Starting from the definition of $|x|$ and the fact that $b_x(v) = 0$ for all $v \in S - s$ we obtain

[4] Observe that there always exists a feasible flow in $\mathcal{N}$ since we have assumed $l \equiv 0$.

$$\begin{aligned}|x| &= b_x(s) + \sum_{i\in S-s} b_x(i)\\ &= \sum_{i\in S}(\sum_{ij\in A} x_{ij} - \sum_{ji\in A} x_{ji})\\ &= x(S,V) - x(V,S)\\ &= x(S,S) + x(S,\overline{S}) - x(\overline{S},S) - x(S,S)\\ &= x(S,\overline{S}) - x(\overline{S},S),\end{aligned}$$

where we also used (3.4). □

Since a feasible flow x satisfies $x \leq u$, every feasible (s,t)-flow must satisfy

$$x(S,\overline{S}) \leq u(S,\overline{S}) \text{ for every } (s,t)\text{-cut } (S,\overline{S}). \tag{3.11}$$

A **minimum (s,t)-cut** is an (s,t)-cut $(S,\overline{S})$ with

$$u(S,\overline{S}) = \min\{u(S',\overline{S'}) : (S',\overline{S'}) \text{ is an } (s,t)\text{-cut in } \mathcal{N}\}.$$

It follows from (3.11) and Lemma 3.5.1 that the capacity of any (s,t)-cut provides an upper bound for the value $|x|$ for any feasible flow x in the network. We also obtain the following useful consequence.

Lemma 3.5.2 *If a flow x has value $|x| = u(S,\overline{S})$ for some (s,t)-cut $(S,\overline{S})$, then $x(\overline{S},S) = 0$, x is a maximum (s,t)-flow and $(S,\overline{S})$ is a minimum (s,t)-cut.* □

Suppose x is an (s,t)-flow in $\mathcal{N}$ and P is an (s,t)-path in $\mathcal{N}(x)$ such that $r_{ij} \geq \epsilon > 0$ for each arc ij on P. Let x'' be the (s,t)-path flow of value ϵ in $\mathcal{N}(x)$ which is obtained by sending ϵ units of flow along the path P. By Theorem 3.4.2, we can obtain a new flow $x' = x \oplus x''$ of value $|x| + \epsilon$ in $\mathcal{N}$, implying that x is not a maximum flow in $\mathcal{N}$. We call a path P in $\mathcal{N}(x)$ as above an **augmenting path** with respect to x. The **capacity** $\delta(P)$ of an augmenting path P is given by

$$\delta(P) = \min\{r_{ij} : ij \text{ is an arc of } P\}. \tag{3.12}$$

We call an arc ij of P for which $x_{ij} < u_{ij}$ a **forward arc** of P and an arc ij of P for which $x_{ji} > 0$ a **backward arc** of P.

When we 'add' the path flow x'' to x according to Definition 3.4.1 we say that we **augment along P** by ϵ units. It follows from the definition of $\delta(P)$ and Theorem 3.4.2 that $\delta(P)$ is the maximum value by which we can augment x along P and still have a feasible flow in $\mathcal{N}$ after the augmentation.

Now we are ready to prove the following fundamental result, due to Ford and Fulkerson, relating minimum (s,t)-cuts and maximum (s,t)-flows.

Theorem 3.5.3 (Max-flow Min-cut theorem) *[246] Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t. For every feasible (s,t)-flow x in $\mathcal{N}$ the following are equivalent:*

(a) The flow x is a maximum (s,t)-flow.
(b) There is no (s,t)-path in $\mathcal{N}(x)$.
(c) There exists an (s,t)-cut $(S, \overline{S})$ such that $|x| = u(S, \overline{S})$.

Proof: We show that (a)⇒(b)⇒(c)⇒(a).

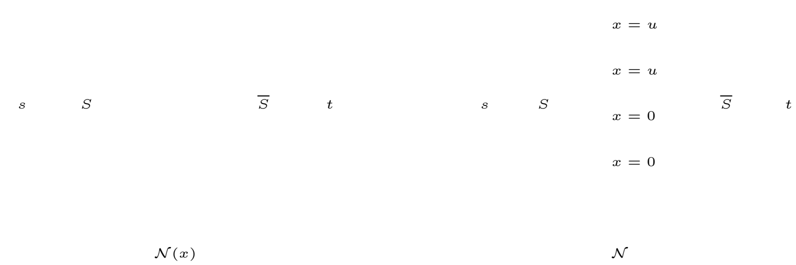


Figure 3.7 Illustration of part (b)⇒(c) in the proof of Theorem 3.5.3. The set S consists of those vertices that are reachable from s in $\mathcal{N}(x)$. The left part shows the situation in the residual network where we have $\overline{S} \Rightarrow S$ and the right part shows the corresponding situation in $\mathcal{N}$.

(a)⇒(b): Suppose x is a maximum flow in $\mathcal{N}$ and that $\mathcal{N}(x)$ contains an (s,t)-path P. Let $\delta(P) > 0$ be the capacity of P and let x' be the (s,t)-path flow in $\mathcal{N}(x)$ which sends $\delta(P)$ units of flow along P. By Theorem 3.4.2 $x \oplus x'$ is a feasible flow in $\mathcal{N}$ of value $|x| + \delta(P) > |x|$, contradicting the maximality of x. Hence (a)⇒(b).

(b)⇒(c): Suppose that $\mathcal{N}(x)$ contains no (s,t)-path. Let

$$S = \{y \in V : \mathcal{N}(x) \text{ contains an } (s,y)\text{-path}\}.$$

By the definition of S, there is no arc from S to $\overline{S}$ in $\mathcal{N}(x)$. Thus the definition of $\mathcal{N}(x)$ implies that for every arc $ij \in (S, \overline{S})$ we have $x_{ij} = u_{ij}$ and for every arc $ij \in (\overline{S}, S)$ we have $x_{ij} = 0$ (see Figure 3.7). This implies that we have $|x| = x(S, \overline{S}) - x(\overline{S}, S) = u(S, \overline{S}) - 0 = u(S, \overline{S})$. Hence we have proved that (b)⇒(c).

(c)⇒(a): This follows directly from Lemma 3.5.2. □

3.5.1 The Ford-Fulkerson Algorithm

The proof of Theorem 3.5.3 suggests the following simple method for finding a maximum (s,t)-flow in a network where all lower bounds are zero. Start with $x \equiv 0$. This is a feasible flow since $0 = l_{ij} \leq u_{ij}$ for all arcs $ij \in A$. Try to find an (s,t)-path P in $\mathcal{N}(x)$. If there is such a path P, then augment x by $\delta(P)$ units along P. Continue this way until there is no (s,t)-path in $\mathcal{N}(x)$ where x is the current flow. This method, due to Ford and Fulkerson [246], is called the **Ford-Fulkerson (FF) algorithm**.

Strictly speaking this is not really an algorithm if we do not specify how we wish to search for an augmenting (s,t)-path. It can be shown (see Exercise 3.17) that, when the capacities are allowed to take non-rational values and there is no restriction on the choice of augmenting paths (other than that one has to augment as much as possible along the current path), then the process above may continue indefinitely and without even converging to the right value of a maximum flow (see Exercise 3.17). For real-life applications this problem cannot occur since all numbers represented in computers are rational approximations of real numbers and in this case the algorithm will always terminate (Exercise 3.18).

Theorem 3.5.4 *If $\mathcal{N} = (V, A, l \equiv 0, u)$ has all capacities integers, then the Ford-Fulkerson algorithm finds a maximum (s,t)-flow in time $O(m|x^*|)$, where x^* is a maximum (s,t)-flow.*

Proof: The following generic process called the **labelling algorithm** will find an augmenting path in $\mathcal{N}(x)$ in time $O(n+m)$ if one exists[5]. Start with all vertices unlabelled except s and every vertex unscanned. In the general step we pick a labelled but unscanned vertex v and scan all its out-neighbours while labelling (by backwards pointers showing where a vertex got labelled from) those vertices among the out-neighbours of v that are un-labelled. If t becomes labelled this way, the process stops and an augmenting path, determined by the backwards pointers, is returned. If all vertices are scanned and t was not labelled the process stops and the set of labelled vertices S and its complement $\overline{S}$ correspond to a minimum (s,t)-cut (recall the proof of Theorem 3.5.3).

Each time we augment along an augmenting path, the value of the current flow increases by at least one, since the capacities in the residual network are all integers (this is clear in the first iteration and easy to establish by induction for the rest of the iterations of the algorithm). Hence there can be no more than $|x^*|$ iterations of the above search for a path and the complexity follows. $\square$

To see that the seemingly very pessimistic estimate in Theorem 3.5.4 for the time spent by the algorithm may in fact be realized, consider the network in Figure 3.8 and the sequence of augmenting paths specified there. The reader familiar with the literature on flows may see that our example is different from the classical example in books on flows. The reason for this is

[5] We could also use path finding algorithms such as BFS and DFS, but the original algorithm by Ford and Fulkerson uses only the generic labelling approach. See also Section 3.6.

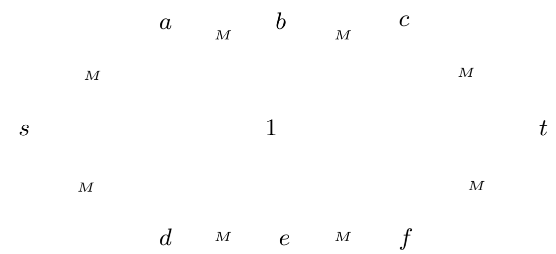


Figure 3.8 A possibly bad network for the Ford-Fulkerson algorithm. The number M denotes a large integer. If we choose augmenting paths of the form $sabeft$ with augmenting capacity 1 in odd numbered iterations and augmenting paths of the form *sdebct* with augmenting capacity 1 in even numbered iterations, then a maximum flow x of value $2M$ will be found only after $2M$ augmentations. Clearly, if instead we augment first along *sabct* and then along $sdeft$, each time by M units, we can find a maximum flow after just two augmentations.

that if we interpret the Ford-Fulkerson algorithm precisely as it is described in [246, page 18] (see also the proof of Theorem 3.5.4), then the algorithm will not behave badly on the usual example, whereas it still will do so on the example in Figure 3.8.

The value of the maximum flow in the example in Figure 3.8 is $2M$. This shows that the complexity of the Ford-Fulkerson algorithm is not bounded by a polynomial in the size of the input (recall from Chapter 1 that we assume that numbers are represented in binary notation). It is worth noting though that Theorem 3.5.4 implies that if all capacities are small integers then we get a very fast algorithm which, due to its simplicity, is easy to implement. The following is an easy but very important consequence of the proof of Theorem 3.5.3:

Theorem 3.5.5 (Integrality theorem for maximum (s,t)-flows) *[246]* *Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t. If all capacities are integers, then there exists an integer maximum (s,t)-flow in $\mathcal{N}$.*

Proof: This follows from our description of the Ford-Fulkerson algorithm. We start with $x \equiv 0$ and every time we augment the flow we do this by adding an integer valued path flow of value $\delta(P) \in \mathcal{Z}_+$. Hence the new (s,t)-flow is also an integer flow. It follows from the fact that all capacities are integers that in a finite number of steps we will reach a maximum flow (by Lemma 3.5.1 $|x|$ cannot exceed the capacity of any cut). Now the claim follows by induction on the number of augmentations needed before we have a maximum flow. □

An (s,t)-flow in a network $\mathcal{N}$ is **maximal** if every (s,t)-path in $\mathcal{N}$ uses at least one arc pq such that $x_{pq} = u_{pq}$ (such an arc is called **saturated**). That is, either x is maximum or after augmenting along an augmenting path P the resulting flow x' has $x'_{ij} < x_{ij}$ for some arc[6]. This is equivalent to saying that

[6] Recall that we always work with netto flows.

every augmenting path with respect to x contains at least one backward arc when P is considered as an oriented path in $\mathcal{N}$. It is important to distinguish between a maximal flow and a maximum flow. An (s,t)-flow x is maximal if it is either maximum, or in order to augment it to a flow with a higher value, we must reduce the flow in some arc. See also Figure 3.9.

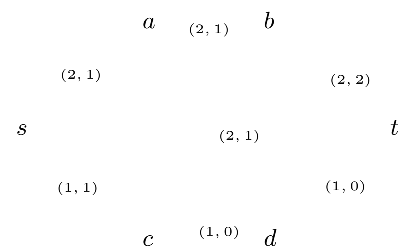


Figure 3.9 A network $\mathcal{N}$ with flow x which is maximal but not maximum as the path $P = sabcdt$ is an (s,t)-path in $\mathcal{N}(x)$. Note that the arc bc is a backward arc of P. The data on each arc are (capacity, flow).

3.5.2 Maximum Flows and Linear Programming

We digress for a short while to give some remarks on the relation between maximum flows and linear programming. First observe that the maximum flow problem (with lower bounds all equal to zero) is equivalent to the following linear programming problem:

$$\begin{aligned} &\text{maximize} \quad k \\ &\text{subject to} \\ &\qquad b_x(v) = \begin{cases} k & \text{if } v = s \\ -k & \text{if } v = t \\ 0 & \text{otherwise.} \end{cases} \\ &\qquad\qquad 0 \le x_{ij} \le u_{ij} \qquad\qquad \text{for every } ij \in A. \end{aligned}$$

The matrix T of the constraints of this linear program is given by $T = \begin{bmatrix} S \\ I \end{bmatrix}$, where S is the vertex-arc incidence matrix[7] of the underlying directed graph of the network (recall the definition of b_x) and I is the $m \times m$ identity matrix. The matrix S has the property that every column contains exactly

[7] The **vertex-arc incidence matrix** $S = [s_{ij}]$ of a digraph $D = (V, A)$ has rows labelled by the vertices of V and columns labelled by the arcs of A and the entry s_{v_i,a_j} equals 1 if the arc a_j has tail v_i, -1 if a_j has head v_i and 0, otherwise.

$+1$ and exactly one -1. This implies that S is **totally unimodular**, i.e., each square submatrix of S has determinant 0, 1, or -1 (see e.g., the book [166] by Cook, Cunningham, Pulleyblank and Schrijver). Hence it follows from Exercise 3.19 that the matrix T is also totally unimodular. Therefore the integrality theorem for maximum flows (Theorem 3.5.5) follows immediately from the Hoffmann-Kruskal characterization of total unimodularity (see [166, Theorem 6.25]).

Since the maximum flow problem is just a linear programming problem, it follows that one can find a maximum flow using any method for solving general linear programming problems. In particular, by the total unimodularity of T, the Simplex algorithm will always return an integer maximum flow provided that all capacities are integers. However, due to the special nature of the problem, more efficient algorithms can be found when we exploit the structure of flow problems. Finally, we remark that the Max-flow Min-cut theorem can be derived from the duality theorem for linear programming (see e.g. the book [600]).

3.6 Polynomial Algorithms for Finding a Maximum (s,t)-Flow

The Ford-Fulkerson algorithm can be modified in various ways to ensure that it becomes a polynomial algorithm. We describe two such modifications (see also Exercises 3.25 and 3.26). After doing so we describe a different approach in which we do not augment the flow by just one path at the time. For the first two subsections we need the following definition.

Definition 3.6.1 *A* **layered network** *is a network* $\mathcal{N} = (V, A, l \equiv 0, u)$ *with the following properties:*

(a) There is a partition $V = V_0 \cup V_1 \cup V_2 \cup \ldots \cup V_k \cup V_{k+1}$ *such that* $V_0 = \{s\}, V_{k+1} = \{t\}$ *and*
(b) every arc of A *goes from a* **layer** V_i *to the next layer* V_{i+1} *for some* $i = 0, 1, \ldots, k$.

See Figure 3.10 for an example of a layered network.

3.6.1 Flow Augmentations Along Shortest Augmenting Paths

Edmonds and Karp [216] observed that in order to modify the Ford-Fulkerson algorithm so as to get a polynomial algorithm, it suffices to choose the augmenting paths as shortest paths with respect to the number of arcs on the path.

Let x be a feasible (s,t)-flow in a network $\mathcal{N}$. Denote by $\delta_x(s,t)$ the length of a shortest (s,t)-path in $\mathcal{N}(x)$. If no such path exists we let $\delta_x(s,t) = \infty$.

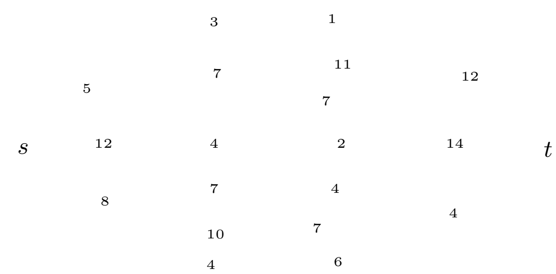


Figure 3.10 A layered network with source s and sink t. The numbers on the arcs indicate the capacities.

Suppose that there is an augmenting path in $\mathcal{N}(x)$ and let P be a shortest such path. Let r be the number of arcs in P. Define the network $\mathcal{LN}(x)$ as the network one obtains from $\mathcal{N}(x)$ by taking the vertices from the distance classes $V_0, V_1, \ldots, V_r$, i.e. $V_i = \{v : \mathrm{dist}_{\mathcal{N}(x)}(s,v) = i\}$, and all arcs belonging to $(V_i, V_{i+1})_{\mathcal{N}(x)}$ for $i = 0, 1, \ldots, r-1$ along with their residual capacities r_{ij}. Observe that, by the definition of distance classes, $\mathcal{LN}(x)$ contains all the shortest augmenting paths with respect to x in $\mathcal{N}(x)$.

The crucial fact that makes augmenting along shortest paths a good approach is the following lemma.

Lemma 3.6.2 *[216] Let x be a feasible (s,t)-flow in $\mathcal{N}$ and let x' be obtained from x by augmenting along a shortest path in $\mathcal{N}(x)$. Then*

$$\delta_{x'}(s,t) \geq \delta_x(s,t). \tag{3.13}$$

Proof: Suppose this is not the case for some x, x' where x' is obtained from x by augmenting along a shortest path P in $\mathcal{N}(x)$. By the remark above $\mathcal{LN}(x)$ contains all the shortest augmenting paths (with respect to x) in $\mathcal{N}(x)$. Let $r = \delta_x(s,t)$. By our assumption $\mathcal{N}(x')$ contains an (s,t)-path P' whose length is less than r. Thus P' must use an arc ij such that $ij \notin A(\mathcal{N}(x))$. However, every arc that is in $\mathcal{N}(x')$ but not in $\mathcal{LN}(x)$ is of the form ji where ij is an arc of P, or is inside a layer of $\mathcal{LN}(x)$. It follows that P' has at least $r+1$ arcs, contradicting the assumption. $\square$

Note that even if $\mathcal{N}(x')$ contains no (s,t)-path of length $\delta_x(s,t)$, it may still contain a path of length $\delta_x(s,t)+1$, since we may use an arc which was inside a layer of $\mathcal{LN}(x)$.

Theorem 3.6.3 (Edmonds, Karp) *[216] If we always augment along shortest augmenting paths, then the Ford-Fulkerson algorithm has complexity $O(nm^2)$.*

Proof: By Lemma 3.6.2, the length of the current augmenting path increases monotonically throughout the execution of the algorithm. It follows from the proof of Lemma 3.6.2 that, if the length of the next augmenting path does not go up, then that path is also a path in $\mathcal{LN}(x)$. Note also that at least one arc from some layer V_i to the next disappears after each augmentation (recall that in each augmentation we augment by $\delta(P)$ units along the current augmenting path P). Hence the number of iterations in which the length of the current augmenting path stays constant is at most m. Since the length can increase at most $n-2$ times (the length of an (s,t)-path is at least 1 and at most $n-1$) and we can find the next augmenting path in time $O(n+m)$ using BFS we obtain the desired complexity. □

Zadeh [753] constructed networks with n vertices and m arcs for which the Edmonds-Karp algorithm requires $\Omega(nm)$ augmentations to find a maximum flow. Hence the estimate on the worst case complexity is tight.

3.6.2 Blocking Flows in Layered Networks and Dinic's Algorithm

Let $\mathcal{L} = (V = V_0 \cup V_1 \cup \ldots \cup V_k, A, l \equiv 0, u)$ be a layered network with $V_0 = \{s\}$ and $V_k = \{t\}$. An (s,t)-flow x in $\mathcal{L}$ is **blocking** if there no (s,t)-path of length k in the residual network $\mathcal{L}(x)$. Note that a blocking flow is also maximal flow (recall the difference between a maximal and a maximum flow as explained in the end of Section 3.5). That is, every augmenting path with respect to x (if there is any) must use at least one arc pq such that $p \in V_j, q \in V_i$ for some $j \geq i$.

We saw above that if we always augment along shortest augmenting paths, then the length of a shortest augmenting path is monotonically increasing. Hence if we have a method to find a blocking flow in a layered network in time $O(p(n,m))$, then we can use that method to obtain an $O(np(n,m))$ algorithm for finding a maximum (s,t)-flow in any given network.

The method of Edmonds and Karp above achieves a blocking flow in time $O(m^2)$. It was observed by Dinic [195] (who also independently and earlier discovered the method of using shortest augmenting paths) that a blocking flow in a layered network can be obtained in time $O(nm)$, thus resulting in an $O(n^2m)$ algorithm for maximum flow.

The idea is to search for a shortest augmenting path in a depth first search manner. We modify slightly the standard DFS algorithm (see Section 4.1) as shown below. The vector π is used to remember the arcs of the augmenting path detected if one is found.

Dinic's algorithm (one phase)

Input: A layered network $\mathcal{L} = (V = V_0 \cup V_1 \cup \ldots \cup V_k, A, l \equiv 0, u)$.
Output: A blocking flow x in $\mathcal{L}$.

1. **Initialization:** $x_{ij} := 0$ for every arc ij in A, let $v := s$ be the current vertex and let $A' := A$.

2. **Searching step:** If there is no arc with tail v in A' (from v to the next layer among the remaining arcs), then if $v = s$ go to Step 5; otherwise go to Step 4;
   If there is an arc $vw \in A'$, then let $v := w$, let $\pi(w) := v$. If $v \neq t$ repeat Step 2.
3. **Augmentation step:** Using the π labels find the augmenting path P detected and augment x along P by $\delta(P)$ units. Delete all arcs ij of A' for which $x_{ij} = u_{ij}$. Erase all labels on vertices ($\pi(i) := nil$ for all $i \in V$). Let $v := s$ and go to Step 2.
4. **Arc deletion step:** (The search above has revealed that there is no (v,t)-path in the current digraph $D' = (V, A')$. Furthermore, $v \neq s$). Delete all arcs with head or tail v from A', let $v := \pi(v)$ and go to Step 2.
5. **Termination:** Return the blocking flow x.

Theorem 3.6.4 *Dinic's algorithm (one phase of) correctly determines a blocking flow in a given layered network $\mathcal{L}$ in time $O(nm)$.*

Proof: Let $\mathcal{L} = (V = V_0 \cup V_1 \cup \ldots \cup V_k, A, l \equiv 0, u)$. Each time the current flow is augmented in the algorithm it is changed along an augmenting path of length k. We only delete an arc from A' when it is no longer present in the residual network $\mathcal{L}(x)$ where x is the current flow. Hence no deleted arc could be used in an augmenting path of length k with respect to the current flow. Furthermore, when the algorithm terminates there is no (s,t)-path in the current digraph $D' = (V, A')$. Here A' consists of those arcs from one layer to the next which are still not filled to capacity by the current x. It follows that the algorithm terminates with a blocking flow.

The complexity follows from the fact that we perform at most $O(n)$ steps between each deletion of an arc which is either saturated (via the actual augmenting path P) or enters a vertex for which we deleted all arcs having that vertex as the head or tail (see Step 4). $\square$

3.6.3 The Preflow-Push Algorithm

The flow algorithms we have seen in the previous sections have the common feature that they all increase the flow along one augmenting path at a time. Very often, when searching for an augmenting path, one finds a path P containing an arc rq whose capacity is relatively small compared to the capacity of the prefix $P[s,r]$ of that path (see e.g. Figure 3.11). This means that along $P[s,r]$ we were able to augment by a large amount of flow, but due to the smaller capacity of the arc rq we only augment by that smaller amount and start all over again. In Dinic's algorithm this could be taken into account by not starting all over again, but instead backtracking until a new forward arc can be found in the layered network. However we are still limited to finding one path at a time. Now we present a different approach, due to Goldberg

Figure 3.11 A bad example for a standard flow algorithm such as the Edmonds-Karp algorithm. The capacities of arcs are either 1, if no number is shown or M, where M is a large number. Algorithms such as the Edmonds-Karp algorithm will augment M times along the path from s to r each time by just one unit.

and Tarjan [324, 325], which allows one to work with more than one augmenting path at a time. The algorithm of Goldberg and Tarjan, called the **preflow-push algorithm**, tries to push as much flow towards t as possible, by first sending the absolute maximum possible, namely $\sum_{sr\in A} u_{sr}$, out of s and then trying to push this forward to t. At some point no more flow can be sent to t and the algorithm returns the excess flow back to s again. This very vague description will be made precise below (the reader should compare this with the so-called MKM-algorithm described in Exercise 3.25).

Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t. A feasible flow x in $\mathcal{N}$ is called a **preflow** if $b_x(v) \le 0$ for all $v \in V - s$. Note that every (s,t)-flow x is also a preflow since we have $b_x(v) = 0$ if $v \in V - \{s,t\}$ and $b_x(t) = -b_x(s) \le 0$. Hence preflows generalize (s,t)-flows, an observation that we shall use below. Let x be a preflow in a network $\mathcal{N}$. A **height function with respect to *x*** is a function $h : V \to \mathcal{Z}_0$ which satisfies

$$\begin{aligned} &h(s) = n, \qquad h(t) = 0; && \qquad (3.14)\\ &h(p) \le h(q) + 1 \qquad \text{for every arc } pq \text{ of } \mathcal{N}(x). \end{aligned}$$

The following useful lemma is an immediate consequence of Theorem 3.3.1(a).

Lemma 3.6.5 *Let x be a preflow in a network $\mathcal{N} = (V, l \equiv 0, u)$ with source s and sink t and let v be a vertex such that $b_x(v) < 0$. Then $\mathcal{N}(x)$ contains a (v,s)-path.*

Proof: By the definition of a preflow, s is the only vertex r for which we have $b_x(r) > 0$. Hence, by Theorem 3.3.1(a), every decomposition of x into path and cycle flows contains an (s,v)-path P. Now it follows that $\mathcal{N}(x)$ contains a (v,s)-path, since every arc of P has positive flow in $\mathcal{N}$ and hence give rise to an oppositely oriented arc in $\mathcal{N}(x)$. $\square$

Now we are ready to describe the (generic) preflow-push algorithm. During the execution of the algorithm, a vertex $v \in V$ is called **active** if $b_x(v) < 0$. An arc pq of $\mathcal{N}(x)$ is **admissible** if $h(p) = h(q) + 1$. The algorithm uses two basic operations **push** and **lift**.

push(pq): Let p be a vertex with $b_x(p) < 0$ and let pq be an admissible arc in $\mathcal{N}(x)$. The operation **push**(pq) changes x_{pq} to $x_{pq} + \rho$, where $\rho = \min\{-b_x(p), r_{pq}\}$.

lift(p): Let p be a vertex with $b_x(p) < 0$ and $h(p) \leq h(q)$ for every arc pq in $\mathcal{N}(x)$. The operation **lift**(p) changes the height of p as follows:

$$h(p) := \min\{h(z) + 1 : pz \text{ is an arc of } \mathcal{N}(x)\}.$$

By the remark after the proof of Lemma 3.6.5, the number $h(p)$ is well-defined. See Figure 3.12 for an illustration of a lift.

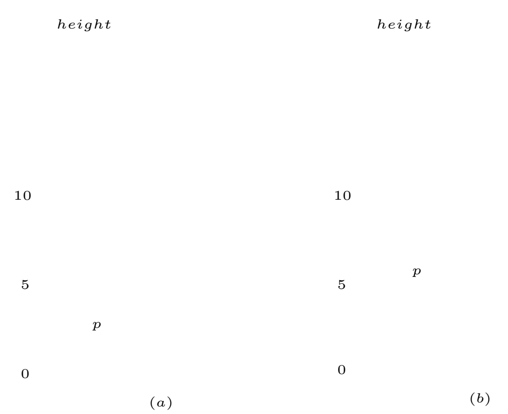


Figure 3.12 Lifting the vertex p form height 4 to height 7.

Lemma 3.6.6 *Let x be a preflow in $\mathcal{N}$ and let h be defined as in (3.14). If $p \in V$ satisfies $b_x(p) < 0$, then at least one of the operations* **push**(pq), **lift**(p) *can be applied.*

Proof: Suppose $b_x(p) < 0$, but we cannot perform a push from p. Then there is no admissible arc with tail p and hence we have $h(p) \leq h(q)$ for every arc pq in $\mathcal{N}(x)$. It follows from Lemma 3.6.5 that there is at least one arc out of p in $\mathcal{N}(x)$ and hence we can perform the operation $\textbf{lift}(p)$. $\square$

The generic preflow-push algorithm
Input: A network $\mathcal{N} = (V, l \equiv 0, u)$ with source s and sink t.
Output: A maximum (s,t)-flow in $\mathcal{N}$.
Preprocessing step:

(a) For each $p \in V$ let $h(p) := \text{dist}_{\mathcal{N}}(p,t)$;
(b) Let $h(s) := n$;
(c) Let $x_{sp} := u_{sp}$ for every arc out of s in $\mathcal{N}$;
(d) Let $x_{ij} := 0$ for all other arcs in $\mathcal{N}$.

Main loop:
While there is an active vertex $p \in V - t$ do the following:
if $\mathcal{N}(x)$ contains an admissible arc pq then $\textbf{push}(pq)$ else $\textbf{lift}(p)$.

Theorem 3.6.7 *The generic preflow-push algorithm correctly determines a maximum (s,t)-flow in $\mathcal{N}$ in time $O(n^2m)$.*

Proof: We first show that the function h remains a height function throughout the execution of the algorithm. Initially this is the case since we use exact distance labels and there are no arcs out of s in $\mathcal{N}(x)$ (Exercise 3.20). Observe that for every vertex p, $h(p)$ is only changed when we perform the operation $\textbf{lift}(p)$ and then it is changed so as to preserve the condition (3.14). Furthermore, the operation $\textbf{push}(pq)$ may introduce a new arc qp in $\mathcal{N}(x)$, but this arc will satisfy $h(q) = h(p) - 1$ and hence does not violate (3.14). It follows that h remains a height function throughout the execution of the algorithm.

It is easy to see that x remains a preflow throughout the execution of the algorithm, since only a push operation affects the current x and by definition a push operation preserves the preflow condition.

Now we prove that, if the algorithm terminates, then it does so with a maximum flow x. Suppose that the algorithm has terminated. This means that no vertex $v \in V$ has $b_x(v) < 0$. Thus it follows from the definition of a preflow that x is an (s,t)-flow. To prove that x is indeed a maximum flow, it suffices to show that there is no (s,t)-path in $\mathcal{N}(x)$. This follows immediately from the fact that h remains a height function throughout the execution of the algorithm. By (3.14), every arc pq in $\mathcal{N}(x)$ has $h(p) \leq h(q) + 1$ and we always have $h(s) = n, h(t) = 0$. Since no (s,t)-path has more than $n-1$ arcs, there is no (s,t)-path in $\mathcal{N}(x)$ and hence, by Theorem 3.5.3, x is a maximum (s,t)-flow.

To prove that the algorithm terminates and to determine its complexity, it is useful to distinguish between two kinds of pushes. An execution of the operation **push**(pq) is a **saturating push** if the arc pq is filled to capacity after the push and hence pq is not an arc of $\mathcal{N}(x)$ immediately after that push. A push which is not saturating is an **unsaturating push**.

We now establish a number of claims from which the complexity of the algorithm follows.

(A) **The total number of lifts is $O(n^2)$:** By Lemma 3.6.5, every vertex p with $b_x(p) < 0$ has a path to s in $\mathcal{N}(x)$. Hence, we have $h(p) \leq 2n-1$, by (3.14). Since the height of a vertex p increases by at least one every time the operation **lift**(p) is performed, no vertex can be lifted more than $2n-2$ times the claim follows.

(B) **The total number of saturating pushes is $O(nm)$:** Let us consider a fixed arc pq and find an upper bound for the number of saturating pushes along this arc in the algorithm. When we perform a saturating push along pq, we have $h(p) = h(q) + 1$ and the arc pq disappears from the residual network. It can only appear again in the current residual network after flow has been pushed from q to p in some later execution of the operation **push**(qp). At that time we have $h(q) = h(p) + 1$. This and the fact that h remains a height function and never decreases at any vertex, implies that before we can perform a new saturating push along pq, $h(p)$ has increased by at least two. We argued above that we always have $h(p) \leq 2n-1$ and now we conclude that there are at most $O(n)$ saturating pushes along any given arc. Thus the total number of saturating pushes is $O(nm)$.

(C) **The total number of unsaturating pushes is $O(n^2m)$:** Let $\Phi = \sum_{b_x(v)<0} h(v)$. Then $\Phi \geq 0$ during the whole execution of the algorithm and since $h(v) < 2n$ at any time during the execution we have $\Phi \leq 2n^2$ after the preprocessing step. Let us examine what happens to the value of Φ after performing the different kinds of operations. A lift will increase Φ by at most $2n-1$. Hence, by (A), the total contribution to Φ from lifts is $O(n^3)$. A saturating push from p to q can increase Φ by at most $h(q) \leq 2n-1$ (it may also decrease Φ if p becomes balanced, but we are not concerned about that here). Hence, by (B), the total contribution to Φ from saturating pushes is $O(n^2m)$. An unsaturating push from p to q will decrease Φ by at least one, since p becomes balanced and $h(p) = h(q)+1$ (if q was balanced before, then Φ decreases by one and otherwise it decreases by $h(p)$).
It follows from the considerations above that the total increase in Φ during the execution of the algorithm is $O(n^2m)$. Now it follows from the fact that Φ is never negative that the total number of unsaturating pushes is $O(n^2m)$. □

It is somewhat surprising that the simple approach above results in an algorithm of such a low complexity. The complexity bound is valid no matter which vertex we choose to push from or lift. This indicates the power of the approach. However, the algorithm does have its drawbacks. If no control is supplied to direct the algorithm (as to which vertices to push from or lift), then a large amount of time may be spent without any effect on the final maximum flow. The reader is asked in Exercise 3.21 to give an example showing that a large amount of useless work may be performed if no extra guidance is given to the choice of pushes. There are several approaches which can improve the performance of the preflow-push algorithm we mention just two of these. For details see e.g. [7].

(a) If we examine the active vertices in a first-in first-out (FIFO) order, then we obtain an $O(n^3)$ algorithm [325].
(b) If we always push from a vertex p which has the largest height $h(p)$ among all active vertices, then we obtain an $O(n^2\sqrt{m})$ algorithm [149, 325].

Cheriyan and Maheshwari [149] have shown by examples that the worst case bounds for the FIFO and maximum height variants are tight. For another way to improve the performance of the generic algorithm in practice, see Exercise 3.22.

3.7 Unit Capacity Networks and Simple Networks

In this section we consider two special cases of networks, both of which occur in applications and for which, due to their special structure, one can obtain faster algorithms for finding a maximum flow. All networks considered in this section are assumed to have a source s and a sink t.

3.7.1 Unit Capacity Networks

A **unit capacity network** is a network $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$, i.e. all arcs have capacity equal to one. Unit capacity networks are important in several applications of flows to problems such as finding a maximum matching in a bipartite graph (Subsection 3.11.1), finding an optimal path cover of an acyclic digraph (Section 5.3) and finding cycle subdigraphs covering specified vertices (Subsection 3.11.5).

Lemma 3.7.1 *If $\mathcal{N}$ is a unit capacity network without cycles of length 2 and x is a feasible (s,t)-flow, then $\mathcal{N}(x)$ is also a unit capacity network.*

Proof: Exercise 3.39. □

Let $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$ be a unit capacity network with source s and sink t. Since the value of a minimum (s,t)-cut in $\mathcal{N}$ is at most $n-1$ (consider the cut $(s, V-s)$), we see from Theorem 3.5.4 that the Ford-Fulkerson algorithm will find a maximum (s,t)-flow in time $O(nm)$. The purpose of this section is to show that one can obtain an even faster algorithm. Our exposition is based on an idea due to Even and Tarjan [232].

Lemma 3.7.2 *Let $\mathcal{L} = (V = V_0 \cup V_1 \cup \ldots \cup V_k, A, l \equiv 0, u \equiv 1)$ be a layered unit capacity network with $V_0 = \{s\}$ and $V_k = \{t\}$. One can find a blocking (s,t)-flow in $\mathcal{L}$ in time $O(m)$.*

Proof: It suffices to see that the capacity of each augmenting path is 1 and no two augmenting paths of the same length can use the same arc. Hence it follows that Dinic's algorithm will find a blocking flow in time $O(m)$. □

Lemma 3.7.3 *Let $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$ be a unit capacity network and let x^* be a maximum (s,t)-flow in $\mathcal{N}$. Then*

$$\mathrm{dist}_{\mathcal{N}}(s,t) \leq 2n/\sqrt{|x^*|} \tag{3.15}$$

Proof: Let $\omega = \mathrm{dist}_{\mathcal{N}}(s,t)$ and let $V_0 = \{s\}, V_1, V_2, \ldots, V_\omega$ be the first ω distance classes from s. Since $\mathcal{N}$ contains no multiple arcs, the number of arcs from V_i to V_{i+1} is at most $|V_i||V_{i+1}|$ for $i = 0, 1, \ldots, \omega - 1$. Since the arcs in (V_i, V_{i+1}) correspond to the arcs across an (s,t)-cut in $\mathcal{N}$, we have $|x^*| \leq |V_i||V_{i+1}|$ for $i = 0, 1, \ldots, \omega - 1$. Thus $\max\{|V_i|, |V_{i+1}|\} \geq \sqrt{|x^*|}$ for $i = 0, 1, \ldots, \omega - 1$. Now we easily see that

$$n = |V| \geq \sum_{i=0}^{\omega} |V_i| \geq \sqrt{|x^*|}\lfloor\frac{\omega+1}{2}\rfloor \tag{3.16}$$

implying that $\omega \leq 2n/\sqrt{|x^*|}$. □

Theorem 3.7.4 *[232] For unit capacity networks the complexity of Dinic's algorithm is $O(n^{\frac{2}{3}}m)$.*

Proof: Let $\mathcal{N}$ be a unit capacity network with source s and sink t. We assume for simplicity that $\mathcal{N}$ has no 2-cycles. The case when $\mathcal{N}$ does have a 2-cycle can be handled similarly (Exercise 3.41). Let q be the number of phases performed by Dinic's algorithm before a maximum (s,t)-flow is found in $\mathcal{N}$. Let $0 \equiv x^{(0)}, x^{(1)}, \ldots, x^{(q)}$ denote the (s,t)-flows in $\mathcal{N}$ which have been calculated after the successive phases of the algorithm. Thus $x^{(0)}$ is the starting flow which is the zero flow and $x^{(i)}$ denotes the flow after phase i of the algorithm. Let $\tau = \lceil n^{\frac{2}{3}} \rceil$ and let $K = |x^{(q)}|$ denote the value of a maximum (s,t)-flow in $\mathcal{N}$.

By Lemmas 3.7.1 and 3.7.2 it suffices to prove that the total number of phases, q, is $O(n^{\frac{2}{3}})$. This is clear in the case when $K \leq \tau$, since we augment the flow by at least one unit after each phase. So suppose that $K > \tau$. Choose j such that $|x^{(j)}| < K - \tau$ and $|x^{(j+1)}| \geq K - \tau$. By Theorem 3.4.2 and Theorem 3.4.3 the value of a maximum flow in $\mathcal{N}(x^{(j)})$ is $K - |x^{(j)}| > \tau$.

Applying Lemmas 3.7.1 and 3.7.3 to $\mathcal{N}(x^{(j)})$, we see that $\mathrm{dist}_{\mathcal{N}(x^{(j)})}(s,t) \leq 2n^{\frac{2}{3}}$. Using Lemma 3.6.2 and the fact that each phase of Dinic's algorithm results in a blocking flow, we see that $j \leq 2n^{\frac{2}{3}}$. Thus, since at most τ phases remain after phase j we conclude that the total number of phases q is $O(n^{\frac{2}{3}})$. □

3.7.2 Simple Networks

A **simple network** is a network $\mathcal{N} = (V, A, l \equiv 0, u)$ with special vertices s, t in which every vertex in $V - \{s, t\}$ has precisely one arc entering or precisely one arc leaving. For an example see Figure 3.13.

s t

Figure 3.13 A simple network. Capacities are not shown.

Below we assume that the simple network in question does not have any 2-cycles. It is easy to see that this is not a serious restriction (Exercise 3.42).

Lemma 3.7.5 *Let $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$ be a simple unit capacity network on n vertices and let x^* be a maximum (s,t)-flow in $\mathcal{N}$. Then*

$$\mathrm{dist}_{\mathcal{N}}(s,t) \le n/|x^*|. \tag{3.17}$$

Proof: Let $\omega = \mathrm{dist}_{\mathcal{N}}(s,t)$ and $V_0 = \{s\}, V_1, V_2, \ldots, V_\omega$ be the first ω distance classes from s. Every unit of flow from s to t passes through the layer V_i for $i = 1, 2, \ldots, \omega - 1$. Furthermore, since $\mathcal{N}$ is a simple unit capacity network, at most one unit of flow can pass through each $v \in V$. Thus $|V_i| \ge |x^*|$, for $i = 1, 2, \ldots, \omega - 1$ and hence

$$|V| > \sum_{i=1}^{\omega-1} |V_i| \ge (\omega - 1)|x^*|,$$

implying that $\omega \le |V|/|x^*|$. □

Lemma 3.7.6 *If $\mathcal{N}$ is a simple unit capacity network, then $\mathcal{N}(x)$ is also a simple unit capacity network.*

Proof: Exercise 3.40. □

Using Lemma 3.7.5 and Lemma 3.7.6 one can prove the following result due to Even and Tarjan. We leave the details as Exercise 3.43.

Theorem 3.7.7 *[232] For simple unit capacity networks Dinic's algorithm has complexity $O(\sqrt{n}m)$.* □

We point out that Dinic's algorithm will also find a maximum (s,t)-flow in time $O(\sqrt{n}m)$ in a simple network even if not all capacities are one, provided that the network has the property that at most one unit of flow can pass through any vertex $v \in V - \{s,t\}$. In particular a vertex may be the tail of an arc with capacity ∞ provided that it is the head of at most one arc and this arc (if it exists) has capacity one. We use this extension of Theorem 3.7.7 in Section 3.11.

3.8 Circulations and Feasible Flows

We now return to the general flow model when lower bounds are present on the arcs. We wish to determine whether a feasible flow exists with respect to the given lower bounds and capacities on the arcs and a prescribed balance vector. As we showed in Section 3.2, in order to study the general case, it suffices to study circulations since we may use Lemmas 3.2.1-3.2.3 to transform the general case to the case of circulations. Note that in this section we always assume that all the data of the network are integers (that is l and u are integers).

We need the following very simple observation. The proof is analogous to that of Lemma 3.5.1.

Lemma 3.8.1 *If x is a circulation in $\mathcal{N}$ then for every partition $S, \overline{S}$ of V we have $x(S,\overline{S}) = x(\overline{S},S)$.* □

The example in Figure 3.14 gives us a starting point for detecting what can prevent the existence of a feasible circulation.

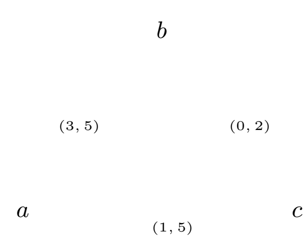


Figure 3.14 A network with no feasible circulation. The specification on the arcs is (l,u).

Let $\mathcal{N}$ be the network in Figure 3.14 and let $S = \{b\}$ and $\overline{S} = \{a,c\}$. Then $l(\overline{S},S) = 3 > 2 = u(S,\overline{S})$. Now using Lemma 3.8.1 we see that if x is a feasible flow in $\mathcal{N}$ we must have

$$2 = u(S,\overline{S}) \geq x(S,\overline{S}) = x(\overline{S},S) \geq l(\overline{S},S) = 3,$$

implying that there is no feasible flow in $\mathcal{N}$. More generally, our argument shows that if $\mathcal{N} = (V, A, l, u)$ is a network for which some partition $S, \overline{S}$ of V satisfies $l(\overline{S}, S) > u(S, \overline{S})$, then $\mathcal{N}$ has no feasible circulation. Hoffman [431] proved that the converse holds as well.

Before we prove Theorem 3.8.2 we remark that Theorem 3.4.2 remains valid for networks with non-zero lower bounds provided that we modify the definition of $x \oplus \tilde{x}$ slightly (see Exercise 3.30).

Theorem 3.8.2 (Hoffman's circulation theorem) *[431] A network $\mathcal{N} = (V, A, l, u)$ with non-negative lower bounds on the arcs has a feasible circulation if and only if the following holds for every proper subset S of V:*

$$l(\overline{S}, S) \le u(S, \overline{S}). \tag{3.18}$$

Proof: Let $\mathcal{N} = (V, A, l, u)$ be a network. We argued above that if x is a feasible circulation in $\mathcal{N}$, then for every partition $(S, \overline{S})$ of V we have $l(\overline{S}, S) \le u(S, \overline{S})$.

To prove the converse we assume that (3.18) holds for all $S \subset V$ and give an algorithmic proof showing how to construct a feasible circulation starting from the all-zero circulation. Clearly $x \equiv 0$ is a circulation in $\mathcal{N}$ and if $l \equiv 0$, then we are done. So we may assume that $l_{ij} > x_{ij}$ for some $ij \in A$.

We try to find a (j, i)-path in $\mathcal{N}(x)$. If such a path P exists, then we let $\delta(P) > 0$ be the minimum residual capacity of an arc on P. Let $\epsilon = \min\{\delta(P), l_{ij} - x_{ij}\}$. By Theorem 3.4.2 (which, as remarked earlier, also holds when some lower bounds are non-zero), we can increase the current flow x by ϵ units along the cycle iP and obtain a new circulation.

We claim that we can continue this process until the current circulation x has $l_{ij} \le x_{ij} \le u_{ij}$ for all arcs $ij \in A$, that is, we can obtain a feasible circulation in $\mathcal{N}$ (observe that the procedure above preserves the inequality $x \le u$). Suppose this is not the case and that at some point we have $x_{st} < l_{st}$ for some arc st and there is no (t, s)-path in $\mathcal{N}(x)$. Define T as follows:

$$T = \{r : \text{ there exists a } (t, r)\text{-path in } \mathcal{N}(x)\}.$$

It follows from the definition of the residual network $\mathcal{N}(x)$ (in particular (3.7)) that in $\mathcal{N}$ we have $x_{ij} = u_{ij}$ for all arcs ij with $i \in T$ and $j \in \overline{T}$ and $x_{qr} \le l_{qr}$ for all arcs qr with $q \in \overline{T}$ and $r \in T$. Using that $s \in \overline{T}$ and $x_{st} < l_{st}$ we obtain that

$$u(T, \overline{T}) = x(T, \overline{T}) = x(\overline{T}, T) < l(\overline{T}, T),$$

contradicting the assumption that (3.18) holds. This and the fact that all data are integers shows that the algorithm we described above will indeed find a feasible circulation in $\mathcal{N}$. □

It is not difficult to turn the proof above into a polynomial algorithm which, given a network $\mathcal{N} = (V, A, l, u)$, either finds a feasible circulation x in $\mathcal{N}$, or a subset S violating (3.18) (Exercise 3.29). □

We conclude with a remark on finding feasible flows with respect to arbitrary balance vectors in general networks. This problem is relevant as a starting point for many algorithms on flows. It follows from the results in Section 3.2 and the fact that the preflow-push algorithm can be turned into an $O(n^3)$ algorithm (using the FIFO implementation) that the following holds.

Theorem 3.8.3 *There exists an $O(n^3)$ algorithm for finding a feasible flow in a given network $\mathcal{N} = (V, A, l, u, b)$. Furthermore, if l, u, b are all integer functions, then an integer feasible flow can be found in time $O(n^3)$.* □

Using Lemma 3.2.2 and Theorem 3.8.2 one can derive the following feasibility theorem for flows by Gale (Exercise 3.44):

Theorem 3.8.4 *[289] There exists a feasible flow in the network $\mathcal{N} = (V, A, l \equiv 0, u, b)$ if and only if*

$$\sum_{s\in S} b(s) \leq u(S, \overline{S}) \qquad \text{for every } S \subset U. \tag{3.19}$$

3.9 Minimum Value Feasible (s,t)-Flows

Let $\mathcal{N} = (V, A, l, u)$ be a network with source s, sink t and non-negative lower bounds on the arcs. A **minimum** feasible (s,t)-flow in $\mathcal{N}$ is a feasible (s,t)-flow whose value is minimum possible among all feasible (s,t)-flows. Although at first glance this problem may seem somewhat artificial, it turns out that for many applications it is actually a minimum feasible flow that is sought (see e.g. Section 5.3 and Section 5.9).

To estimate the value of a minimum (s,t)-flow, let us define the **demand**, $\gamma(S, \overline{S})$ of an (s,t)-cut $(S, \overline{S})$ as the number

$$\gamma(S, \overline{S}) = l(S, \overline{S}) - u(\overline{S}, S). \tag{3.20}$$

Let x be a feasible flow. Then, by Lemma 3.5.1, for every (s,t)-cut $(S, \overline{S})$ we have

$$\begin{aligned} |x| &= x(S, \overline{S}) - x(\overline{S}, S) \\ &\geq l(S, \overline{S}) - u(\overline{S}, S) \\ &= \gamma(S, \overline{S}). \end{aligned} \tag{3.21}$$

Hence the demand of any (s,t)-cut provides a lower bound for the value of a minimum feasible (s,t)-flow. The next result shows that the minimum value of an (s,t)-flow is exactly the maximum demand of an (s,t)-cut.

Theorem 3.9.1 (Min-flow Max-demand theorem) *Let $\mathcal{N} = (V, A, l, u)$ be a network with non-negative lower bounds on the arcs. Suppose x is a minimum feasible (s,t)-flow in $\mathcal{N}$. Then*

$$|x| = \max\{\gamma(S, \overline{S}) : s \in S, t \in \overline{S}\}. \tag{3.22}$$

Furthermore we can find a minimum feasible (s,t)-flow by two applications of any algorithm for finding a maximum (s,t)-flow.

Proof: Suppose x is a feasible (s,t)-flow in $\mathcal{N}$. If $|x| = 0$, then x is clearly a minimum (s,t)-flow (since all lower bounds are non-negative). Hence we may assume that $|x| > 0$. Suppose that y is a feasible (t,s)-flow in $\mathcal{N}(x)$. Then $x \oplus y$ is a feasible flow in $\mathcal{N}$ of value $|x| - |y|$, by Theorem 3.4.2 (as we remarked in the last section, this lemma is also valid in the general case of non-zero lower bounds). Now suppose that y is a maximum (t,s)-flow in $\mathcal{N}(x)$. Apply Theorem 3.5.3 to y and $\mathcal{N}(x)$ and let $(T, \overline{T})$ be a minimum (t,s)-cut in $\mathcal{N}(x)$. The capacity of $(T, \overline{T})$ is by definition equal to $r(T, \overline{T})$, where r is the capacity function of $\mathcal{N}(x)$. By the choice of $(T, \overline{T})$ and the definition of the residual capacities we have

$$\begin{aligned}
|y| &= r(T, \overline{T}) \\
&= \sum_{ij \in (T, \overline{T})} (u_{ij} - x_{ij}) + \sum_{qp \in (\overline{T}, T)} (x_{qp} - l_{qp}) \\
&= u(T, \overline{T}) - l(\overline{T}, T) + x(\overline{T}, T) - x(T, \overline{T}) \\
&= u(T, \overline{T}) - l(\overline{T}, T) + |x|, \qquad (3.23)
\end{aligned}$$

by Lemma 3.5.1. Rearranging this, we obtain that $|x| - |y| = l(\overline{T}, T) - u(T, \overline{T})$. This implies that the flow $x \oplus y$ (whose value is $|x| - |y|$) is a minimum feasible (s,t)-flow and proves (3.22).

It remains to prove the second claim on how to find a minimum (s,t)-flow. It follows from the argument above that once we have any feasible (s,t)-flow, we can find a minimum (s,t)-flow by just one max flow calculation. On the other hand it follows from Lemma 3.2.1 and Lemma 3.2.2 that we can find a feasible (s,t)-flow in $\mathcal{N}$ (if any exists) by performing the two transformations suggested in those lemmas and then using a max flow algorithm to check whether there is a feasible flow in the last network constructed (now feasibility is with respect to the value of $b(s)$ and all lower bounds are zero). □

3.10 Minimum Cost Flows

We now turn to networks with costs on the arcs and study the following problem called the **minimum cost flow problem**: Given a network $\mathcal{N} = (V, A, l, u, b, c)$ find a feasible flow of minimum cost (recall that the cost of a flow is given by $\sum_{ij \in A} x_{ij} c_{ij}$). By the results in Section 3.2, without loss of generality, we may treat the problem only for networks with lower bound zero on all arcs and furthermore assume that we are looking for either an (s,t)-flow of value $b(s)$ or a circulation of minimum cost. However, for different applications, different flow models may be more convenient than others. Hence, except for always assuming that the lower bounds are zero, we will treat the general case, and hence all the special cases also, below.

We mentioned in Section 3.2 that the shortest path problem is a special case of the minimum cost flow problem. To see this, let $D = (V, A, c)$ be an arc weighted digraph with special vertices s, t and assume that D has no cycle of negative weight. Let $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1, c)$ be the network obtained from D by adding a lower bound of zero and a capacity of 1 to each arc of D and interpreting the weight of an arc in D as its cost in $\mathcal{N}$. We claim that a shortest (s,t)-path in D corresponds to a minimum cost integer (s,t)-flow of value 1 in $\mathcal{N}$. Clearly, any (s,t)-path P of weight M in D can be transformed into an (s,t)-flow of cost M just by sending one unit of flow along P in $\mathcal{N}$. Thus it suffices to prove that every (s,t)-flow x of value one and cost M can be transformed into an (s,t)-path in D of weight at most M. By Theorem 3.3.1 we may decompose x into a path flow of value one along an (s,t)-path P' and a number of cycle flows. All these cycles have non-negative cost since D has no negative cycle. Hence it follows that P' has cost at most M. It follows from our observations above that every minimum cost (s,t)-flow of value 1 in $\mathcal{N}$ can be decomposed into an (s,t)-path of the same cost and some cycle flows along cycles of cost zero.

In Exercise 3.47 the reader is asked to show that the maximum flow problem is also a special case of the minimum cost flow problem. However, the minimum cost flow problem is interesting not only because it generalizes these two problems, but also because a large number of practical applications can be formulated as minimum cost flow problems. The very comprehensive book by Ahuja, Magnanti and Orlin [7] contains a large number of such applications. We will discuss one of these in a reformulated form below.

A small cargo company uses a ship with a capacity to carry at most r units of cargo. The ship sails on a long route (say from Southampton to Alexandria) with several stops at ports in between. At these ports cargo may be unloaded and new cargo loaded. At each port there is an amount b_{ij} of cargo which is waiting to be shipped from port i to port $j > i$ (ports are numbered after the order in which the ship visits them). Let f_{ij} denote the income for the company from transporting one unit of cargo from port i to port j. The goal for the cargo company is to plan how much cargo to load at each port so as to maximize the total income while never exceeding the capacity of the ship. We illustrate how to model this problem, which we call the **ship loading problem**, as a minimum cost flow problem because it shows not only that sometimes it is easier to work with the general model, but also that allowing negative costs on the arcs may simplify the formulation.

Let n be the number of stops including the starting port and the terminal port. Let $\mathcal{N} = (V, A, l \equiv 0, u, c)$ be the network defined as follows:

$$V = \{v_1, v_2, \dots, v_n\} \cup \{v_{ij} : 1 \le i < j \le n\},$$
$$A = \{v_1v_2, v_2v_3, \dots, v_{n-1}v_n\} \cup \{v_{ij}v_i, v_{ij}v_j : 1 \le i < j \le n\}.$$

The capacity of the arc v_iv_{i+1} is r for $i = 1, 2, \dots n-1$ and all other arcs have capacity ∞. The cost of the arc $v_{ij}v_i$ is $-f_{ij}$ for $1 \le i < j \le n$. All other arcs have cost zero (including those of the form $v_{ij}v_j$). The balance vector of v_{ij} is b_{ij} for $1 \le i < j \le n$ and the balance vector of v_i is $-(b_{1i} + b_{2i} + \dots + b_{i-1i})$ for $i = 1, 2, \dots, n$. (See Figure 3.15.)

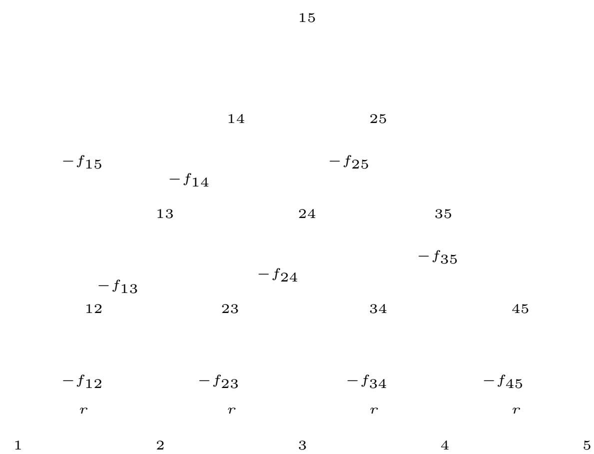


Figure 3.15 The network for the ship loading problem with 3 intermediate stops. For readability vertices are named by numbers only. The costs (capacities) are only shown when non-zero (not infinite). The balance vectors are as specified in the description in the text, i.e. the balance vector of the vertex 34 is b_{34} and the balance vector of the vertex 4 is $-(b_{14} + b_{24} + b_{34})$.

We claim that this network models the ship loading problem. Indeed, suppose that $t_{12}, t_{13}, \dots, t_{1n}, t_{23}, \dots, t_{n-1n}$ are cargo numbers, where $t_{ij} (\le b_{ij})$ denote the amount of cargo the ship will transport from port i to port j and that the ship is never loaded above capacity. The total income from these cargo loads is $I = \sum_{1 \le i < j \le n} t_{ij} f_{ij}$. Let x be the flow in $\mathcal{N}$ defined as follows. The flow on an arc of the form $v_{ij}v_i$ is t_{ij}, the flow on an arc of the form $v_{ij}v_j$ is $b_{ij} - t_{ij}$ and the flow on an arc of the form v_iv_{i+1}, $i = 1, 2, \dots, n-1$, is the sum of those t_{ab} for which $a \le i$ and $b \ge i+1$. It follows from the fact that t_{ij}, $1 \le i < j \le n$, are legal cargo numbers that x is feasible with respect to the balance vector and the capacity restriction. It is also easy to see that the cost of x is $-I$.

Conversely, suppose that x is a feasible flow in $\mathcal{N}$ of cost J. We claim that we get a feasible cargo assignment s_{ij}, $1 \leq i < j \leq n$ with income $-J$ by letting s_{ij} be the value of x on the arc $v_{ij}v_i$. This is easy to check and we leave the details to the reader. It follows that a minimum cost flow in $\mathcal{N}$ corresponds to an optimal loading of the ship and vice versa.

Below we consider the minimum cost flow problem in some detail. Further applications are given in Section 3.11. See also Section 3.12 for two important special cases of the minimum cost flow problem.

We use the notion of the **cost** of a path or a cycle in a network. This is simply the sum of the costs of all arcs in the path or cycle. An **augmenting** path (cycle) with respect to a given flow x in a network $\mathcal{N}$ is a path (cycle) in $\mathcal{N}(x)$. Whenever we speak about an augmenting cycle or path P we use $\delta(P)$ to denote the minimum residual capacity of an arc on P in $\mathcal{N}(x)$. Furthermore, for every $\beta \leq \delta(P)$ we denote by $x' := x \oplus \beta P$ the flow we obtain from x by augmenting along P with β units.

Whenever we say that a flow x is **optimal** in a network $\mathcal{N}$, we mean by this that x is a minimum cost flow among all flows in $\mathcal{N}$ with balance vector b_x.

3.10.1 Characterizing Minimum Cost Flows

Recall from Theorem 3.5.3 that, when we consider maximum (s,t)-flows, we can verify optimality by showing that there is no (s,t)-path in the residual network with respect to the current flow. It turns out that we can also use the residual network to check whether a given feasible flow in a network $\mathcal{N} = (V, A, l, u, c)$ has minimum cost among all flows with the same balance vector. Suppose first that x is feasible in $\mathcal{N}$ and that there is some cycle W in $\mathcal{N}(x)$ such that the cost $c(W)$ of W is negative. Let δ denote the minimum residual capacity of an arc on W and let x' be the cycle flow in $\mathcal{N}(x)$ which sends δ units around W. Then it follows from Theorem 3.4.2 that $x \oplus x'$ is a flow in $\mathcal{N}$ with the same balance vector as x and cost $c^T x + c^T x' = c^T x + \delta c(W) < c^T x$. Thus if $\mathcal{N}(x)$ contains a cycle of negative cost, then x is not a minimum cost feasible flow in $\mathcal{N}$ with respect to the balance vector b_x.

The interesting thing is that the other direction holds as well. Indeed, suppose x is feasible in $\mathcal{N} = (V, A, l, u, b, c)$ and that $\mathcal{N}(x)$ contains no cycle of negative cost. Let y be an arbitrary feasible flow in $\mathcal{N}$. Since we have specified a balance vector b for $\mathcal{N}$, it follows from Corollary 3.4.4 that there exist a collection of at most m cycles $W_1, W_2, \ldots, W_k$ in $\mathcal{N}(x)$ and cycle flows $f(W_1), \ldots, f(W_k)$ in $\mathcal{N}(x)$ such that $c^T y = c^T x + \sum_{i=1}^{k} c(W_i)\delta_i$, where $\delta_i > 0$ is the amount of flow that $f(W_i)$ sends along W_i in $\mathcal{N}(x)$. Since $\mathcal{N}(x)$ has no negative cost cycle, $c(W_i) \geq 0$ for $i = 1, 2, \ldots, k$ and we see that[8] $c^T y \geq c^T x$.

[8] In fact, our argument shows that $c^T y = c^T x$ if and only if y can be obtained from x by 'adding' zero or more cycle flows, each of cost zero, in $\mathcal{N}(x)$.

Thus we have established the following important optimality criterion for the minimum cost flow problem.

Theorem 3.10.1 *Let x be a feasible flow in the network $\mathcal{N} = (V, A, l, u, b, c)$. Then x is a minimum cost feasible flow in $\mathcal{N}$ if and only if $\mathcal{N}(x)$ contains no directed cycle of negative cost.* □

It is natural to ask how useful is this optimality criterion is. First observe that using the Bellman-Ford-Moore algorithm (Subsection 2.3.4) we can check whether an arbitrary given network contains a negative cycle in time $O(nm)$. Thus we obtain the following algorithm, due to Klein [480], for finding a minimum cost feasible flow in a network.

The cycle canceling algorithm
Input: A network $\mathcal{N} = (V, A, l, u, b, c)$.
Output: A minimum cost feasible flow in $\mathcal{N}$.

1. Find a feasible flow x in $\mathcal{N}$.
2. Search for a negative cycle in $\mathcal{N}(x)$.
3. If such a cycle W is found then augment x by $\delta(W)$ units along W and go to Step 2.
4. Return x.

Just as is the case for the Ford-Fulkerson algorithm, the cycle canceling algorithm may never terminate if the capacities are non-rational numbers. It is easy to modify the example in Exercise 3.17 to show this. However, if all lower bounds and capacities are integers (or just rational numbers) then this is indeed an algorithm, although not always a very fast one. See Figure 3.16 for an illustration of the algorithm.

Let U and C denote the maximum capacity of $\mathcal{N}$ and the maximum numerical value among all costs of $\mathcal{N}$.

Theorem 3.10.2 *If all lower bounds, capacities, costs and balance vectors of the input network $\mathcal{N}$ are integers, then the cycle canceling algorithm finds an optimum flow in time $O(nm^2CU)$.*

Proof: By Theorem 3.8.3 we can find a feasible flow x in $\mathcal{N}$ in time $O(n^3)$. Hence Step 1 can be performed within the promised time bound, since we assume that all networks in this chapter have $m = \Omega(n)$. The maximum possible cost of a feasible flow in $\mathcal{N}$ is mUC and the minimum possible cost is $-mUC$. Since we decrease the cost of the current flow by at least one in Step 3 it follows that after at most $O(mUC)$ executions of Step 3 we obtain a minimum cost feasible flow. Now the complexity follows from the fact that Step 2 can be performed in time $O(nm)$ using the Bellman-Ford-Moore algorithm. □

Furthermore, just as it was the case for maximum flows, we have a nice integrality property.

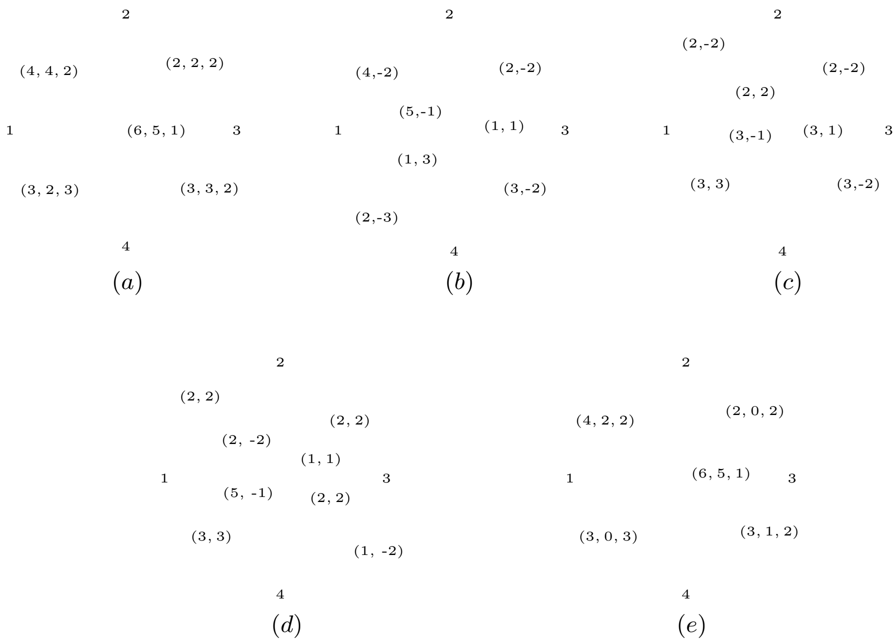


Figure 3.16 An illustration of the cycle canceling algorithm. (a) A network $\mathcal{N}$ with a feasible flow x with respect to the balance vector $(b(1), b(2), b(3), b(4)) = (2, 3, 1, -6)$. The data on the arcs are (capacity, flow, cost); (b) the residual network $\mathcal{N}(x)$. The data on the arcs are (residual capacity, cost); (c) the residual network after augmenting by 2 units along the cycle 1421; (d) the residual network after augmenting by 2 units along the cycle 2432; (e) the final optimal flow.

Theorem 3.10.3 (Integrality theorem for minimum cost flows) *If all lower bounds, capacities and balance vectors of the network $\mathcal{N}$ are integers, then there exists an integer minimum cost flow.*

Proof: This is an easy consequence of the proof of Theorem 3.10.2. By Theorem 3.8.3 we may assume that the flow x after Step 1 is an integer flow. Now the claim follows easily by induction of the number of augmentations made by the cycle canceling algorithm since in each augmentation we change the current flow by an integer amount along the arcs of the augmenting cycle. $\square$

For arbitrary networks with integer valued data the complexity of the cycle canceling algorithm is not very impressive and the algorithm is clearly not polynomial since its running time is exponential in both the maximum capacity and the maximum (absolute value of the) cost. It is easy to construct examples for which the algorithm, without some guidance as to how the next negative cycle should be chosen, may use $O(mUC)$ augmentations before it arrives at an optimum flow (Exercise 3.52). However, for several applications, such as when we are looking for certain structures in digraphs, both U and C are small and then the algorithm is quite attractive due to its simplicity (see e.g. some of the results in Section 3.11).

The problem of finding a strongly polynomial algorithm[9] for the minimum cost flow problem was posed by Edmonds and Karp [216] in 1972 and remained open until Tardos [687] found the first such algorithm in 1985. We mentioned above that if we use just any negative cycle in Step 3, then the cycle canceling algorithm may use a non-polynomial number of iterations. Goldberg and Tarjan showed that the following variant of the algorithm is already strongly polynomial [326]. The **mean cost** of a cycle W is the number $c(W)/|A(W)|$.

Theorem 3.10.4 *[326] If we always augment along a cycle of minimum mean cost (as negative mean cost as possible) in Step 3, then the cycle canceling algorithm has complexity $O(n^2m^3 \log n)$ even if some arcs have non-rational data.* □

The correctness of the algorithm, provided that it terminates, follows from Theorem 3.10.1, since there is no negative cycle in the current residual network at termination. Due to space considerations we will not prove the complexity part of the theorem here. We refer the interested reader to [7, 578] for nice accounts for the complexity of this algorithm. It is interesting to note that, although the proof of the complexity statement of Theorem 3.10.4 is quite non-trivial, it uses just the basic definitions of flows along with some new concepts which are used to make the proof smoother.

3.10.2 Building up an Optimal Solution

The cycle canceling algorithm starts from a (generally) non-optimal but feasible flow and continues through a sequence of feasible flows until an optimal flow is found (provided the algorithm ever terminates). In this subsection we describe another approach, due to Jewell [460] and Busacker and Gowen [138], in which we start from a (generally) in-feasible flow which is optimal[10] and continue through a sequence of optimal but in-feasible flows until a feasible and optimal flow is reached.

Theorem 3.10.5 (The buildup theorem) *[460, 138] Suppose that x is a minimum cost feasible flow in a network $\mathcal{N} = (V, A, l \equiv 0, u, c)$ with respect to the balance vector $b = b_x$ and let P be a minimum cost (p, q)-path in $\mathcal{N}(x)$. Let $\alpha \leq \delta(P)$ and let $f(P)$ be the path flow of value in $\mathcal{N}(x)$. Then the flow $x' := x \oplus f(P)$ is a minimum cost feasible flow in $\mathcal{N}$ with respect to the balance vector b' given by*

[9] A graph algorithm is **strongly polynomial** if (counting each arithmetic operation as constant time) the number of operations is bounded by a polynomial in n and m.

[10] Recall that optimality is with respect to flows with the same balance vector.

$$b'(v) = \begin{cases} b(v) & \textit{if } v \notin \{p, q\} \\ b(p) + \alpha & \textit{if } v = p \\ b(q) - \alpha & \textit{if } v = q. \end{cases}$$

Proof: By Theorem 3.10.1 it is sufficient to prove that there is no negative cycle in $\mathcal{N}(x')$. Since x is optimal there is no negative cycle in $\mathcal{N}(x)$. Suppose that $\mathcal{N}(x')$ contains a negative cycle W. By the definition of x', every arc in $\mathcal{N}(x')$ is either an arc of $\mathcal{N}(x)$ or the opposite of an arc on P. Consider the directed multigraph H that we obtain from $A(P) \cup A(W)$ considered as a multiset by deleting all arcs a such that both a and the opposite arc is in $A(P) \cup A(W)$. It is easy to see that if we add the arc qp to H then we obtain a directed multigraph M such that each connected component of M is eulerian. Hence, by Exercise 3.8, we can decompose $A(H)$ into a (p, q)-path P' and a number of cycles $W_1, W_2, \ldots, W_k$. It follows from our remark above and the way we defined H that all arcs of $P', W_1, W_2, \ldots, W_k$ are arcs of $\mathcal{N}(x)$. By (3.2) opposite arcs have costs which cancel and hence, using that $c(W) < 0$ we obtain

$$\begin{aligned} c(P) &> c(P) + c(W) \\ &= c(P') + \sum_{i=1}^{k} c(W_i) \\ &\geq c(P'), \end{aligned}$$

since the cost of each W_i must be non-negative because W_i is a cycle in $\mathcal{N}(x)$. Thus we see that P' is a (p, q)-path with a cost smaller than that of P, contradicting the minimality of P. Hence W cannot exist and the proof is complete. □

Based on Theorem 3.10.5 we can construct an algorithm, called the **buildup algorithm** [460, 138], for finding an optimal feasible flow in a network $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$. The algorithm described below only works if there are no negative cycles in the starting network. This restriction poses no practical problems since, according to Exercise 3.49, we may reduce the general minimum cost flow problem to the case when all costs are non-negative. Under the assumption that $\mathcal{N}$ has no negative cycles, the flow $x \equiv 0$ is an optimal circulation in $\mathcal{N}$. At any time during the execution of the buildup algorithm the sets U_x, Z_x are defined with respect to the current flow x as follows:

$$U_x = \{v | b_x(v) < b(v)\}, \qquad Z_x = \{v | b_x(v) > b(v)\}.$$

Observe that $U_x = \emptyset$ if and only if $Z_x = \emptyset$.

The buildup algorithm
Input: A network $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$.
Output: A minimum cost feasible flow in $\mathcal{N}$ with respect to b or a proof that the problem is infeasible.

1. Let $x_{ij} := 0$ for every $ij \in A$;
2. If $U_x = \emptyset$ then go to Step 8;
3. If there is no (U_x, Z_x)-path in $\mathcal{N}(x)$ go to Step 9;
4. Let p and q be chosen such that $p \in U_x, q \in Z_x$ and $\mathcal{N}(x)$ contains a (p, q)-path;
5. Find a minimum cost (p, q)-path P in $\mathcal{N}(x)$;
6. Let $\epsilon = \min\{\delta(P), b(p) - b_x(p), b_x(q) - b(q)\}$ ($\delta(P)$ is the residual capacity of P);
7. Let $x := x \oplus \epsilon P$; Modify U_x, Z_x and go to Step 2;
8. Return x;
9. Return 'no feasible solution'.

See Figure 3.17 for an illustration of the algorithm.

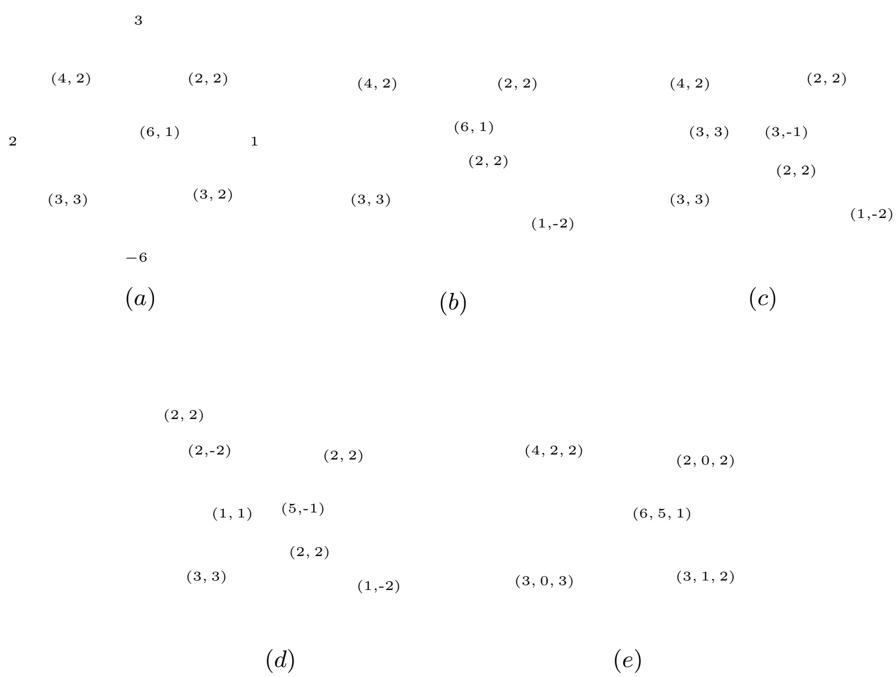


Figure 3.17 The buildup algorithm performed on the network from Figure 3.16(a). Part (a)-(d) show the current residual network with respect to the flow x, starting from $x \equiv 0$ in (a). For each arc (u, c) is specified and in (a) $b(v)$ is specified for each vertex. White circles correspond to the set U_x and white boxes correspond to Z_x. Black circles represent vertices that have reached the desired balance value. Part (e) shows the final optimal flow.

Theorem 3.10.6 *[460, 138] Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ have all data integers and no negative costs. The buildup algorithm correctly determines a min-*

imum cost feasible flow x in $\mathcal{N}$ or detects that no feasible flow exists in $\mathcal{N}$. The algorithm can be performed in time $O(n^2mM)$, where $M = \max_{v \in V} |b(v)|$. Furthermore, if there is a feasible flow in $\mathcal{N}$, then the algorithm will find an integer optimal feasible flow in $\mathcal{N}$.

Proof: Exercise 3.50. □

The following result shows that, when we consider minimum cost (s,t)-flows, the cost of successive augmenting (s,t)-paths form a monotonically increasing function. One can make a more general statement (Exercise 3.51), but for simplicity we consider only (s,t)-flows here.

Proposition 3.10.7 *Let $\mathcal{N}$ be a network with distinct vertices s,t and let x be an optimal (s,t)-flow in $\mathcal{N}$. Suppose x' is obtained from x by augmenting along a minimum cost (s,t)-path P in $\mathcal{N}(x)$ and that x'' is obtained from x' by augmenting along a minimum cost (s,t)-path P' in $\mathcal{N}(x')$. Then*

$$c^Tx - c^Tx' \geq c^Tx' - c^Tx''. \tag{3.24}$$

Proof: Let x, x', x'' and P, P' be as described in the proposition. Analogously to the way we argued in the proof of Theorem 3.10.5 we can show that the directed multigraph H' obtained from the multiset of arcs from $A(P) \cup A(P')$ by deleting arcs that are opposite in the two paths can be decomposed into two (s,t)-paths Q, R and some cycles $W_1, \ldots, W_p$ such that all arcs of these paths and cycles are in $\mathcal{N}(x)$. Since x is optimal each cycle W_i, $i = 1, 2, \ldots, p$ has non-negative cost by Theorem 3.10.1. Using that P is a minimum cost (s,t)-path in $\mathcal{N}(x)$ we conclude that each of R, Q have cost at least $c(P)$ implying that $c(P') \geq c(P)$. Hence (3.24) holds. □

3.11 Applications of Flows

In this section we illustrate the applicability of flows to a large spectrum of problems both of a theoretical and practical nature. For further applications see e.g. Section 3.12 and Chapter 7. Since we will need these results in later chapters the main focus is on finding certain substructures in digraphs.

3.11.1 Maximum Matchings in Bipartite Graphs

Let $G = (V, E)$ be an undirected graph. Recall that a matching in G is a set of edges from E, no two of which share a vertex and a maximum matching of G is a matching of maximum cardinality among all matchings of G. Matching problems occur in many practical applications such as the following scheduling problem. We are given a set $T = \{t_1, t_2, \ldots, t_r\}$ of tasks (such as handling a certain machine) to be performed and a set $P = \{p_1, p_2, \ldots, p_s\}$ of persons, each of which is capable of performing some of the tasks from T. The goal is to find a maximum number of tasks such that each task can be performed by some person who does not at the same time perform any other task and no task is performed by more than one person. This can be formulated as a matching problem as follows. Let $B = (P, T; E)$ be the bipartite graph whose vertex set is $P \cup T$ and such that for each i, j such that $1 \leq i \leq s$, $1 \leq j \leq r$, E contains the edge $p_i t_j$ whenever person p_i can perform task t_j. Now it is easy to see that the answer to the problem above is a matching in B which covers the maximum possible number of vertices in T (see also Exercise 3.53). For arbitrarily graphs finding a maximum matching fast is quite complicated and it was a great breakthrough when Edmonds [210] found a polynomial algorithm. For the case of bipartite graphs we describe a simple algorithm based on flows.

Theorem 3.11.1 *For bipartite graphs the maximum matching problem is solvable in time* $O(\sqrt{n}m)$.

Proof: Let $B = (X, Y; E)$ be an undirected bipartite graph with bipartition (X, Y). Construct a network $\mathcal{N}_B = (X \cup Y \cup \{s, t\}, A, l \equiv 0, u)$ as follows (see Figure 3.18):

$A = \{ij : i \in X, j \in Y \text{ and } ij \in E\} \cup \{si : i \in X\} \cup \{jt : j \in Y\}$, $u_{ij} = \infty$ for all $ij \in (X, Y)$, $u_{si} = 1$ for all $i \in X$ and $u_{jt} = 1$ for all $j \in Y$.

s t

B $\mathcal{N}_B$

Figure 3.18 A bipartite graph and the corresponding network. Capacities are one on all arcs of the form sv, ut and ∞ on all arcs corresponding to edges of B.

We claim that the value of a maximum (s, t)-flow in $\mathcal{N}_B$ equals the size of a maximum matching in B. To see this suppose that x is an integer flow in $\mathcal{N}$ of value k. Let $M = \{ij : i \in X, j \in Y \text{ and } x_{ij} > 0\}$. For each $i \in X$ the flow on the arc x_{si} is either zero or one. Furthermore, if $x_{si} = 1$, then it follows from the fact that x is integer valued and $b_x(i) = 0$ that precisely one arc from i to Y has non-zero flow. Similarly , for each $j \in Y$, if $x_{jt} = 1$ then precisely one arc from X to j has non-zero flow. It follows that M is a matching of size k in B and hence, by Theorem 3.5.5, the size of a maximum matching in B is at least the value of a maximum flow in $\mathcal{N}_B$.

On the other hand, if $M' = \{q_ir_i : q_i \in X, r_i \in Y, i = 1, 2, \ldots, h\}$ is a matching in B, then we obtain a feasible (s,t)-flow of value h in $\mathcal{N}_B$ by sending one unit of flow along each of the internally disjoint paths sq_ir_it, $i = 1, 2, \ldots, h$. This shows that the opposite inequality also holds and the claim follows.

It follows from the arguments above that, given a maximum integer flow x, we can obtain a maximum matching M of B by taking precisely those arcs of the form u_iv_i, $u_i \in X, v_i \in Y$ which have flow value equal to 1. Note that $\mathcal{N}_B$ is a simple network. Hence the complexity claim follows from the fact that we can find a maximum flow in $\mathcal{N}$ in time $O(\sqrt{n}m)$, using the algorithm of Theorem 3.7.7 (recall that this complexity is also valid for simple networks where not all capacities are 1, provided that at most one unit of flow can pass through any vertex distinct from s, t). □

In the case of dense graphs a slightly faster algorithm of complexity $O(n^{1.5}\sqrt{m/\log n})$ was given by Alt, Blum, Mehlhorn and Paul in [23]. It is still possible to obtain fast algorithms for finding a maximum matching in general graphs, see e.g. Tarjan's book [690]. However, it does not seem possible to formulate the maximum matching problem for an arbitrary graph as an instance of the maximum flow problem in some network. In [482] a generalization of flows which contains the maximum matching problem for general graphs as a special case was studied by Kocay and Stone.

A **vertex cover** of an undirected graph $G = (V, E)$ is a subset $U \subseteq V$ such that every edge $e \in E$ has at least one of its end vertices in U. Since no two edges of a matching share a vertex, it follows that for every vertex cover U in G, the size of U is at least the size of a maximum matching. For general graphs there does not have to be equality between the size of a maximum matching and the size of a minimum vertex cover. For instance if G is just a 5-cycle, then the size of a maximum matching is 2 and no vertex cover has less than 3 vertices. We now prove the following result, due to König [498], which shows that for bipartite graphs equality does hold. The proof illustrates the power of the Max-flow Min-cut theorem.

Theorem 3.11.2 (König's theorem) *[498] Let $B = (X, Y; E)$ be an undirected bipartite graph with bipartition (X, Y). The size of a maximum matching in B equals the size of a minimum vertex cover in B.*

Proof: Let $\mathcal{N}_B = (V \cup \{s,t\}, A, l \equiv 0, u)$ be defined as in the proof of Theorem 3.11.1. Let x be a maximum flow in $\mathcal{N}_B$ and let $(S, \overline{S})$ be the minimum cut defined as in the proof of Theorem 3.5.3 with respect to x (see Figure 3.19). Recall that S is precisely those vertices of $V \cup \{s,t\}$ which can be reached from s in $\mathcal{N}_B(x)$. Since the capacity of each arc from X to Y is ∞, there is no edge from $S \cap X$ to $\overline{S} \cap Y$ in G. Thus $U = (X \cap \overline{S}) \cup (Y \cap S)$ is a vertex cover in B. Furthermore, it follows from the definition of S that we must have $x_{si} = 1$ for all $i \in X \cap \overline{S}$ and $x_{jt} = 1$ for all $j \in Y \cap S$. This shows

$X \cap S$ $Y \cap S$

S

$x = 1$ $x = 1$

s t

$\overline{S}$

$X \cap \overline{S}$ $Y \cap \overline{S}$

Figure 3.19 The situation when a maximum flow has been found. The thick dotted arc indicates that there is no arc between the two sets $X \cap S$ and $Y \cap \overline{S}$.

that $|x| = |X \cap \overline{S}| + |Y \cap S|$. We showed in the proof of Theorem 3.11.1 that $|M^*| = |x| = |X \cap \overline{S}| + |Y \cap S|$, where M^* is a maximum matching in B. Hence $|M^*| = |U|$, implying that U is a minimum vertex cover and the proof is complete. □

Recall that a matching is perfect if it covers all vertices. We saw above that the simple proof of Theorem 3.11.1 was easily modified to a proof of König's theorem. Not surprisingly we can also derive the following characterization of the existence of a perfect matching in a bipartite graph. The result below is a slight weakening of a result (dealing with matchings that meet all vertices of one bipartition class of bipartite graphs) due to Hall [393]. For an undirected graph $G = (V, E)$ and a subset $U \subset V$, we denote by $N(U)$ the set of vertices in $V - U$ which have at least one edge to a vertex in U.

Theorem 3.11.3 (Hall's theorem) *[393] A bipartite graph $B = (X, Y; E)$ has a perfect matching if and only if $|X| = |Y|$ and the following holds:*

$$|N(U)| \geq |U| \qquad \textit{for every } U \subset X. \tag{3.25}$$

Proof: The necessity of $|X| = |Y|$ and (3.25) is clear since every vertex in U has a private neighbour in Y if B has a perfect matching.

Suppose now that (3.25) holds and let x be an integer maximum flow in the network $\mathcal{N}_B$ which is defined as in the proof of Theorem 3.11.1. If we can prove that $|x| = |X|$ then it follows from the proof of Theorem 3.11.1 that B has a perfect matching. So suppose $|x| < |X|$. By the proof of Theorem 3.11.2 we have $|x| = |X \cap \overline{S}| + |Y \cap S|$, where S is the set of vertices that are reachable from s in $\mathcal{N}_B(x)$. Since (3.25) holds and we argued in the proof of Theorem 3.11.2 that all neighbours of $X \cap S$ are in $Y \cap S$, we also have

$$|X| = |X \cap S| + |X \cap \overline{S}| \leq |Y \cap S| + |X \cap \overline{S}| = |x| < |X|,$$

a contradiction. Hence we must have $|x| = |X|$ and the proof is complete. □

3.11.2 The Directed Chinese Postman Problem

Suppose a postman has to deliver mail along all the streets in a small[11] town. Assume furthermore that on one-way streets the mail boxes are all on one side of the street, whereas for two-way streets, there are mail boxes on both sides of the street. For obvious reasons the postman wishes to minimize the distance he has to travel in order to deliver all the mail and return home to his starting point. We show below how to solve this problem in polynomial time using minimum cost flows.

We can model the problem by a directed graph $D = (V, A)$ and a weight function $w : A \rightarrow \mathcal{R}_+$ where V contains a vertex for each intersection of streets in the town and the arcs model the streets. A 2-cycle corresponds to a two-way street and an arc which is not in a 2-cycle corresponds to a one-way street in the obvious way. The weight of an arc corresponds to the length of the corresponding street. Now it is easy to see that an optimal route for the postman corresponds to a closed walk in D which traverses each arc at least once.

We have seen in Theorem 1.6.3 that if a digraph is eulerian, then it contains a closed trail which covers all arcs precisely once. Thus if D is eulerian the optimalwalk is simply a eulerian trail in D (using each arc exactly once). Below we show how to solve the general case by reducing the problem to a minimum cost circulation problem. First observe that there is no solution to the problem if D is not strongly connected, since any closed walk is strongly connected as a digraph. Hence we assume below that the digraph in question is strong, a realistic assumption when we think of the postman problem.

Let $D = (V, A)$ be a strong digraph and let c be a weight function on A. The cost $c(W)$ of a walk W is $\sum_{ij \in A} c_{ij} W_{ij}$ where W_{ij} denotes the number of times the arc ij occurs on W. Define $\mathcal{N}$ as the network $\mathcal{N} = (V, A, l \equiv 1, u \equiv \infty, c)$, that is, all arcs have lower bounds one, capacity infinity and cost equal to the weight on each arc.

Theorem 3.11.4 *The cost of a minimum cost circulation in $\mathcal{N}$ equals the minimum cost of a Chinese postman walk in D.*

Proof: Suppose W is a closed walk in D which uses each arc $ij \in A$ $W_{ij} \geq 1$ times. Then it is easy to see that we can obtain a feasible circulation of cost $c(W)$ in $\mathcal{N}$ just by sending W_{ij} units of flow along each arc $ij \in A$.

Conversely, suppose x is an integer feasible circulation in $\mathcal{N}$. Form a directed multigraph $D' = (V, A')$ by letting A' contain x_{ij} copies of the arc ij for each $ij \in A$. It follows from the fact that x is an integer circulation that D' is an eulerian directed multigraph (see Figure 3.20). Hence, by Theorem 1.6.3, D' has an eulerian tour T. The tour T corresponds to a closed walk W in D which uses each arc at least once and clearly we have $c(W) = c^T x$. $\square$

[11] This assumption is to make sure that the postman can carry all the mail in his backpack, say. Without this assumption the problem becomes much harder.

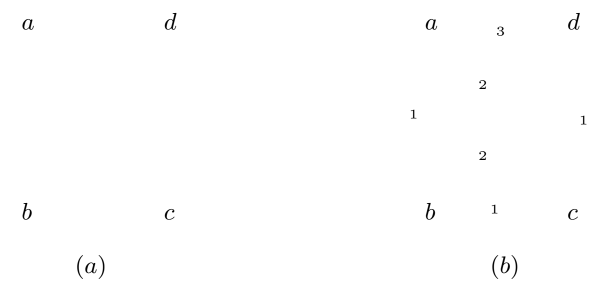


Figure 3.20 Part (a) shows a digraph with cost 1 (not shown) on every arc. Part (b) shows the values of a minimum cost circulation in the corresponding network. This circulation corresponds to the postman tour *abdacdacbda*.

3.11.3 Finding Subdigraphs with Prescribed Degrees

In some algorithms on directed multigraphs an important step is to decide whether a directed multigraph D contains a subdigraph with prescribed degrees on the vertices. One such example is when we are interested in checking whether D contains a cycle factor (see Chapter 5). Below we show that such problems and more general versions of these problems can be answered using flows. See Exercise 3.67 for another application of flows to a similar question involving construction of directed multigraphs with specified in- and out-degrees. Another application of the techniques illustrated in this subsection can be found in Section 7.16.

Theorem 3.11.5 *There exists a polynomial algorithm for the following problem. Given a directed multigraph $D = (V, A)$ with $V = \{v_1, v_2, \dots, v_n\}$ and integers $a_1, a_2, \dots, a_n$, $b_1, b_2, \dots, b_n$, find a subdigraph $D' = (V, A^*)$ of D which satisfies $d^+_{D'}(v_i) = a_i$ and $d^-_{D'}(v_i) = b_i$ for each $i = 1, 2, \dots, n$, or show that no such subdigraph exists. Furthermore, if there are costs specified for each arc, then we can also find in polynomial time the cheapest (minimum cost) subdigraph which satisfies the degree conditions.*

Proof: We may assume that $a_i \leq d^+_D(v_i)$, $b_i \leq d^-_D(v_i)$ for each $i = 1, 2, \dots, n$ and that $\sum_{i=1}^n a_i = \sum_{i=1}^n b_i$. Clearly each of these conditions is necessary for the existence of D' and they can all be checked in time $O(n)$. Let $M = \sum_{i=1}^n a_i$ and define a network $\mathcal{N}$ as follows: $\mathcal{N} = (V' \cup V'' \cup \{s, t\}, A', l \equiv 0, u)$, where $V' = \{v'_1, v'_2, \dots, v'_n\}$, $V'' = \{v''_1, v''_2, \dots, v''_n\}$ and $A' = \{sv'_i : i = 1, 2, \dots, n\} \cup \{v''_j t : j = 1, 2, \dots, n\} \cup \{v'_i v''_j : v_i v_j \in A\}$. Finally, we let $u_{sv'_i} = a_i$, $u_{v''_i t} = b_i$ for $i = 1, 2, \dots, n$ and all other arcs have capacity one.

Clearly the maximum possible value of an (s, t)-flow in $\mathcal{N}$ is M. We claim that $\mathcal{N}$ has an (s, t)-flow of value M if and only if D has the desired subdigraph.

Suppose first that $D' = (V, A^*)$ is a subdigraph satisfying $d^+_{D'}(v_i) = a_i$ and $d^-_{D'}(v_i) = b_i$ for each $i = 1, 2, \dots, n$. Then the following is an (s, t)-flow of value M in $\mathcal{N}$: $x_{sv'_i} = a_i$, $x_{v''_i t} = b_i$, for each $i = 1, 2, \ldots, n$ and $x_{v'_i v''_j}$ equals one if $v_i v_j \in A^*$ and zero otherwise.

Suppose now that x is an integer (s,t)-flow of value M in $\mathcal{N}$ and let $A^* = \{v_i v_j : x_{v'_i v''_j} = 1\}$. Then $D' = (V, A^*)$ is the desired subdigraph.

It follows from our arguments above that we can find the desired subdigraph D' in polynomial time using any polynomial algorithm for finding a maximum flow in a network.

Observe also that, if we have a cost function c on the arcs of D and let $\mathcal{N}$ inherit costs in the obvious way (arcs incident to s or t have cost zero), then finding a minimum cost subdigraph D' can be solved using any algorithm for minimum cost flows. □

It follows from Theorem 3.11.5 that we can decide whether a given digraph has a spanning k-regular subdigraph for some specified natural number k in polynomial time. In fact, using minimum cost flows we can even find the cheapest such subdigraph in the case that there are costs on the arcs. What happens if we do not require the regular subdigraph to be spanning? If $k = 1$, then the existence version of the problem is trivial, since such a subdigraph exists unless D is acyclic. Yannakakis and Alon observed that already when $k \geq 2$ the existence version of the problem becomes $\mathcal{NP}$-complete. For details see [279].

3.11.4 Path-Cycle Factors in Directed Multigraphs

We saw in the last subsection that we can use flows to find a cycle factor in a given digraph or to prove that none exists. We now show that flows are in fact very useful for studying the more general path-cycle factors in digraphs. Finding this type of subdigraph is an important ingredient in several polynomial algorithms for hamiltonian path and cycle algorithms for generalizations of tournaments (see Chapter 5).

We start with three necessary and sufficient conditions for the existence of a cycle factor in a digraph. The reason for giving all three is that in certain cases one of them provides a better way to deal with the problem under consideration than the other two. The first two parts are given in Ore's book [595]; the last is due to Yeo [748].

Proposition 3.11.6 *Let $D = (V, A)$ be a directed multigraph.*

(a) D has a cycle factor if and only if the bipartite representation $BG(D)$ of D contains a perfect matching.
(b) D has a cycle factor if and only if there is no subset X of V such that either $|\bigcup_{v \in X} N^+(v)| < |X|$ or $|\bigcup_{v \in X} N^-(v)| < |X|$.
(c) D has a cycle factor if and only if V cannot be partitioned into subsets Y, Z, R_1, R_2 such that $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$, $|Y| > |Z|$ and Y is an independent set.

Proof: (a): The reader was asked to prove (a) in Exercise 1.62, but we give the proof here for completeness. Suppose $BG(D)$ has a perfect matching consisting of edges $v_1'v_{\pi(1)}'', ..., v_n'v_{\pi(n)}''$, where π is a permutation of the set $\{1, ..., n\}$. Then the arcs $v_1v_{\pi(1)}, ..., v_nv_{\pi(n)}$ form a cycle factor. Indeed, in the digraph D' induced by these arcs every vertex v_i has out-degree and in-degree equal to one and such a digraph is precisely a disjoint union of cycles (Exercise 3.57).

Conversely, if $C_1 \cup C_2 \cup \ldots \cup C_k$ is a cycle factor in D, then for every $v_i \in V$ let $\pi(i)$ be the index of the successor of v_i on the cycle containing v_i. Then π induces a permutation of V and $\{v_iv_{\pi(i)} : v_i \in V\}$ is a perfect matching in $BG(D)$.

(b): Clearly D has a cycle factor if and only if the converse of D has a cycle factor, so it suffices to show that D has a cycle factor if and only if there is no subset X satisfying $|\bigcup_{v\in X} N^+(v)| < |X|$. Necessity is clear because if $|\bigcup_{v\in X} N^+(v)| < |X|$ holds for some X then there can be no cycle subdigraph which covers all vertices of X (there are not enough distinct out-neighbours). So suppose $|\bigcup_{v\in X} N^+(v)| \geq |X|$ holds for all $X \subset V$. Then it is easy to see that $|N(X')| \geq |X'|$ holds for every subset $X' \subset V'$ of $BG(D)$ (where $V(BG(D)) = V' \cup V''$, recall Section 1.6). It follows from Theorem 3.11.3 that $BG(D)$ has a perfect matching and now we conclude from (a) that D has a cycle factor.

(c): We first prove the necessity. Suppose D has a cycle factor $\mathcal{F}$ and yet there is a partition Y, R_1, R_2, Z as described in (c). By deleting suitable arcs from the cycles in $\mathcal{F}$ we can find a collection of $|Y|$ vertex-disjoint paths such that all these paths start in Y and end at vertices of $V - Y$ each of which dominate some vertex in Y (here we used that Y is an independent set). However this contradicts the existence of the partition Y, R_1, R_2, Z as described in (c), since it follows from the fact that $|Z| < |Y|$ that there can be at most $|Z|$ such paths in D (all such paths must pass through Z).

Now suppose that D has no cycle factor. Then we conclude from (b) that there exists a set X such that $|\bigcup_{v\in X} N^+(v)| < |X|$ holds. Let

$$Y = \{v \in X : d^-_{D\langle X\rangle}(v) = 0\}, R_1 = V - X - N^+(X), R_2 = X - Y, Z = N^+(X).$$

Then $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$ and Y is an independent set. Furthermore, since $|\bigcup_{v\in X} N^+(v)| < |X|$ we also have $|Z| + |X - Y| = |\bigcup_{v\in X} N^+(v)| < |X| = |X - Y| + |Y|$, implying that $|Z| < |Y|$. This shows that Y, Z, R_1, R_2 form a partition as in (c). $\square$

It is not difficult to show that Proposition 3.11.6 remains valid for directed pseudographs (where we allow loops) provided that we consider a loop as a cycle (Exercise 3.58). We will use that extension below.

Combining Proposition 3.11.6 with Theorem 3.11.1 we obtain

Corollary 3.11.7 *The existence of a cycle factor in a digraph can be checked and a cycle factor found (if one exists) in time $O(\sqrt{n}m)$.* $\square$

Recall that the path-cycle covering number $\mathrm{pcc}(D)$ of a directed pseudograph is the least positive integer k such that D has a k-path-cycle factor. The next result (whose proof is left as Exercise 3.68) and Theorem 3.11.1 imply that we can calculate $\mathrm{pcc}(D)$ in polynomial time for any directed pseudograph.

Proposition 3.11.8 *Let n be the number of vertices in a directed pseudograph D and let ν be the number of edges in a maximum matching of $BG(D)$. If $\nu = n$, then* $\mathrm{pcc}(D) = 1$, *otherwise* $\mathrm{pcc}(D) = n - \nu$. □

The following result by Gutin and Yeo generalizes Proposition 3.11.6(c).

Corollary 3.11.9 *[377] A digraph D has a k-path-cycle factor ($k \geq 0$) if and only if $V(D)$ cannot be partitioned into subsets Y, Z, R_1, R_2 such that $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$, $|Y| > |Z| + k$ and Y is an independent set.*

Proof: Assume that $k \geq 1$. Let D' be an auxiliary digraph obtained from D by adding k new vertices $u_1, ..., u_k$ together with the arcs $\{u_i w,\ wu_i :\ w \in V(D),\ i = 1, 2, ..., k\}$. Observe that D has a k-path-cycle factor if and only if D' has a cycle factor. By Proposition 3.11.6 (c), D' has a cycle factor if and only if its vertex set cannot be partitioned into sets Y, Z', R_1, R_2 that satisfy $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$, $|Y| > |Z'|$ and Y is an independent set. Note that if Y, Z', R_1, R_2 exist in D' then the vertices $u_1, ..., u_k$ are in Z'. Let $Z = Z' - \{u_1, ..., u_k\}$. Clearly, the subsets Y, Z, R_1, R_2 satisfy $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$, $|Y| > |Z| + k$ and Y is an independent set. □

The proof above and Corollary 3.11.7 easily implies the first part of the following proposition.

Proposition 3.11.10 *Let D be a directed pseudograph and let k be a fixed non-negative integer. Then*

(a) In time $O(\sqrt{n}m)$ we can check whether D has a k-path-cycle-factor and construct one (if it exists).
(b) Given a k-path-cycle factor in D, in time $O(m)$, we can check whether D has a $(k-1)$-path-cycle factor and construct one (if it exists).

Proof: Exercise 3.69. □

3.11.5 Cycle Subdigraphs Covering Specified Vertices

In the solution of several algorithmic problems, such as finding the longest cycle in an extended semicomplete digraphor a semicomplete bipartite digraph, it is an important subproblem to find a cycle subdigraph which covers as many vertices as possible. Below we show how to solve this problem using a reduction to the assignment problem, due to Alon (see [363]).

Theorem 3.11.11 *There is an $O(n^3)$ algorithm which finds, for any given digraph D, a cycle subdigraph covering the maximum number of vertices in D.*

Proof: Let D be a digraph and let D' be the directed pseudograph one obtains by adding a loop at every vertex. Let B be the weighted bipartite graph one obtains from the bipartite representation $BG(D')$ of D by adding the following weights to the edges: the weight of an edge xy' of B equals 1 if $x \neq y$ and equals 2 if $x = y$. It is easy to see (Exercise 3.63) that, by solving the assignment problem for B (in time $O(n^3)$, see Section 3.12) and then removing all the edges with weight 2 from the solution, we obtain a set of edges of B corresponding to some 1-regular subdigraph F of D of maximum order. $\square$

Jackson and Ordaz [452] proved the following sufficient condition for the existence of a cycle factor in a digraph. (For undirected graphs the analogous condition implies that the graph has a hamiltonian cycle [161].)

Proposition 3.11.12 *[452] If D is a k-strong digraph such that the maximum size of an independent set in D is at most k, then D has a spanning cycle subdigraph.*

We now prove a generalization of this result and discuss its relevance to the problem of finding a cycle through a specified set of vertices in certain generalizations of tournaments. Deciding whether there is a cycle containing all vertices from a prescribed set X in an arbitrary digraph is an $\mathcal{NP}$-complete problem already when $|X| = 2$ (see Theorems 9.2.3 and 9.2.6). Proposition 3.11.12 corresponds to the special case $X = V$ in the following theorem, due to Bang-Jensen, Gutin and Yeo.

Theorem 3.11.13 *[70] Let $D = (V, A)$ be a k-strong digraph and let $X \subset V(D)$ be such that $\alpha(D\langle X\rangle) \leq k$, then D has a cycle subdigraph (not necessarily spanning) covering X.*

Proof: This can be proved directly from Theorem 3.8.2 (Exercise 3.65). We give a simple proof based on Proposition 3.11.6 which also holds for directed pseudographs (see Exercise 3.58).

Let D and X be as defined in the theorem. Form the directed pseudograph D' from D by adding a loop at each vertex not in X. Then D has a cycle subdigraph covering X if and only if D' has a cycle factor, because the new arcs cannot contribute to cycles which cover vertices from X. Suppose D' has no cycle factor. Then by Proposition 3.11.6 (c) we can partition the vertices of V into sets R_1, R_2, Y, Z so that $(Y, R_1) = \emptyset$, $(R_2, R_1 \cup Y) = \emptyset$, $|Y| > |Z|$ and Y is an independent set. Note that no vertex with a loop can be in an independent set (see Section 1.6 for the definition of an independent set of vertices). Thus we have $Y \subseteq X$. It follows from the description of the arcs between the sets above that there is no path from Y to R_1 in $D-Z$. Thus we must have $|Z| \geq k$ since D is k-strong. But now we have the contradiction

$$k \leq |Z| < |Y| \leq \alpha(D\langle X\rangle) \leq k.$$

Thus D' has a cycle factor, implying that D has a cycle subdigraph covering X. □

Theorem 3.11.13 shows that the obvious necessary condition for the existence of a cycle covering a specified subset X, namely that there exists some collection of disjoint cycles covering X is satisfied in many cases. Indeed, if D is k-strong, then we may take X arbitrarily large, provided its independence number stays below $k+1$.

We point out that, when $|X| = k$ and D is k-strong, then the existence of a cycle subdigraph covering X can also be proved easily using Menger's theorem (Theorem 7.3.1). See Exercise 7.17.

The proof above combined with that of Theorem 3.11.11 immediately implies the following result.

Theorem 3.11.14 *There exists an $O(n^3)$ algorithm for checking whether a given digraph $D = (V, A)$ with a prescribed subset $X \subseteq V$ has a cycle subdigraph covering X.* □

3.12 The Assignment Problem and the Transportation Problem

In this section we study two special cases of the minimum cost flow problem, both of which occur frequently in practical applications. Being special cases of the minimum cost flow problem, they can be solved using any of the algorithms described in Section 3.10. The purpose of this section is to illustrate a general approach, the primal dual algorithm, for solving linear programming problems while using the transportation problem as an example. In order to read parts of this section the reader is supposed to have some basic knowledge of linear programming and the duality theorem for linear programming (see e.g. the book [600] by Papadimitriou and Steiglitz).

In the **assignment problem**, the input consists of a set of persons $P_1, P_2, \ldots, P_n$, a set of jobs $J_1, J_2, \ldots, J_n$ and an $n \times n$ matrix $M = [M_{ij}]$ whose entries are non-negative integers. Here M_{ij} is a measure for the skill of person P_i in performing job J_j (the lower the number the better P_i performs job J_j). The goal is to find an assignment π of persons to jobs so that each person gets exactly one job and the sum $\sum_{i=1}^{n} M_{i\pi(i)}$ is minimized. Note that it is easy to formulate the weighted bipartite matching problem (given a complete[12] undirected bipartite graph $K_{n,n}$ with weights

[12] Assuming that the graph is complete is no restriction since we can always replace non-edges by edges of weight ∞.

on its edges, find a perfect matching of minimum total weight) as an instance of the assignment problem. On the other hand, it is also easy to see that, given any instance of the assignment problem, we may form a complete bipartite graph $B = (U, V; E)$ where $U = \{P_1, P_2, \ldots, P_n\}$, $V = \{J_1, J_2, \ldots, J_n\}$ and E contains the edge P_iJ_j with the weight M_{ij} for each $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$. This shows that the assignment problem is equivalent to the weighted bipartite matching problem.

It is also easy to see from this observation that the assignment problem is a (very) special case of the minimum cost flow problem. In fact, if we think of M_{ij} as a cost, then what we are seeking is a flow of minimum cost so that the balance vector is one for each P_i, $i = 1, 2, \ldots, m$ and the balance vector is minus one for each J_j, $j = 1, 2, \ldots, n$.

In the **transportation problem** we are given a set of production plants $S_1, S_2, \ldots, S_m$ who produce a certain product to be shipped to a set of retailers $T_1, T_2, \ldots, T_n$. For each pair (S_i, T_j) there is a real-valued cost c_{ij} of transporting one unit of the product from S_i to T_j. Each plant produces a_i, $i = 1, 2, \ldots, m$, units per time unit and each retailer needs b_j, $j = 1, 2, \ldots, n$, units of the product per time unit. We assume below that $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ (this is no restriction of the model as shown in Exercise 3.71). The goal is to find a transportation schedule for the whole production (i.e. how many units to send from S_i to T_j for $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$) in order to minimize the total transportation cost.

Again the transportation problem is easily seen to be a special case of the minimum cost flow problem. Consider a bipartite network $\mathcal{N}$ with bipartition classes $S = \{S_1, S_2, \ldots, S_m\}$ and $T = \{T_1, T_2, \ldots, T_n\}$ and all possible arcs from S to T where the capacity of the arc S_iT_j is ∞ and the cost of sending one unit of flow along S_iT_j is c_{ij}. Now it is easy to see that an optimal transportation schedule corresponds to a minimum cost flow in $\mathcal{N}$ with respect to the balance vectors

$$b(S_i) = a_i, i = 1, 2, \ldots, m, \text{ and } b(T_j) = -b_j, j = 1, 2, \ldots, n.$$

The fact that both the assignment problem and the transportation problem are special cases of the minimum cost flow problem allows one to use any algorithm for finding a minimum cost flow to solve these problems. Below we are going to describe how to obtain more efficient algorithms for the transportation problem and the assignment problem by using the so-called primal-dual algorithm approach to linear programming problems. First we formulate the transportation problem as a linear programming problem.

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1, 2, \ldots, m \end{aligned} \tag{3.26}$$

$$\begin{aligned}&\sum_{i=1}^{m} x_{ij} = b_j \quad , j = 1, 2, \dots, n\\&x_{ij} \geq 0 \quad \text{for all } i, j.\end{aligned}$$

The linear programming dual of the transportation problem is

$$\begin{aligned}\max \quad & \sum_{i=1}^{m} \alpha_i a_i + \sum_{j=1}^{n} \beta_j b_j \\ \text{s.t.} \quad & \alpha_i + \beta_j \leq c_{ij} \quad \text{for all } i, j \\ & \alpha_i, \beta_j \text{ unrestricted} \quad \text{for all } i, j.\end{aligned} \tag{3.27}$$

Here the dual variables $\alpha_1, \dots, \alpha_m$ correspond to the first set of equalities and the dual variables $\beta_1, \dots, \beta_n$ correspond to the second set of equalities in the transportation problem.

Assume that we are given a feasible solution $\alpha_1, \dots, \alpha_m, \beta_1, \dots, \beta_n$ to the dual (3.27) and define a set $\mathcal{IJ}$ of indices by $\mathcal{IJ} = \{(i, j) : \alpha_i + \beta_j = c_{ij}\}$. Suppose that x is a feasible solution to the transportation problem and that $x_{ij} = 0$ for all $(i, j) \notin \mathcal{IJ}$. Then we have

$$\begin{aligned}\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} &= \sum_{(i,j) \in \mathcal{IJ}} c_{ij} x_{ij} \\ &= \sum_{(i,j) \in \mathcal{IJ}} (\alpha_i + \beta_j) x_{ij} \\ &= \sum_{i=1}^{m} \alpha_i \Big(\sum_{\{j:(i,j) \in \mathcal{IJ}\}} x_{ij} \Big) + \sum_{j=1}^{n} \beta_j \Big(\sum_{\{i:(i,j) \in \mathcal{IJ}\}} x_{ij} \Big) \\ &= \sum_{i=1}^{m} \alpha_i a_i + \sum_{j=1}^{n} \beta_j b_j .\end{aligned}$$

Combining this with the weak duality theorem for linear programming[13] shows that x is an optimal solution to the transportation problem.

In order to study how to use this observation algorithmically, we define the **restricted primal** problem with respect to the given dual solution (α, β):

[13] When the primal is a minimization problem, then the value of the dual objective function is at most the value of the primal objective function for any pair of feasible solutions to the dual and the primal.

$$\begin{aligned}
\min \quad & \sum_{i=1}^{m+n} r_i \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} + r_i = a_i, \quad i = 1, 2, \ldots, m \\
& \sum_{i=1}^{m} x_{ij} + r_{m+j} = b_j, \quad j = 1, 2, \ldots, n \\
& x_{ij} \geq 0 \quad \text{for all } (i, j) \in \mathcal{IJ} \\
& x_{ij} = 0 \quad \text{for all } (i, j) \notin \mathcal{IJ} \\
& r_i \geq 0, \quad i = 1, 2, \ldots, m + n.
\end{aligned} \tag{3.28}$$

The variables $r_1, r_2, \ldots, r_{m+n}$ are usually called **slack variables**. They ensure that (3.28) always has a feasible solution. Furthermore, the optimum in (3.28) is zero if and only if (3.26) has a feasible solution. The dual of (3.28), called the **dual of the restricted primal** problem, is as follows:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{m} \alpha_i a_i + \sum_{j=1}^{n} \beta_j b_j \\
\text{s.t.} \quad & \alpha_i + \beta_j \leq 0 \qquad \text{for all } (i, j) \in \mathcal{IJ} \\
& \alpha_i, \beta_j \leq 1 \quad \text{for all } i, j.
\end{aligned} \tag{3.29}$$

Let x, r be an optimal solution to the restricted primal problem (that is, one that minimizes $\sum_{i=1}^{m+n} r_i$). Observe that if $r \equiv 0$, then x is also a feasible solution to the transportation problem and since $x_{ij} = 0$ for all $(i, j) \notin \mathcal{IJ}$, we see from the argument above that x is in fact an optimal solution to the transportation problem. Furthermore, it follows from (3.28) that minimizing $\sum_{i=1}^{m+n} r_i$ is equivalent to the following maximization problem:

$$\begin{aligned}
\max \quad & \sum_{(i,j) \in \mathcal{IJ}} x_{ij} \\
\text{s.t.} \quad & \sum_{i=1}^{m} x_{ij} \leq a_i \;\; i = 1, 2, \ldots, m \\
& \sum_{j=1}^{n} x_{ij} \leq b_j \;\; j = 1, 2, \ldots, n \\
& x_{ij} \geq 0 \text{ for } (i, j) \in \mathcal{IJ} \\
& x_{ij} = 0 \text{ for } (i, j) \notin \mathcal{IJ}.
\end{aligned} \tag{3.30}$$

This is just a maximum flow problem. Indeed, let $\mathcal{N}_{(\alpha,\beta)} = (V, A, l \equiv 0, u)$ be the network whose vertices are $V = X \cup Y \cup \{s,t\}$, where $X = \{s_1, s_2, \ldots, s_m\}$ and $Y = \{t_1, t_2, \ldots, t_n\}$ and whose arcs are $A = \{ss_i : i = 1,2,\ldots,m\} \cup \{t_jt : j = 1,2,\ldots,n\} \cup \{s_it_j : (i,j) \in \mathcal{IJ}\}$. The capacity of the arc ss_i is a_i, $i = 1,2,\ldots,m$, the capacity of the arc t_jt is b_j, $j = 1,2,\ldots,n$ and the capacity of each arc of the form s_it_j is ∞. We call $\mathcal{N}_{(\alpha,\beta)}$ the **admissible network with respect to $(\boldsymbol{\alpha}, \boldsymbol{\beta})$**. It is not difficult to show that there is a 1-1 correspondence between maximum (s,t)-flows in $\mathcal{N}_{(\alpha,\beta)}$ and optimal solutions to (3.30).

What do we do if the value of the maximum (s,t)-flow in $\mathcal{N}_{(\alpha,\beta)}$ is strictly smaller than $\sum_{i=1}^{m} a_i$ (recall that this is equivalent to saying that the optimum value in (3.28) is strictly greater than zero)? In this case x, restricted to the arcs $\{s_it_j : (i,j) \in \mathcal{IJ}\}$, is not a feasible solution to the transportation problem. However, this is where the main step in the primal-dual algorithm comes into play. We now show that in this case it is always possible to modify the current dual solution (α, β) to a new feasible dual solution (α', β') in such a way that the value of a maximum (s,t)-flow in the network $\mathcal{N}_{(\alpha',\beta')}$ is at least as large as the corresponding value in $\mathcal{N}_{(\alpha,\beta)}$. Furthermore, if it is the same, then after a finite number of repetitions of dual solution changes, the value of a maximum flow in the current admissible network will increase.

Let x be a maximum flow in $\mathcal{N}_{(\alpha,\beta)}$ and suppose that $|x| < \sum_{i=1}^{m} a_i$. Let S be the set of vertices that are reachable from s in $\mathcal{N}_{(\alpha,\beta)}(x)$. Let $I = \{1,2,\ldots,m\}$, $J = \{1,2,\ldots,n\}$ and define I^*, J^* by

$$I^* = \{i \in I : s_i \in S\}; \quad J^* = \{j \in J : t_j \in S\}.$$

As we saw in the proof of Theorem 3.5.3, $(S, \overline{S})$ is a minimum (s,t)-cut in $\mathcal{N}_{(\alpha,\beta)}$. In particular, since all arcs of the form s_it_j have capacity ∞, there is no pair $(i,j) \in \mathcal{IJ}$ for which $i \in I^*$ and $j \in J - J^*$ (compare this with the proof of Theorem 3.11.2). Thus, arguing as we did in the proof of Theorem 3.11.2 and using Theorem 3.5.3 we obtain

$$|x| = \sum_{i \in I - I^*} a_i + \sum_{j \in J^*} b_j. \tag{3.31}$$

Going back to the problem (3.30) and using the fact that $|x|$ is exactly the value of an optimal solution to this problem, we see from (3.31) that the optimal solution for the current problem (3.28) is given by

$$\begin{aligned}
\min \sum_{i=1}^{n+m} r_i &= \sum_{i=1}^{m} a_i + \sum_{j=1}^{n} b_j - 2|x| \\
&= \sum_{i=1}^{m} a_i + \sum_{j=1}^{n} b_j - 2\Big(\sum_{i \in I - I^*} a_i + \sum_{j \in J^*} b_j\Big)
\end{aligned}$$

$$= \sum_{i\in I^*} a_i - \sum_{i\in I-I^*} a_i + \sum_{j\in J-J^*} b_j - \sum_{j\in J^*} b_j.$$

This implies that the following feasible solution $(\bar{\alpha}, \bar{\beta})$ is optimal for (3.29):

$$\bar{\alpha}_i^* = \begin{cases} 1 & \text{if } i \in I^* \\ -1 & \text{if } i \in I - I^* \end{cases}$$
$$\bar{\beta}_j^* = \begin{cases} -1 & \text{if } j \in J^* \\ 1 & \text{if } j \in J - J^*. \end{cases} \tag{3.32}$$

Let

$$\begin{aligned} \epsilon &:= \min\{\frac{c_{ij} - \alpha_i - \beta_j}{\bar{\alpha}_i + \bar{\beta}_j} : (i,j) \notin \mathcal{IJ} \text{ and } \bar{\alpha}_i + \bar{\beta}_j > 0\} \qquad (3.33) \\ &= \min\{\frac{c_{ij} - \alpha_i - \beta_j}{2} : i \in I^*, j \in J - J^*\}, \end{aligned}$$

and define (α^*, β^*) as follows:

$$\alpha_i^* = \begin{cases} \alpha_i + \epsilon & \text{if } i \in I^* \\ \alpha_i - \epsilon & \text{if } i \in I - I^* \end{cases}$$
$$\beta_j^* = \begin{cases} \beta_j - \epsilon & \text{if } j \in J^* \\ \beta_j + \epsilon & \text{if } j \in J - J^*. \end{cases} \tag{3.34}$$

It follows easily from the fact that $|x| < \sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ that $I^* \neq \emptyset$ and $J - J^* \neq \emptyset$. Furthermore, since there is currently no arc $s_i t_j$ with $(i,j) \notin \mathcal{IJ}$, we have $c_{ij} - \alpha_i - \beta_j > 0$ for all such pairs (i,j). This shows that ϵ exists and is strictly greater than zero.

Lemma 3.12.1 *Let $\alpha, \beta, \alpha^*, \beta^*$ be as above. Then the following holds:*

(a) (α^*, β^*) is a feasible solution to the dual (3.27) of the transportation problem.
(b) For every arc $s_i t_j$ in $\mathcal{N}_{(\alpha,\beta)}$ such that x is non-zero the arc $s_i t_j$ is also an arc of $\mathcal{N}_{(\alpha^*,\beta^*)}$
(c) The network $\mathcal{N}_{(\alpha^*,\beta^*)}$ contains at least one arc $s_i t_j$ for which $i \in I^*$ and $j \in J - J^*$.
(d) The value of a maximum (s,t)-flow in $\mathcal{N}_{(\alpha^*,\beta^*)}$ is at least as large as the value of the current maximum flow x in $\mathcal{N}_{(\alpha,\beta)}$.

Proof: Exercise 3.72. □

Putting the observations we made above together, we obtain the following algorithm for the transportation problem.

The primal-dual algorithm for the transportation problem
Input: An instance of the transportation problem.
Output: An optimal transportation schedule[14].

1. Initialize the dual variables as follows:
   For $i := 1$ to m let $\alpha_i := \min\{c_{ij} : j = 1, 2, \dots, m\}$;
   For $j := 1$ to n let $\beta_j := \min\{c_{ij} - \alpha_i : i = 1, 2, \dots, m\}$;
2. Construct the admissible network $\mathcal{N}_{(\alpha,\beta)}$;
3. Find a maximum flow x in $\mathcal{N}_{(\alpha,\beta)}$;
4. If $|x| = \sum_{i=1}^{m} a_i$ then return x;
5. Update the dual variables according to (3.33) and (3.34);
6. Construct the new admissible network and go to Step 3.

Theorem 3.12.2 *The primal-dual algorithm will find an optimum solution for any given transportation problem with m plants and n retailers in time $O(M(n+m)^2)$, where $M = \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$.*

Proof: We give a brief sketch which gives a complexity of $O(M(n+m)^3)$. In Exercise 3.74 the reader is asked to show how to implement the algorithm so that one obtains the desired complexity.

It is easy to check that the dual variables which are calculated in Step 1 form a feasible solution and that the admissible network will contain at least one arc from X to Y. Forming $\mathcal{N}_{(\alpha,\beta)}$ can be done in time $O((n+m)^2)$ and we can find the first maximum flow in time $O((n+m)^2 M)$ using the Ford-Fulkerson algorithm (see Theorem 3.5.4).

We can easily construct $\mathcal{N}_{(\alpha^*,\beta^*)}$ from $\mathcal{N}_{(\alpha,\beta)}$ in time $O((n+m)^2)$. By Lemma 3.12.1(b) we do not have to start all over when we wish to calculate a maximum flow in the updated admissible network $\mathcal{N}_{(\alpha^*,\beta^*)}$. In fact, the current flow x (interpreted in the obvious way) is a feasible (s,t)-flow in $\mathcal{N}_{(\alpha^*,\beta^*)}$. Thus starting from x and searching for an augmenting path in the residual network, we can either find an augmenting path or detect that the current x is still maximum in time $O((n+m)^2)$. This and the fact that we always augment by an integer amount of flow implies that, in order to prove the complexity $O(M(n+m)^3)$ for the algorithm, it suffices to show that the number of changes in the dual variables between two consecutive increases in the value of the maximum flow in the admissible network is at most m.

Suppose that the current flow x has value less than $\sum_{i=1}^{m} a_i$ and let us estimate the number of times we can change the dual variables without enabling an increase in the flow value. Let (α,β) be the actual dual variables, let S be the set of vertices that are reachable from s in $\mathcal{N}_{(\alpha,\beta)}(x)$ and define

[14] In the form of an optimal flow, from which the schedule can be read out easily.

S^* similarly for $\mathcal{N}_{(\alpha^*,\beta^*)}$. By Lemma 3.12.1(b), no arc which carries flow disappears when we change from $\mathcal{N}_{(\alpha,\beta)}$ to $\mathcal{N}_{(\alpha^*,\beta^*)}$. It is easy to show that this implies that $S \subset S^*$. By Lemma 3.12.1(c) we add at least one new arc s_it_j such that $s_i \in S$ and $t_j \in \overline{S}$ (in $\mathcal{N}_{(\alpha,\beta)}$ there is no such arc since they all have infinite capacity) and hence we obtain that $|S^* \cap Y| > |S \cap Y|$. Since $|Y| = m$ it follows that after at most m changes of dual variables we can increase the flow in the current admissible network. □

For the assignment problem we have $n = m$ and $M = n$, implying that the following holds (see also Exercise 3.76).

Theorem 3.12.3 *The assignment problem on n persons and n jobs is solvable in time $O(n^3)$.* □

For the assignment problem the $O(n^3)$ implementation of the primal-dual algorithm above is due to Kuhn [505] and is also known under the name **the Hungarian method**. The interested reader can find more details on the implementation of the primal-dual algorithm for the transportation and the assignment problems in e.g. the book [578] by Murty.

In practice it is not necessary to work explicitly on the network $\mathcal{N}_{(\alpha,\beta)}$. Suppose we keep a table containing the following information: the cost matrix, the supplies and demands for the actual instance of the transportation problem and the actual values of the dual variables (α, β). These can all be kept compactly as shown below.

							α	a
	5	6	5	2	8	4	2	6
	12	10	11	11	11	12	10	6
	3	4	5	5	4	3	3	4
	6	8	10	1	4	3	1	10
β	0	0	1	0	1	0		
b	5	4	3	4	5	5		

The cost matrix can be found in the upper left part of the diagram. Each cell corresponding to an entry in the matrix is divided into an upper and a lower part. In the lower part we have specified the cost c_{ij} of sending one unit from plant i to retailer j. No numbers are specified in the upper halves of each cell at this point (see below). The values of the supplies and demands are specified as the vectors a (in the rightmost column) and b (in the bottom row of the diagram). There is also a column which specifies the initial value of the α vector and a row specifying the initial value of the β vector. These have been calculated according to Step 1 of the primal-dual algorithm. Finally, shaded cells indicate the set $\mathcal{IJ}$.

Equipped with such a diagram we may first find a feasible flow x which may or may not be maximum in the current admissible network, e.g. by a greedy approach. The search for a new augmenting path with respect to x can also be modeled by adding a small amount of information to the diagram. Namely, we show labels which indicate how a search might progress. We start by labelling those rows $i \leq m$ for which $b_x(i) < a_i$ by '$s, +$'. Then we search for an augmenting (s,t)-path as follows (compare this with the proof of Theorem 3.5.4):

> If a row i' is labelled then every column j' for which the cell $i'j'$ is admissible (the corresponding arc is an arc of the admissible network) may be labelled (capacity is ∞ here). We label such a column by '$i', +$'.
> If a column j is already labelled and $x_{ij} > 0$, then we may label the row i by '$j, -$'.
> If at some point we label a column j for which $b_x(j) < b_j$ then we have a breakthrough: an augmenting path corresponding to the labels we can trace backwards from j has been found. In this case we augment the flow as much as possible, delete all labels and start the labelling process again.
> If no more rows or columns can be labelled, the process stops.

It is easy to see that the description above is merely a specification of the Ford-Fulkerson algorithm on the residual network with respect to x and the current admissible network.

When a maximum flow in $\mathcal{N}_{(\alpha,\beta)}$ has been found and it has a value less than $\sum_{i=1}^{m} a_i$, the primal-dual algorithm updates the dual variables. Given the labels above we can easily identify the sets I^*, J^* as the set of labeled rows and columns and calculate the new dual variables (α^*, β^*) according to (3.34). Note that in order to avoid fractional values of α^*, β^* it is more convenient to use the following choice for the new dual variables α^*, β^* (here ϵ is as defined in (3.33)). In Exercise 3.77 the reader is asked to show that this choice for α^*, β^* still gives a feasible solution and one which has a higher value for the objective function in (3.27).

$$
\begin{aligned}
\alpha_i^* &= \begin{cases} \alpha_i + 2\epsilon & \text{if } i \in I^* \\ \alpha_i & \text{if } i \in I - I^* \end{cases} \\
\beta_j^* &= \begin{cases} \beta_j - 2\epsilon & \text{if } j \in J^* \\ \beta_j & \text{if } j \in J - J^*. \end{cases}
\end{aligned}
\tag{3.35}
$$

Below we show a diagram representation of the algorithm on the example above, starting from a maximum flow in the network $\mathcal{N}_{(\alpha,\beta)}$. Recall that shaded cells indicate the arcs of the current admissible network.

							α	a	
	5	6	5	^{4}2	8	4	2	6	s,+
	12	^{4}10	^{2}11	11	11	12	10	6	
	^{4}3	4	5	5	4	3	3	4	
	6	8	10	1	4	3	1	10	s,+
β	0	0	1	0	1	0			
b	5	4	3	4	5	5			
				1,+					

No augmenting path found so we make a dual change:

$$\begin{aligned}
\epsilon_1 &= \min\{\text{5-2-0,6-2-0,}\underline{\text{5-2-1}}\text{,8-2-1,}\underline{\text{4-2-0}}\} = 2 \\
\epsilon_4 &= \min\{\text{6-1-0,8-1-0,10-1-1,}\underline{\text{4-1-1}}\text{,}\underline{\text{3-1-0}}\} = 2 \\
2\epsilon &= \min\{\epsilon_1, \epsilon_4\} = 2.
\end{aligned}$$

The new diagram, with updated dual variables and admissible arcs indicated by shaded cells, together with the new labelling step is shown below:

							α	a	
	5	6	5	^{4}2	8	4	4	6	s, +
	12	^{4}10	^{2}11	11	11	12	10	6	
	^{4}3	4	5	5	4	3	3	4	
	6	8	10	1	4	3	3	10	s, +
β	0	0	1	-2	1	0			
b	5	4	3	4	5	5			
			1, +	1, +		1, +			

breakthrough

Augment along each of the paths ss_1t_3t and ss_1t_6t by one unit along each. After this columns 4, 5 and 6 can be labelled '$4, +$' and now we can send 5 units along ss_4t_5t and 4 units along ss_4t_6t. After these augmentations the next labelling step results in the following labels:

							α	a	
	5	6	$^{1}5$	$^{4}2$	8	$^{1}4$	4	6	4,-
	12	$^{4}10$	$^{2}11$	11	11	12	10	6	3,-
	$^{4}3$	4	5	5	4	3	3	4	
	6	8	10	1	$^{5}4$	$^{4}3$	3	10	s, +
β	0	0	1	-2	1	0			
b	5	4	3	4	5	5			
		2,+	1,+	4,+	4,+	4,+			

No augmenting path found so we make a dual change:

$$\begin{aligned}
\epsilon_1 &= \min\{\text{5-4-0}\} = 1 \\
\epsilon_2 &= \text{12-10-0} = 2 \\
\epsilon_4 &= \text{6-3-0} = 3 \\
2\epsilon &= \min\{\epsilon_1, \epsilon_2, \epsilon_4\} = 1.
\end{aligned}$$

							α	a	
	5	6	$^{1}5$	$^{4}2$	8	$^{1}4$	5	6	4,-
	12	$^{4}10$	$^{2}11$	11	11	12	11	6	
	$^{4}3$	4	5	5	4	3	3	4	
	6	8	10	1	$^{5}4$	$^{4}3$	4	10	s, +
β	0	-1	0	-3	0	-1			
b	5	4	3	4	5	5			
	1,+		1,+	4,+	4,+	4,+			

breakthrough

Now we can augment by one unit along the path $ss_4t_4s_1t_1t$.

							α	a
	$^{1}5$	6	$^{1}5$	$^{3}2$	8	$^{1}4$	5	6
	12	$^{4}10$	$^{2}11$	11	11	12	11	6
	$^{4}3$	4	5	5	4	3	3	4
	6	8	10	$^{1}1$	$^{5}4$	$^{4}3$	4	10
β	0	-1	0	-3	0	-1		
b	5	4	3	4	5	5		

A feasible solution to (3.26) has been found. Control for optimality:

$$\sum_{i=1,j+1}^{m,n} c_{ij}x_{ij} = 5+5+6+4+40+22+12+1+20+12 = \underline{127}$$

$$\sum_{i=1}^{m} \alpha_i a_i + \sum_{j=1}^{n} \beta_j b_j = 30+66+12+40-4-12-5 = \underline{127}.$$

Above we have illustrated the primal-dual algorithm when applied to the transportation problem. We would like to stress that this approach is quite general. It works for any linear programming problem and its dual, provided that both problems have feasible solutions. We refer the reader to the book by Papadimitriou and Steiglitz [600] for an excellent account of the primal-dual algorithm approach.

3.13 Exercises

Unless otherwise stated, all numerical data in the exercises below are integers.

3.1. Find a feasible flow in the network $\mathcal{N}$ of Figure 3.21.

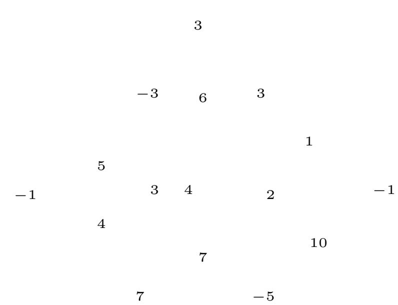


Figure 3.21 A network $\mathcal{N}$ with balance vector b specified at each vertex. All lower bounds and costs are zero and capacities are shown on the arcs.

3.2. Suppose the network $\mathcal{N} = (V, A, l, u, b, c)$ has some 2-cycle iji for which $c_{ij} \neq -c_{ji}$. Show how to transform $\mathcal{N}$ into another network $\mathcal{N}'$ without 2-cycles such that every feasible flow in $\mathcal{N}$ corresponds to a feasible flow in $\mathcal{N}'$ of the same cost. What is the complexity of this transformation?

3.3. Prove Lemma 3.2.1 (a).

3.4. Prove Lemma 3.2.2.

3.5. Prove Lemma 3.2.3. In particular, argue why we need to take $l_{ts} = M$ rather than $l_{ts} = 0$.

3.6. Prove Lemma 3.2.4.

3.7. **(+) Fast decomposition of flows.** Prove Lemma 3.3.2.

3.8. **Decomposing an eulerian directed multigraph into arc-disjoint cycles**. Prove that the arc set of every eulerian directed multigraph can be decomposed into arc-disjoint cycles. Hint: form a circulation in an appropriate network and apply Theorem 3.3.1.

3.9. Find the residual network corresponding to the network and flow indicated in Figure 3.22.

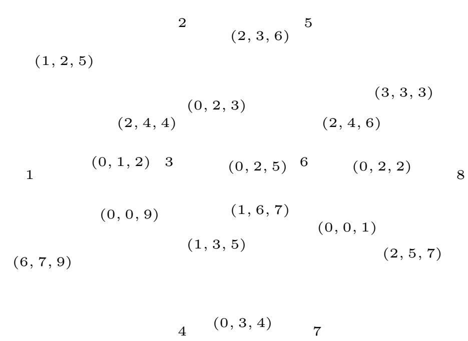


Figure 3.22 A network with a flow x. The notation for the arcs are (l, x, u).

3.10. Find the balance vector b_x for the flow x in Figure 3.22.

3.11. **Eliminating lower bounds on arcs in maximum flow problems.** Show how to reduce the maximum (s,t)-flow problem in a network $\mathcal{N}$ with some non-zero lower bounds on the arcs to the maximum (s',t')-flow problem in a network $\mathcal{N}'$ with source s' and sink t' and all lower bounds equal to zero.

3.12. Let x be a flow in $\mathcal{N} = (V, A, l \equiv 0, u, c)$ and let $f(W)$ be a cycle flow of value δ in $\mathcal{N}(x)$. Show that the flow $x^* = x \oplus f(W)$ has the same balance vector as x in $\mathcal{N}$. Show also that the cost of x^* is given by $c^T x + c^T f(W)$.

3.13. Prove that the flow $\bar{x}$ defined in the proof of Theorem 3.4.3 is a feasible flow in $\mathcal{N}(x)$.

3.14. Let x be a feasible flow in $\mathcal{N} = (V, A, l \equiv 0, u, c)$ and let y be a feasible flow in $\mathcal{N}(x)$. Show that $\mathcal{N}(x \oplus y) = \mathcal{N}(x)(y)$, where $\mathcal{N}(x)(y)$ denotes the residual network of $\mathcal{N}(x)$ with respect to y. That is, show that the two networks contain the same arcs and with the same residual capacities.

3.15. **An alternative decomposition of a flow.** Consider the proof of Theorem 3.3.1 and suppose that, instead of taking $\mu = \min\{b_x(i_0), -b_x(i_k), \delta\}$, we let $\mu = \delta$. What kind of decomposition into path and cycle flows will we get and what is the bound on their number?

3.16. **Structure of minimum (s,t)-cuts.** Decide which of the following is true or false. In each case either give a counter-example or a proof of correctness.
(a) If all arcs have different capacities, then there is a unique minimum (s,t)-cut.
(b) If we multiply the capacity of each arc by a constant k, then the structure (as subset of the vertices) of the minimum (s,t)-cuts is unchanged.
(c) If we add a constant k to the capacity of each arc, then the structure (as subset of the vertices) of the minimum (s,t)-cuts is unchanged.

3.17. **(+) The Ford-Fulkerson algorithm may never terminate if capacities are real numbers**.

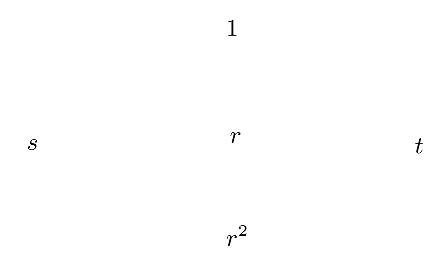


Figure 3.23 A bad network for the generic Ford-Fulkerson algorithm. All arcs except the three in the middle have capacity $r+2$. Those in the middle have capacities $1, r, r^2$, where r is the golden ratio.

Let $\mathcal{N}$ be the network in Figure 3.23. Here r is the golden ratio, i.e. $r^2 = 1-r$. Observe that $r^{n+2} = r^n - r^{n+1}$ for $n = 1, 2, \dots$.
(a) Show that the value of a maximum flow in $\mathcal{N}$ is $1 + r + r^2 = 2$.
(b) Devise an infinite sequence of augmentations along properly chosen augmenting paths in the current residual network so that the flow value will converge towards $1 + \sum_{i=2}^{\infty} r^i = 2$. This shows that, when the capacities are non-rational numbers, the Ford-Fulkerson algorithm may never terminate. Hint: first augment by one unit and then by r^i units in the ith augmentation step, $i \geq 2$, along an appropriately chosen augmenting path.

3.18. **(+)** Prove that the Ford-Fulkerson algorithm will always terminate if all capacities are rational numbers.

3.19. Let S be a totally unimodular $p \times q$ matrix and I the $p \times p$ identity matrix. Show that the matrix $[S\ I]$ is also totally unimodular.

3.20. **Exact distance labels give a height function for the preflow-push algorithm.** Let $\mathcal{N}$ be a network with source s and sink t and let x be a preflow in $\mathcal{N}$ such that there is no (s,t)-path in $\mathcal{N}(x)$. Prove that if we let $h(i)$ equal the distance from i to t in $\mathcal{N}(x)$ for $i \in V - s$ and $h(s) = n$, then we obtain a height function.

3.21. **Bad performance of the preflow-push algorithm**. Give an example which shows that the preflow-push algorithm may use many applications of **push** and **lift** without sending any extra flow into t or back to s.

3.22. **Eliminating some useless work in the preflow-push algorithm**. Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t. Suppose that we execute the generic preflow-push algorithm on $\mathcal{N}$. Let h be a height function with respect to $\mathcal{N}$ and x. We say that h has a **hole** at position $i+1$, for some $i < n$ at some point in the execution of the algorithm if at that time the following holds:

$|\{v : h(v) = j\}| > 0$ for every $j \leq i$ and
$|\{v : h(v) = i+1\}| = 0$.

Let h' be defined as follows:

$h'(v) = h(v)$ if $h(v) \in \{1, 2, \ldots, i\} \cup \{n, n+1, \ldots, 2n-1\}$
$h'(v) = n+1$ if $i < h(v) < n$.

(a) Prove that h' is a height function, that is, (3.14) is satisfied.
(b) Describe how to implement this modification of the height function efficiently so that it may be used as a subroutine in the preflow-push algorithm.
(c) Explain why changing the height function as above when a hole is detected may help speed up the preflow-push algorithm.

3.23. **Using the height function to detect a minimum cut after termination of the preflow-push algorithm.** Suppose x is a maximum (s,t)-flow that been found by executing the preflow-push algorithm on a network $\mathcal{N} = (V, A, l \equiv 0, u)$. Describe a method to detect a minimum (s,t)-cut in $O(n)$ steps using the values of the height function upon termination of the algorithm.

3.24. **(+) Re-optimizing a maximum (s,t)-flow**. Suppose x is a maximum flow in a network $\mathcal{N} = (V, A, l \equiv 0, u)$. Show how to re-optimize x (that is, to change it to a feasible flow of maximum value) in each of the following cases:
(a) Increase the capacity of one arc by k units. Show that the new optimal solution can be found in time $O(km)$.
(b) Decrease the capacity of one arc by k units. Show that new optimal solution can be found in time $O(km)$. Hint: use Theorem 3.3.1.

3.25. **(+) Pulling and pushing flow, the MKM-algorithm.** The purpose of this exercise is to introduce another, very efficient, method for finding a blocking (s,t)-flow in a layered network due to Malhotra, Kumar and Maheshwari [544]. Let $\mathcal{L} = (V = V_0 \cup V_1 \cup \ldots \cup V_k, A, l \equiv 0, u)$ be a layered network with $V_0 = \{s\}$ and $V_k = \{t\}$. Let y be a feasible (s,t)-flow which is not blocking in $\mathcal{L}$. For each vertex $i \in V - \{s,t\}$ let $\alpha_i, \beta_i, \rho_i$ be defined as follows:

$$\alpha_i = \sum_{ji \in A} u_{ji} - y_{ji} \tag{3.36}$$

$$\beta_i = \sum_{ij \in A} u_{ij} - y_{ij} \tag{3.37}$$

$$\rho_i = \min\{\alpha_i, \beta_i\}. \tag{3.38}$$

Let

$$\rho_s = \sum_{sj \in A} u_{sj} - y_{sj}, \rho_t = \sum_{jt \in A} u_{jt} - y_{jt}. \tag{3.39}$$

Finally let $\rho = \min_{i \in V}\{\rho_i\}$.
Suppose that $\rho > 0$ and let $i \in V$ be chosen such that $\rho = \rho_i$.

(a) Prove that it is possible to send an additional amount of ρ units from i to t (called **pushing from i to t**) and ρ units of flow from s to i in $\mathcal{L}$ (called **pulling from s to i**). Hint: use that the network is layered.

The observation above leads to the following algorithm $\mathcal{A}$ for finding a blocking flow in a layered network. Below the ρ-values always refer to the current flow.

The MKM-algorithm

1. Start with the zero flow $y \equiv 0$ and calculate ρ_i for all $i \in V$. If some $i \in V$ has $\rho_i = 0$ then go to Step 6;
2. Choose i such that $\rho_i = \rho$;
3. Push ρ units of flow from i to t and pull ρ units from s to i;
4. Delete all arcs which are saturated with respect to the new flow. If this results in some vertex of in- or out-degree zero, then also delete that vertex and all incident arcs. Continue this until no more arcs can be deleted;
5. Calculate ρ_i for all vertices in the current layered network. If $\rho_i > 0$ for all vertices then go to Step 2. Otherwise go to Step 6.
6. If $\rho_s = 0$ or $\rho_t = 0$, then halt;
7. If there is a vertex i with $\rho_i = 0$, then delete all such vertices and their incident arcs;
8. Go to Step 5.

(b) Prove that the algorithm above correctly determines a blocking flow in the input layered network $\mathcal{L}$.
The complexity of $\mathcal{A}$ depends on how we perform the different steps, especially Step 3. Suppose we apply the following rule for performing Step 3. We always push/pull ρ units one layer at a time. If j is the current vertex from (to) which we wish to send flow to (from) the next (previous) layer, then we always fill an arc with tail (head) j completely if there is still enough flow left and then continue to fill the next arc as much as possible.

(c) Argue that, using the rule above, we can implement the algorithm to run in $O(n^2)$ time. Hint: at least one vertex will be deleted between two consecutive applications of Step 3. Furthermore, one can keep the ρ-values effectively updated (explain how).

(d) Illustrate the algorithm on the layered network in Figure 3.10.

3.26. **Finding maximum (s,t)-flows by scaling.** Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t and let U denote the maximum capacity of an arc in $\mathcal{N}$.

(a) **(−)** Prove that the capacity of a minimum (s,t)-cut is at most $U|A|$.

(b) Let C be a constant and let x be a feasible (s,t)-flow in $\mathcal{N}$. Show that in time $O(|A|)$ one can find an augmenting path of capacity at least C, or detect that no such path exists in $\mathcal{N}(x)$. Hint: consider the subnetwork of $\mathcal{N}(x)$ containing only arcs whose capacity is at least C.

(c) Consider the following algorithm:

Max-flow by scaling

1. $U := \max\{u_{ij} : ij \in A\}$;
2. $x_{ij} := 0$ for every $ij \in A$;
3. $C := 2^{\lfloor \log_2 U \rfloor}$;
4. while $C \geq 1$ do
5.   while $\mathcal{N}(x)$ contains an augmenting path of capacity at least C

do augment x along P;
6. $C := C/2$
7. return x

Prove that the algorithm correctly determines a maximum flow in the input network $\mathcal{N}$.

(d) Argue that every time Step 4 is performed the residual capacity of every minimum (s,t)-cut is at most $2C|A|$.
(e) Argue that the number of augmentations performed in Step 5 is at most $O(|A|)$ before Step 6 is executed again.
(f) Conclude that **Max-flow by scaling** can be implemented so that its complexity becomes $O(|A|^2 \log U)$. Compare this complexity to that of other flow algorithms in this chapter.

3.27. Show how to find a maximum (s,t)-flow in the network of Figure 3.24 using
(a) The Ford-Fulkerson method.
(b) Dinic's algorithm.
(c) The preflow-push algorithm.
(d) The MKM-algorithm described in Exercise 3.25.
(e) The scaling algorithm described in Exercise 3.26.

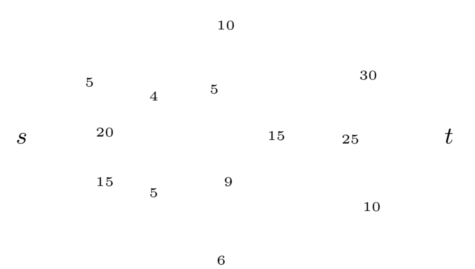


Figure 3.24 A network with lower bounds and cost equal to zero on all arcs and capacities as indicated on the arcs.

3.28. **(+) Rounding a real-valued flow**. Let $\mathcal{N} = (V, A, l, u)$ be a network with source s and sink t and all data on the arcs non-negative integers (note that some of the lower bounds may be non-zero). Suppose x is a real-valued feasible flow in $\mathcal{N}$ such that x_{ij} is a non-integer for at least one arc.
(a) Prove that there exists a feasible integer flow x' in $\mathcal{N}$ with the property that $|x_{ij} - x'_{ij}| < 1$ for every arc $ij \in A$.
(b) Suppose now that $|x|$ is an integer. Prove that there exists an integer feasible flow x'' in $\mathcal{N}$ such that $|x''| = |x|$.
(c) Describe algorithms to find the flows x', x'' above. What is the best complexity you can achieve?

3.29. **Finding a feasible circulation.** Turn the proof of Theorem 3.8.2 into a polynomial algorithm which either finds a feasible circulation, or a proof that none exists. What is the complexity of the algorithm?

3.30. **Residual networks of networks with non-zero lower bounds**. Show how to modify the definition of $x \oplus \tilde{x}$ in order to obtain an analogue of

Theorem 3.4.2 for the case of networks where some lower bounds are non-zero.

3.31. Show that a feasible circulation (if one exists) can always be found by just one max flow calculation in a suitable network. Hint: transform the network into an (s,t)-flow network with all lower bounds equal to zero.

3.32. **(+) Flows with balance vectors within prescribed intervals.** Let $\mathcal{N} = (V, A, l, u)$ be a network where $V = \{1, 2, \ldots, n\}$ and let $a_i \leq b_i$, $i = 1, 2 \ldots, n$ be integers. Prove that there exists a flow x in $\mathcal{N}$ which satisfies

$$l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall ij \in A \tag{3.40}$$

$$a_i \leq b_x(i) \leq b_i \quad \forall i \in V \tag{3.41}$$

if and only if the following three conditions are satisfied:

$$\sum_{i \in V} a_i \leq 0 \tag{3.42}$$

$$\sum_{i \in V} b_i \geq 0 \tag{3.43}$$

$$u(X, \overline{X}) \geq l(\overline{X}, X) + \max\{a(X), -b(\overline{X})\} \quad \forall X \subset V, \tag{3.44}$$

where $a(X) = \sum_{i \in X} a_i$.

Hint: construct a network which has a feasible circulation if and only if (3.40) and (3.41) holds. Then apply Theorem 3.8.2.

3.33. **Submodularity of the capacity function for cuts**. Let $\mathcal{N} = (V, A, l, u)$ be a network with source s and sink t. Prove that, if $(S, \overline{S})$ and $(T, \overline{T})$ are (s,t)-cuts, then

$$u(S, \overline{S}) + u(T, \overline{T}) \geq u(S \cap T, \overline{S \cap T}) + u(S \cup T, \overline{S \cup T}).$$

Hint: consider the contribution of each arc in the network to the four cuts.

3.34. Show that, if $(S, \overline{S})$ and $(T, \overline{T})$ are minimum (s,t)-cuts, then so are $(S \cap T, \overline{S \cap T})$ and $(S \cup T, \overline{S \cup T})$. Hint: use Exercise 3.33.

3.35. **(+) Finding special minimum cuts**. Suppose that x is a maximum (s,t)-flow in a network $\mathcal{N} = (V, A, l, u)$. Let

$$U = \{i : \text{ there exists an } (s,i)\text{-path in } \mathcal{N}(x)\},$$

$$W = \{j : \text{ there exists an } (j,t)\text{-path in } \mathcal{N}(x)\}.$$

Prove that $(U, \overline{U})$ and $(\overline{W}, W)$ are minimum (s,t)-cuts. Then prove that for every minimum (s,t)-cut (S, T) we have $U \subseteq S$ and $W \subseteq T$.

3.36. **(+)** Let x be an (s,t)-flow in a network $\mathcal{N} = (V, A, l, u)$. Explain how to find an augmenting path of maximum capacity in polynomial time. Hint: use a variation of Dijkstra's algorithm.

3.37. **(+) Augmenting along maximum capacity augmenting paths.** Show that, if we always augment along an augmenting path with the maximum residual capacity, then the Ford-Fulkerson algorithm becomes a polynomial algorithm (Edmonds and Karp [216]). Hint: show that the number of augmentations is $O(m \log U)$, where U is the maximum capacity of an arc.

3.38. **Converting a maximum preflow to a maximum (s,t)-flow.** Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t. A preflow x in $\mathcal{N}$ is **maximum** if $|b_x(t)|$ equals the value of a maximum (s,t)-flow in $\mathcal{N}$.
(a) Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t and let y be a maximum preflow in $\mathcal{N}$. Prove that there exists a maximum (s,t)-flow x in $\mathcal{N}$ with the property that $x_{ij} \leq y_{ij}$ for every arc $ij \in A$. Hint: use flow decomposition.
(b) How fast can you convert a maximum preflow to a maximum (s,t)-flow?

3.39. **(−)** Prove Lemma 3.7.1.

3.40. **(−)** Prove Lemma 3.7.6.

3.41. Show that the complexity of Dinic's algorithm for unit capacity networks remains $O(n^{\frac{2}{3}}m)$ even if we allow the network to have 2-cycles. Hint: prove a modified version of Lemma 3.7.3 and apply that as we applied Lemma 3.7.3 in the proof of Theorem 3.7.4.

3.42. **Elimination of 2-cycles from simple networks.** Suppose that $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$ is a simple unit capacity network with source s, sink t and that uvu is a 2-cycle in $\mathcal{N}$. Show that we may always delete one of the arcs uv or vu without affecting the value of a maximum (s,t)-flow in $\mathcal{N}$.

3.43. Prove Theorem 3.7.7. Hint: see the proof of Theorem 3.7.4.

3.44. Show how to derive Theorem 3.8.4 from Lemma 3.2.2 and Theorem 3.8.2.

3.45. **Scheduling jobs on identical machines.** Let J be a set of jobs which are to be processed on a set of identical machines (such as processors, airplanes, trucks etc). Each job is processed by one machine. There is a fixed schedule for the jobs, specifying that job $j \in J$ must start at time s_j and finish at time f_j. Furthermore, there is a transition time t_{ij} required to set up a machine which has just performed job i to perform job j (e.g., jobs could be different loads for trucks and t_{ij} could be time to drive a truck from the position of load i to that of load j). The goal is to find a feasible schedule for the jobs which requires as few machines as possible. Show how to formulate this problem as a minimum value (s,t)-flow problem.

3.46. **(+) Scheduling supervision of projects**. This exercise deals with a practical problem concerning the assignment of students to various projects in a course. All projects which are chosen by at least one student are to be supervised by one or more qualified teachers. Each student is supervised by one teacher only. There are n students, m different projects and t possible supervisors for the projects.
Let b_i, $i = 1, 2, \ldots, m$, denote the maximum number of students who may choose the same project (they work alone and hence need individual supervision). For each project i, $i = 1, 2, \ldots, m$, there is a subset $A_i \subseteq \{1, \ldots, t\}$ of the teachers who are capable of supervising the ith project. Finally each teacher j, $j = 1, 2, \ldots, t$ has an upper limit of k_j on the number of students (s)he can supervise.
Every student must be assigned exactly one project. We also assume that each student has ranked the projects from 1 to m according to the order of preference. Namely, the project the student would like best is ranked one. Denote the rank of project j by student i by r_{ij}.

The goal is to find an assignment $p(1), p(2), \dots, p(n)$ of students to projects (that is student i is assigned project $p(i)$) which respects the demands above and at the same time minimizes the sum $\sum_{i=1}^{n} r_{ip(i)}$.

(a) Show how to formulate the problem as a minimum cost flow problem.
(b) If we only wish to find a feasible assignment (i.e. one that does not violate the demands above), then which is the fastest algorithm you can device?
(c) Which minimum cost flow algorithm among those in Section 3.10 will give the fastest algorithm for the problem when formulated as in question (a)?
(d) Let $p(1), p(2), \dots, p(n)$ be an optimal assignment of students to projects. Suppose that *before* the actual supervision of the projects starts, some supervisor $j \in \{1, 2, \dots, t\}$ lowers his/her capacity for supervision from k_j to $k'_j < k_j$.
Describe a fast algorithm which either proves that no feasible assignment exists or changes the assignment $p(1), p(2), \dots, p(n)$ to a new optimal assignment $p'(1), p'(2), \dots, p'(n)$ with respect to the new restrictions.
(e) Suppose now that the change in capacity only happens *after* the students have started working on the projects. The goal now is to find a new optimal and feasible solution or show that no feasible solution exists, while at the same time rescheduling as few students as possible to new projects (we assume that rescheduled students must start all over again). Explain briefly how to solve this variant of the problem. Hint: devise some measure of cost for rescheduling a student in a minimum cost flow model.

3.47. **(−)** Let $\mathcal{N} = (V, A, l \equiv 0, u)$ be a network with source s and sink t and let $\mathcal{N}' = (V, A', l' \equiv 0, u', c')$ be obtained from $\mathcal{N}$ by adding a new arc ts with $u_{ts} = \infty$ and $c_{ts} = -1$ taking $u'_{ij} = u_{ij}$ for all $ij \in A$ and $c'_{ij} = 0$ for all $ij \in A$. Prove that there is a 1-1 correspondence between the minimum cost circulations in $\mathcal{N}'$ and the maximum (s,t)-flows in $\mathcal{N}$.

3.48. Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network with some arcs of infinite capacity and some arcs of negative cost.
(i) Show that there exists a finite optimal solution to the minimum cost flow problem (finding a feasible flow in $\mathcal{N}$ of minimum cost) if and only if $\mathcal{N}$ has no cycle C of negative cost such that all arcs of C have infinite capacity. Hint: study the difference between an arbitrary feasible solution and some fixed solution of finite cost.
(ii) Let K be the sum of all finite capacities and those b-values that are positive. Show that, if there exists a finite optimal solution to the minimum cost flow problem for $\mathcal{N}$, then there exists one for which no arc has flow value more than K. Hint: use flow decomposition.

3.49. **Eliminating negative cost arcs from minimum cost flow problems.** Suppose $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ contains an arc uv of negative cost, but no cycle of infinite capacity and negative cost (see Exercise 3.48). Derive a result similar to Lemma 3.2.1 which can be used to transform $\mathcal{N}$ into a new network $\mathcal{N}^+$ in which all costs are non-negative and such that given any feasible flow x^+ in $\mathcal{N}^+$ we can obtain a feasible flow x in $\mathcal{N}$ and find the cost of x efficiently, given the cost of x^+. Hint: reverse arcs of negative costs, negate the costs of such arcs and update balance vectors.

3.50. Prove Theorem 3.10.6.

3.51. Try to generalize the statement of Proposition 3.10.7 to the case when the paths P, P' do not necessarily have the same end vertices. Hint: consider the network $\mathcal{N}_{st}$ obtained as in Lemma 3.2.2.

3.52. Show by an example that the cycle canceling algorithm may use $\Omega(mUC)$ augmentations before arriving at an optimal flow.

3.53. **(−)** Show how to reduce the problem of finding a matching in a bipartite graph $B = (X, Y, E)$ which maximizes the number of edges incident vertices in X to the problem of finding a maximum matching in a bipartite graph.

3.54. **(+)** Prove that, if D is a k-regular semicomplete digraph on n vertices, then D contains a spanning tournament T which is regular or almost regular ($|\delta^+(T) - \delta^-(T)| \leq 1$) depending on whether n is odd or even. Observe that every regular tournament has an odd number of vertices (Bang-Jensen [47]).

3.55. **(+) Generalized matchings in undirected graphs.** Let $G = (V, E)$ be an undirected graph. Recall that for any subset $S \subset V$ we denote by $N_G(S)$ the set of vertices in $V - S$ which have at least one edge to S. Prove that every graph G either has a vertex disjoint collection of edges $e_1, \ldots, e_k$ and odd cycles $C_1, \ldots, C_r$ covering V, or a set $S \subset V$ with $|N_G(S)| < |S|$ and S is independent. Derive an algorithm from your proof which either finds the desired generalized matching, or an independent subset S such that $|N(S)| < |S|$. Hint: use Theorem 3.8.2 on an appropriate network.

3.56. Prove the following theorem due to König [499]. Every regular bipartite graph has a perfect matching.

3.57. **(−) 1-regular digraphs**. Prove that, if D is a 1-regular digraph, then D is precisely a collection of vertex disjoint cycles $C_1, \ldots, C_k$ for some $k \geq 1$.

3.58. **Cycle factors of directed pseudographs**. Prove that Proposition 3.11.6 also holds for directed pseudographs provided we consider a loop as a cycle.

3.59. **(+) Calculating the path-cycle covering number of a digraph.** Show how to find in time $O(\sqrt{n}m)$ the minimum integer k such that a given digraph D has a path-cycle factor with k paths. Hint: use minimum value flows in an appropriately constructed simple network.

3.60. **(+) Path-cycle covering numbers of extensions of digraphs.** Let R be a digraph on r vertices, and let $l_1 \leq u_1, l_2 \leq u_2, ..., l_r \leq u_r$ be $2r$ non-negative integers. Let I_p denote an independent set on p vertices. Show how to find $\min\{\mathrm{pcc}(R[I_{p_1}, ..., I_{p_r}]) : \ l_i \leq p_i \leq u_i, \ i = 1, ..., r\}$ in time $O(n^3)$. Hint: generalize the network you used in Exercise 3.59 (Bang-Jensen and Gutin [65, 365]).

3.61. Let $k \in \mathcal{Z}_+$. Show that a directed graph $D = (V, A)$ has a k-path-cycle factor if and only if $|\bigcup_{v \in X} N^+(v)| \geq |X| - k$ and $|\bigcup_{v \in X} N^-(v)| \geq |X| - k$.

3.62. Show how to decide in time $O(\sqrt{n}m)$ whether or not a given input digraph D with special vertices x, y contains a 1-path-cycle factor such that the path is a path between x and y.

3.63. Complete the proof of Theorem 3.11.11.

3.64. **Heaviest cycle subdigraphs in digraphs.** Describe an $O(n^3)$ algorithm to find, in a digraph with non-negative weights on the arcs, a cycle subdigraph of maximum weight. Hint: use the same approach as in the proof of Theorem 3.11.11.

3.65. (+) Prove Theorem 3.11.13 directly from Theorem 3.8.2. Show that your proof implies the existence of an algorithm, which given a k-strong digraph D and a subset $X \subseteq V(D)$, either finds a collection of disjoint cycles covering all the vertices of X, or an independent set $X' \subseteq X$ of size more than k.

3.66. Find a minimum cost Chinese postman walk in the digraph of Figure 3.25.

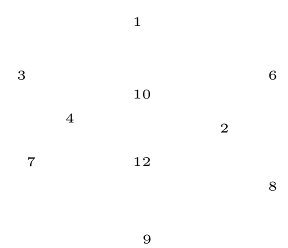


Figure 3.25 A digraph with weights on the arcs.

3.67. Show how to formulate the following problem as a flow problem. Given two sequences of non-negative integers $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ decide whether or not there exists a directed multigraph $D = (\{v_1, v_2, \ldots, v_n\}, A)$ such that $d^+_D(v_i) = a_i$ and $d^-_D(v_i) = b_i$ for each $i = 1, 2, \ldots, n$. Hint: use Theorem 3.11.3 or the proof idea of this theorem.

3.68. Prove Proposition 3.11.8.

3.69. Prove Proposition 3.11.10. Hint: use the same network as in Exercise 3.59.

3.70. **Every regular directed multigraph has a cycle factor.** Prove this claim.

3.71. Show how to reduce the case when $\sum_{i=1}^{m} a_i \neq \sum_{j=1}^{n} b_j$ to the case when the equality holds for the transportation problem. Hint: introduce new plants or retailers.

3.72. Prove Lemma 3.12.1.

3.73. Prove that Lemma 3.12.1 also holds when we consider the dual variables α^*, β^* which are updated as in (3.35).

3.74. (+) Show that by using appropriate data structures and by keeping labels (used in previous searches for augmenting paths) until a new augmenting path has been found (implying that the value of the current flow can be increased), one can implement the primal-dual algorithm for the transportation problem so that it runs in time $O(M(n+m)^2)$.

3.75. Solve the following assignment problem using the primal-dual method.

12	9	10	8	11
8	6	6	5	9
13	10	10	11	11
6	2	4	3	5
11	7	10	9	11

3.76. Show that the buildup algorithm of Section 3.10 can be applied to solve the assignment problem in time $O(n^3)$.

3.77. Show that if we update the dual variables according to (3.35) we still obtain a feasible solution to (3.27) whose objective function value is strictly higher than that of α, β.

3.78. The following table shows an instance of the transportation problem after some iterations of the primal-dual method outlined in Section 3.12. Complete the algorithm on this example.

							α	a
	5	$^{3}\,3$	7	$^{1}\,3$	8	5	5	4
	$^{2}\,5$	6	12	5	$^{1}\,7$	11	5	3
	$^{1}\,2$	8	$^{2}\,3$	4	8	2	2	3
	9	6	10	$^{1}\,5$	10	9	7	7
β	0	-2	1	-2	2	0		
b	3	3	6	2	1	2		

3.79. **Tree solution to a flow problem**. Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network with n vertices for which there exists a feasible flow and let $D = (V, A)$ be the underlying digraph of $\mathcal{N}$. Prove that there exists a feasible flow x in $\mathcal{N}$ such that the number of arcs on which $0 < x_{ij} < u_{ij}$ is at most $n-1$. We call such a feasible flow a **tree solution**. Hint: show that, if C is a cycle in $UG(D)$ where $0 < x_{ij} < u_{ij}$ for every arc on the cycle, then we can change the current flow such that the resulting flow x' is either 0 or u_{ij} for at least one arc ij of C and no new arc pq with $0 < x'_{pq} < u_{pq}$ is created.

3.80. Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network with n vertices for which there exists a feasible flow. Prove that there exists an optimal feasible flow which is a tree solution.

3.81. **Vertex potentials and flows.** Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network and x a feasible flow in $\mathcal{N}$. Prove that x is an optimal flow if and only if there exists a function $\pi : V \rightarrow \mathcal{R}$ such that $c^{\pi}_{ij} \geq 0$ for every arc ij in $\mathcal{N}(x)$. Here $c^{\pi}_{ij} = c_{ij} - \pi(i) + \pi(j)$ is the **reduced cost function** and the costs in $\mathcal{N}(x)$ are with respect to c^{π} instead of c. Hint: see Exercises 2.16-2.18.

3.82. **Complementary slackness conditions for optimality of a flow**. Let $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ be a network and x a feasible flow in $\mathcal{N}$. Prove that x is an optimal flow if and only if there exists a function $\pi : V \rightarrow \mathcal{R}$ such that the following holds:

$$c^{\pi}_{ij} > 0 \quad \Rightarrow \quad x_{ij} = 0 \tag{3.45}$$
$$c^{\pi}_{ij} < 0 \quad \Rightarrow \quad x_{ij} = u_{ij} \tag{3.46}$$
$$0 < x_{ij} < u_{ij} \quad \Rightarrow \quad c^{\pi}_{ij} = 0. \tag{3.47}$$

Here $c^{\pi}_{ij} = c_{ij} - \pi(i) + \pi(j)$ as above. Hint: use Exercise 3.81.

3.83. **(+) A primal-dual algorithm for minimum cost flows.** Let $\mathcal{N} = (V, A, l \equiv 0, u, c)$ be a network with source s and sink t for which the value of a maximum (s,t)-flow is $K > 0$. Let x be an optimal (feasible) (s,t)-flow of value $k < K$ and let $\pi : V \rightarrow \mathcal{R}$ be chosen such that $c^{\pi}_{ij} \geq 0$ for every arc ij in $\mathcal{N}(x)$ (see Exercise 3.81). Define A_0 as those arcs ij of $\mathcal{N}(x)$ for which we have $c^{\pi}_{ij} = 0$ and let $\mathcal{N}_0$ be the subnetwork of $\mathcal{N}(x)$ induced by the arcs of A_0.

(a) Show that if y is a feasible (s,t)-flow in $\mathcal{N}_0$ of value p then $x' = x \oplus y$ is an optimal (s,t)-flow of value $k + p$ in $\mathcal{N}$. Hint: verify that $c^{\pi}_{ij} \geq 0$ holds for every arc ij in $\mathcal{N}(x')$.

(b) Suppose y is a maximum (s,t)-flow in $\mathcal{N}_0$, but $x' = x \oplus y$ has value less than K. Let S denote the set of vertices which are reachable from s in $\mathcal{N}_0(y)$. Let $\epsilon, \epsilon_1, \epsilon_2$ be defined as follows. Here we let $\epsilon_i = \infty$ if there are no arcs in the corresponding set, $i = 1, 2$:

$$\epsilon_1 = \min\{c^{\pi}_{ij} | i \in S, j \in \overline{S}, c^{\pi}_{ij} > 0 \text{ and } x_{ij} < u_{ij}, \}$$
$$\epsilon_2 = \min\{-c^{\pi}_{ij} | i \in \overline{S}, j \in S, c^{\pi}_{ij} < 0 \text{ and } x_{ij} > 0\}.$$

Let $\epsilon = \min\{\epsilon_1, \epsilon_2\}$. Prove that $\epsilon < \infty$.

(c) Now define π' as follows: $\pi'(v) := \pi(v) + \epsilon$ if $v \in S$ and $\pi'(v) := \pi(v)$ if $v \in \overline{S}$. Let $\mathcal{N}'_0$ contain those arcs of $\mathcal{N}(x')$ for which we have $c^{\pi'}_{ij} = 0$ and let S' denote the set of vertices which are reachable from s in $\mathcal{N}'_0$. Show that S is a proper subset of S' and that $c^{\pi'}_{ij} \geq 0$ holds for all arcs in $\mathcal{N}(x')$. Hint: use Exercise 3.14.

(d) If $t \notin S'$, then we can change π' as above (based on the set S' rather than S). Conclude that after at most $n - 1$ such updates of the vector π', the current network $\mathcal{N}'_0$ contains an (s,t)-path.

(e) Use the observations above to design an algorithm that finds a minimum cost (s,t)-flow of value K in $\mathcal{N}$ by solving a sequence of maximum flow problems. What is the complexity of this algorithm?

4. Classes of Digraphs

In this chapter we introduce several classes of digraphs. We study these, along with the classes of digraphs defined already in Chapter 1, with respect to their characterization, recognition and decomposition. We also consider some basic properties of these classes. Further properties of the classes are studied in the following chapters of this book.

We start this chapter by introducing Depth-First Search (DFS), an important technique in algorithms on graphs. This technique is used in this chapter and some other chapters to design fast algorithms. In particular, DFS is used in Section 4.2, where we describe a fast algorithm to find an acyclic ordering in an acyclic digraph. In Section 4.3, we introduce and study the transitive closure and a transitive reduction of a digraph. We use these notions in Section 4.7. A linear time algorithm for finding strong components of a digraph based on DFS is given in Section 4.4.

Several characterizations and a recognition algorithm for line digraphs are given in Section 4.5. We investigate basic properties of de Bruijn and Kautz digraphs and their generalizations in Section 4.6. These digraphs are extreme or almost extreme with respect to their diameter and vertex-strong connectivity. Series-parallel digraphs are introduced and studied in Section 4.7. These digraphs are of interest due to various applications such as scheduling. In the study of series-parallel digraphs we use notions and results of Sections 4.3 and 4.5.

An interesting generalization of transitive digraphs, the class of quasi-transitive digraphs, is considered in Section 4.8. The path-merging property of digraphs which is quite important for investigation of some classes of digraphs including tournaments is introduced and studied in Section 4.9. Two classes of path-mergeable digraphs, locally in-semicomplete and locally out-semicomplete digraphs, both generalizing the class of tournaments, are defined and investigated with respect to their basic properties in Section 4.10. The intersection of these two classes forms the class of locally semicomplete digraphs, which are studied in Section 4.11. There we give a very useful classification of locally semicomplete digraphs, which is applied in several proofs in other chapters. A characterization of a special subclass of locally semicomplete digraphs, called round digraphs, is also proved.

Three classes of totally decomposable digraphs forming important generalizations of quasi-transitive digraphs as well as some other classes of digraphs are studied in the above-mentioned sections. We investigate recognition of these three classes in Section 4.12. Some properties of intersection digraphs are given in Section 4.13. Planar digraphs are discussed in Section 4.14. The last section is devoted to an application of digraphs to solving systems of linear equations.

4.1 Depth-First Search

In this section we will introduce a simple, yet very important, technique in algorithmic graph theory called **depth-first search**. While depth-first search (DFS) has certain similarities with BFS (see Section 2.3.1), DFS and BFS are quite different procedures, each with its own features.

Let $D = (V, A)$ be a digraph. In DFS, we start from an arbitrary vertex of D. At every stage of DFS, we **visit** some vertex x of D. If x has an **unvisited** out-neighbour y, we visit the vertex y[1] . We call the arc xy a **tree arc**. If x has no unvisited out-neighbour, we call x **explored** and return to the predecessor $\mathrm{pred}(x)$ of x (the vertex from which we have moved to x). If x does not have a predecessor, we find an unvisited vertex to 'restart' the above procedure. If such a vertex does not exist, we stop.

In our formal description of DFS, each vertex x of D gets two time-stamps: $\mathrm{tvisit}(x)$ once x is visited and $\mathrm{texpl}(x)$ once x is declared explored.

DFS
Input: A digraph $D = (V, A)$.
Output: $\mathrm{pred}(v)$, $\mathrm{tvisit}(v)$ and $\mathrm{texpl}(v)$ for every $v \in V$.

1. For each $v \in V$ set $\mathrm{pred}(v) :=$ **nil**, $\mathrm{tvisit}(v) := 0$ and $\mathrm{texpl}(v) := 0$.
2. Set $time := 0$.
3. For each vertex $v \in V$ do: if $\mathrm{tvisit}(v) = 0$ then perform DFS-PROC(v).

DFS-PROC(v):

1. Set $time := time + 1$, $\mathrm{tvisit}(v) := time$.
2. For each $u \in N^+(v)$ do: if $\mathrm{tvisit}(u) = 0$ then $\mathrm{pred}(u) := v$ and perform DFS-PROC(u).
3. Set $time := time + 1$, $\mathrm{texpl}(v) := time$.

Clearly, the main body of the algorithm takes $O(n)$ time. The total time for executing the different calls of the procedure DFS-PROC is $O(m)$ (as $\sum_{x \in V} d^+(x) = m$ by Proposition 1.2.1). As a result, the time complexity of DFS is $O(n + m)$.

[1] If x has more than one unvisited out-neighbour, we choose y as an arbitrary unvisited out-neighbour.

Unlike BFS, in the end of DFS, the tree arcs may form a non-connected spanning subdigraph F of D (recall that we perform BFS from a prescribed vertex). The arc set of F is $\{(\text{pred}(v), v) : v \in V, \text{pred}(v) \neq \textbf{nil}\}$. Since each component of $UG(F)$ is a tree, F is a forest. We call F a **DFS forest**; a tree in F is a **DFS tree**; the **root** of a DFS tree is some vertex v used in Step 3 of the main body of DFS to initiate DFS-PROC. Clearly, the root r of a DFS tree T is the only vertex of T whose in-degree is zero. According to the above description of DFS every vertex in T can be reached from r by a path (hence T is an out-branching rooted at r in the subdigraph induced by $V(T)$). We say that a vertex x in T is a **descendant** of another vertex y in T (denoted by $x \succ y$) if y lies on the (r, x)-path in T. Note that in general there may be many different DFS forests for a given digraph D.

It is convenient to classify the non-tree arcs of a digraph $D = (V, A)$ with respect to a given DFS forest of D. If we visit a vertex x and consider an already visited out-neighbour y of x, then the following possibilities may occur.

1. The vertex y is explored, i.e., $\text{texpl}(y) \neq 0$. This means that x and y belong to different DFS trees. In this case, the arc xy is a **cross arc**.
2. The vertex y is not explored. If $x \succ y$ then xy is a **backward arc**. If $y \succ x$ then xy is a **forward arc**. If none of the above two variants occurs, xy is (again) a **cross arc**.

We illustrate the DFS algorithm and the above classification of arcs in Figure 4.1. The tree arcs are in bold. The non-tree arcs are labeled B,C or F depending on whether they are backward, cross, or forward arcs. Every vertex u is time-stamped by $\text{tvisit}(u)/\text{texpl}(u)$ if one or both of them have been changed from the initial value of zero.

Observe that, for every vertex $v \in V$, we have $\text{tvisit}(v) < \text{texpl}(v)$. There is no pair u, v of vertices such that $\text{tvisit}(u) = \text{tvisit}(v)$ or $\text{tvisit}(u) = \text{texpl}(v)$ or $\text{texpl}(u) = \text{texpl}(v)$ due to the fact that before assigning any time to $\text{tvisit}(\dots)$ or $\text{texpl}(\dots)$ the value of *time* is increased. We consider some additional simple properties of DFS. We denote the interval from time t to time $t' > t$ by $[t, t']$ and write $I \subseteq I'$ if the interval I is contained in the interval I'.

Proposition 4.1.1 *Let* $D = (V, A)$ *and let the numbers* $\text{tvisit}(v), \text{texpl}(v)$, $v \in V$*, be calculated using DFS. For every pair of vertices* u *and* v*, one of the assertions below holds:*

(1) The intervals $[\text{tvisit}(u), \text{texpl}(u)]$ *and* $[\text{tvisit}(v), \text{texpl}(v)]$ *are disjoint;*
(2) $[\text{tvisit}(u), \text{texpl}(u)] \subseteq [\text{tvisit}(v), \text{texpl}(v)]$;
(3) $[\text{tvisit}(v), \text{texpl}(v)] \subseteq [\text{tvisit}(u), \text{texpl}(u)]$.

Proof: Without loss of generality, we may assume that $\text{tvisit}(u) < \text{tvisit}(v)$. If $\text{texpl}(u) < \text{tvisit}(v)$, then the first assertion is valid. So, assume that

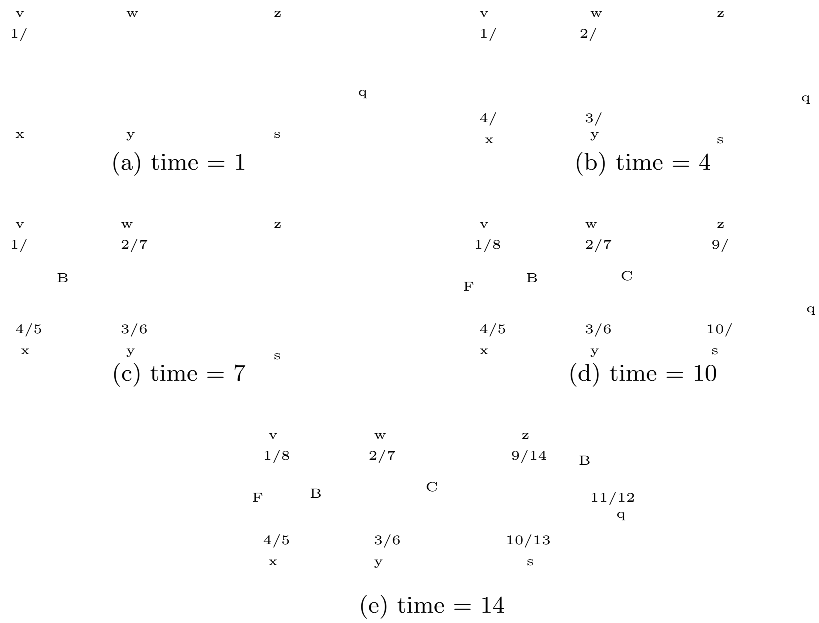


Figure 4.1 Some steps of DFS on a digraph starting from the vertex v.

texpl(u) $>$ tvisit(v). This means that v was visited when u has been already visited but u was not explored yet. Thus, there is a directed path from u to v in the DFS forest, implying that $v \succ u$. Since u cannot become explored when v is still unexplored, texpl$(v) <$ texpl(u). This implies the third assertion. □

This proposition implies immediately the following.

Corollary 4.1.2 *For a pair x, y of distinct vertices of D, we have $y \succ x$ if and only if* tvisit$(x) <$ tvisit$(y) <$ texpl$(y) <$ texpl(x). □

Proposition 4.1.3 *Let F be a DFS forest of a digraph $D = (V, A)$ and let x, y be vertices in the same DFS tree T of F. Then $y \succ x$ if and only if, at the time* tvisit(x)*, the vertex y can be reached from x along a path all of whose internal vertices are unvisited.*

Proof: Assume that $y \succ x$. Let z be an internal vertex of the (x, y)-path in T. Thus, $z \succ x$. By Corollary 4.1.2, tvisit$(x) <$ tvisit(z). Hence, z is unvisited at time tvisit(x).

Suppose that y is reachable from x along a path P of unvisited vertices at time tvisit(x), but $y \not\succ x$. We may assume that $z = y^-_P$ (the predecessor of y on P) is a descendant of x in T, that is, $z \succ x$ holds. By Corollary 4.1.2, texpl$(z) <$ texpl(x). Since y is an out-neighbour of z, y is visited before z is explored. Hence, $\text{tvisit}(y) < \text{texpl}(z)$. Clearly, $\text{tvisit}(x) < \text{tvisit}(y)$. Therefore, $\text{tvisit}(x) < \text{tvisit}(y) < \text{texpl}(x)$. By Proposition 4.1.1, it means that the interval $[\text{tvisit}(y), \text{texpl}(y)]$ is contained in the interval $[\text{tvisit}(x), \text{texpl}(x)]$. By Corollary 4.1.2, we conclude that $y \succ x$; a contradiction. □

4.2 Acyclic Orderings of the Vertices in Acyclic Digraphs

Acyclic digraphs play a very important role in both theory and applications of digraphs (the reader will see this fact in this and the following chapters of the book). Some basic properties of acyclic digraphs have been studied in Section 1.4 where we showed that every acyclic digraph D has an acyclic ordering of the vertices (Proposition 1.4.3). The purpose of this subsection is to show how to find an acyclic ordering fast[2].

Let D be an acyclic digraph and let $v_1, v_2, \dots, v_n$ be an ordering of the vertices in D. We recall that this ordering is acyclic if the existence of an arc v_iv_j in D implies $i < j$. By Proposition 1.4.3 every acyclic digraph has an acyclic ordering of its vertices. Now we demonstrate that using DFS one can find an acyclic ordering of the vertices of D in (optimal) linear time.

Below we assume that the input to the DFS algorithm is an acyclic digraph $D = (V, A)$. In the formal description of DFS let us add the following: $i := n + 1$ in the line 2 of the main body of DFS and $i := i - 1$, $v_i := v$ in the last line of DFS-PROC. We obtain the following algorithm which we denote by **DFSA**):

DFSA(D)
Input: A digraph $D = (V, A)$.
Output: An acyclic ordering $v_1, \dots, v_n$ of D.

1. For each $v \in V$ set $\text{pred}(v) :=$ **nil**, $\text{tvisit}(v) := 0$ and $\text{texpl}(v) := 0$.
2. Set $time := 0$, $i := n + 1$.
3. For each vertex $v \in V$ do: if $\text{tvisit}(v) = 0$ then perform DFSA-PROC(v).

DFSA-PROC(v)

1. Set $time := time + 1$, $\text{tvisit}(v) := time$.
2. For each $u \in N^+(v)$ do: if $\text{tvisit}(u) = 0$ then $\text{pred}(u) := v$ and perform DFSA-PROC(u).
3. Set $time := time + 1$, $\text{texpl}(v) := time$, $i := i - 1$, $v_i := v$.

[2] Notice that in the majority of literature an acyclic ordering is called a topological sorting. We feel that the name acyclic ordering is more appropriate, since no topology is involved. Knuth [481] was the first to give a linear time algorithm for topological sorting.

Theorem 4.2.1 *The algorithm DFSA correctly determines an acyclic ordering of any acyclic digraph in time $O(n+m)$.*

Proof: Since the algorithm is clearly linear (as DFS is linear), it suffices to show that the ordering $v_1, v_2, \ldots, v_n$ is acyclic. Observe that according to our algorithm

$$\text{texpl}(v_i) > \text{texpl}(v_j) \text{ if and only if } i < j. \tag{4.1}$$

Assume that D has an arc $v_k v_s$ such that $k > s$. Hence, $\text{texpl}(v_s) > \text{texpl}(v_k)$. The arc $v_k v_s$ is not a cross arc, because if it were, then by Proposition 4.1.1 and Corollary 4.1.2, the intervals for v_k and v_s would not intersect, i.e., v_k would be visited and explored before v_s would be visited; this and (4.1) make the existence of $v_k v_s$ impossible. The arc $v_k v_s$ is not a forward arc, because if it were, $\text{texpl}(v_s)$ would be smaller than $\text{texpl}(v_k)$. Therefore, $v_k v_s$ must be a backward arc, i.e., $v_k \succ v_s$. Thus, there is a (v_s, v_k)-path in D. This path and the arc $v_k v_s$ form a cycle, a contradiction. □

Figure 4.2 illustrates the result of applying DFSA to an acyclic digraph. The resulting acyclic ordering is z, w, u, y, x, v.

In Section 4.4 we apply DFSA to an arbitrary not necessarily acyclic digraph and see that the ordering $v_1, v_2, \ldots, v_n$ obtained by DFSA is very useful to determine the strong components of a digraph.

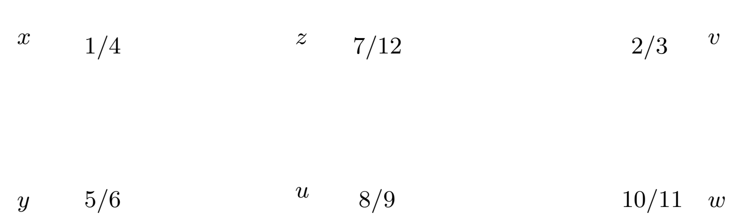


Figure 4.2 The result of applying DFSA to an acyclic digraph

4.3 Transitive Digraphs, Transitive Closures and Reductions

Recall that a digraph D is transitive if, for every pair xy and yz of arcs in D with $x \neq z$, the arc xz is also in D. Transitive digraphs form the underlying graph-theoretical model in a number of applications. For example, transitive oriented graphs correspond very naturally to partial orders (see Section 5.3 for some terminology on partial orders, the correspondence between transitive oriented graphs and partial orders and some basic results). The aim of this section is to give a brief overview of some properties of transitive digraphs as well as transitive closures and reductions of digraphs.

Clearly, a strong digraph D is transitive if and only if D is complete[3]. We have the following simple characterization of transitive digraphs; its proof is left as Exercise 4.2.

Proposition 4.3.1 *Let D be a digraph with an acyclic ordering $D_1, D_2, \ldots,$ D_p of its strong components. The digraph D is transitive if and only if each of D_i is complete, the digraph H obtained from D by contraction of $D_1, \ldots, D_p$ followed by deletion of multiple arcs is a transitive oriented graph, and $D = H[D_1, D_2, \ldots, D_p]$, where $p = |V(H)|$.* □

The **transitive closure** $TC(D)$ of a digraph D is a digraph with $V(TC(D)) = V(D)$ and, for distinct vertices u, v, the arc $uv \in A(TC(D))$ if and only if D has a (u, v)-path. Clearly, if D is strong then $TC(D)$ is a complete digraph. The uniqueness of the transitive closure of an arbitrary digraph is obvious. To compute the transitive closure of a digraph one can obviously apply the Floyd-Warshall algorithm in Chapter 2. To obtain a faster algorithm for the problem one can use the fact discovered by a number of researchers (see, e.g., the paper [238] by Fisher and Meyer, or [286] by Furman) that the transitive closure problem and the matrix multiplication problem are closely related: there exists an $O(n^a)$-algorithm, with $a \geq 2$, to compute the transitive closure of a digraph of order n if and only if the product of two boolean $n \times n$ matrices can be computed in $O(n^a)$ time. Coppersmith and Winograd [168] showed that there exists an $O(n^{2.376})$-algorithm for the matrix multiplication. Goralcikova and Koubek [333] designed an $O(nm_{red})$-algorithm to find the transitive closure of an acyclic digraph D with n vertices and m_{red} arcs in the transitive reduction of D (the notion of transitive reduction is introduced below). This algorithm was also studied and improved by Mehlhorn [561] and Simon [672].

An arc uv in a digraph D is **redundant** if there is a (u, v)-path in D which does not contain the arc uv. A **transitive reduction** of a digraph D is a spanning subdigraph H of D with no redundant arc such that the transitive closures of D and H coincide. Not every digraph D has a unique transitive reduction. Indeed, if D has a pair of hamiltonian cycles, then each of these cycles is a transitive reduction of D. Below we show that a transitive reduction of an acyclic digraph is unique, i.e., we may speak of *the* transitive reduction of an acyclic digraph. The **intersection of digraphs** $D_1, \ldots, D_k$ with the same vertex set V is the digraph H with vertex set V and arc set

[3] By the definition of a transitive digraph, a 2-cycle xyx does not force a loop at x and y.

$A(D_1) \cap \ldots \cap A(D_k)$. Similarly one can define the union of digraphs with the same vertex sets (see Section 1.3). Let $\mathcal{S}(D)$ be the set of all spanning subdigraphs L of D for which $TC(L) = TC(D)$.

Theorem 4.3.2 *[5] For an acyclic digraph D, there exists a unique digraph D' with the property that $TC(D') = TC(D)$ and every proper subdigraph H of D' satisfies $TC(H) \neq TC(D')$. The digraph D' is the intersection of digraphs in $\mathcal{S}$.*

The proof of this theorem, which is due to Aho, Garey and Ullman, follows from Lemmas 4.3.3 and 4.3.4.

Lemma 4.3.3 *Let D and H be a pair of acyclic digraphs on the same vertex set such that $TC(D) = TC(H)$ and $A(D) - A(H) \neq \emptyset$. Then, for every $e \in A(D) - A(H)$, we have $TC(D - e) = TC(D)$.*

Proof: Let $e = xy \in A(D) - A(H)$. Since $e \notin A(H)$, H must have an (x, y)-path passing through some other vertex, say z. Hence, D has an (x, z)-path P_{xz} and a (z, y)-path P_{zy}. If P_{xz} contains e, then D has a (y, z)-path. The existence of this path contradicts the existence of P_{zy} and the hypothesis that D is acyclic. Similarly, one can show that P_{zy} does not contain e. Therefore, $D - e$ has an (x, y)-path. Hence, $TC(D - e) = TC(D)$. □

Lemma 4.3.4 *Let D be an acyclic digraph. Then the set $\mathcal{S}(D)$ is closed under union and intersection.*

Proof: Let G, H be a pair of digraphs in $\mathcal{S}(D)$. Since $TC(G) = TC(H) = TC(D)$, $G \cup H$ is a subdigraph of $TC(D)$. The transitivity of $TC(D)$ now implies that $TC(G \cup H)$ is a subdigraph of $TC(D)$. Since G is a subdigraph of $G \cup H$, we have $TC(D)$ $(= TC(G))$ is a subdigraph of $TC(G \cup H)$. Thus, we conclude that $TC(G \cup H) = TC(D)$ and $G \cup H \in \mathcal{S}(D)$.

Now let $e_1, \ldots, e_p$ be the arcs of $G - A(G \cap H)$. By repeated application of Lemma 4.3.3, we obtain

$$TC(G - e_1 - e_2 - \ldots - e_p) = TC(G).$$

This means that $TC(G \cap H) = TC(G) = TC(D)$, hence $G \cap H \in \mathcal{S}(D)$. □

Aho, Garey and Ullman [5] proved that there exists an $O(n^a)$-algorithm, with $a \geq 2$, to compute the transitive closure of an arbitrary digraph D of order n if and only if a transitive reduction of D can be constructed in time $O(n^a)$. Therefore, we have

Proposition 4.3.5 *For an arbitrary digraph D, the transitive closure and a transitive reduction can be computed in time $O(n^{2.376})$.* □

Simon [673] described an $O(n+m)$-algorithm to find a transitive reduction of a strong digraph D. The algorithm uses DFS and two digraph transformations preserving $TC(D)$. This means that to have a linear time algorithm for finding transitive reductions of digraphs from a certain class $\mathcal{D}$, it suffices to design a linear time algorithm for the transitive reduction of strong component digraphs of digraphs in $\mathcal{D}$. (Recall that the strong component digraph $SC(D)$ of a digraph D is obtained by contracting every strong component of D to a vertex followed by deletion of parallel arcs.) Such algorithms are considered, e.g., in the paper [385] by Habib, Morvan and Rampon.

While Simon's linear time algorithm in [673] finds a *minimal* subdigraph D' of a strong digraph D such that $TC(D') = TC(D)$, no polynomial algorithm is known to find a subdigraph D'' of a strong digraph D with *minimum* number of arcs such that $TC(D'') = TC(D)$. This is not surprising due to the fact that the corresponding optimization problem is $\mathcal{NP}$-hard. Indeed, the problem to verify whether a strong digraph D of order n has a subdigraph D'' of size n such that $TC(D'') = TC(D)$ is equivalent to the hamiltonian cycle problem, which is $\mathcal{NP}$-complete by Theorem 5.0.1.

A subdigraph D'' of a digraph D with minimum number of arcs such that $TC(D'') = TC(D)$ is sometimes called a **minimum equivalent subdigraph** of D. By the above discussion, we see that a minimum equivalent subdigraph of an acyclic digraph is unique and can be found in polynomial time. This means that the main difficulty of finding a minimum equivalent subdigraph of an arbitrary digraph D lies in finding such subdigraphs for the strong components of D. This issue is addressed in Section 6.11 for some classes of digraphs studied in this chapter. For the classes in Section 6.11, the minimum equivalent subdigraph problem is polynomial time solvable.

4.4 Strong Digraphs

In many problems on digraphs it suffices to consider the case of strong digraphs. For example, if we wish to find a cycle through a given vertex x in a digraph D, we need only consider the strong component of D containing x. Furthermore, certain properties, such as being hamiltonian, imply that the digraph in question must be strong. The aim of this section is to develop a fast algorithm for finding strong components in a digraph and in particular to recognize strong digraphs.

Tarjan [688] was the first to obtain an $O(n + m)$-algorithm to compute the strong components of a digraph. We start this section by presenting this algorithm, then we discuss its complexity and prove its correctness. Our presentation is adapted from the book [169] by Cormen, Leiserson, and Rivest.

SCA(D)
Input: A digraph D.
Output: The vertex sets of strong components of D.

1. Call DFSA(D) to compute the 'acyclic' ordering $v_1, v_2, \ldots, v_n$.
2. Compute the converse D' of D.
3. Call DFS(D'), but in the main loop of DFS consider the vertices according to the ordering $v_1, v_2, \ldots, v_n$. In the process of DFS(D') output the vertices of each DFS tree as the vertices of a strong component of D.

Figure 4.3 illustrates the strong component algorithm (SCA). Clearly, the complexity of SCA is $O(n+m)$. It is more difficult to establish the correctness of SCA. Several lemmas are needed.

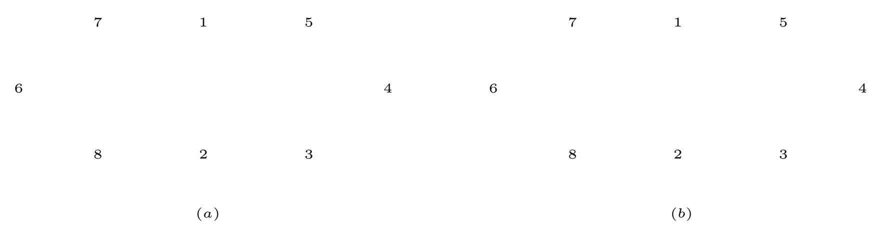


Figure 4.3 (a) A digraph D; the order of vertices found by DFSA is shown. (b) The converse D' of D; the bold arcs are the arcs of a DFS forest for D'.

The proof of our first lemma is simple and left as an exercise, Exercise 4.3.

Lemma 4.4.1 *If a pair x, y of vertices belongs to the same strong component S of a digraph D, then the vertices of every path between x and y are in S.* □

Lemma 4.4.2 *In any execution of DFS on a digraph, all vertices of the same strong component are placed in the same DFS tree.*

Proof: Let S be a strong component of a digraph D, let r be the first vertex of S visited by DFS and let x be another vertex of S. Consider the time tvisit(r) of DFS. By Lemma 4.4.1, all vertices on an (r, x)-path belong to S and apart from r are unvisited. Thus, by Proposition 4.1.3, x belongs to the same DFS tree as r. □

In the rest of this section tvisit(u) and texpl(u) are the time-stamps calculated during the first step of SCA (recall that these depends on the order in which the DFS routine visits the vertices). The **forefather** $\phi(u)$ of a vertex u is the vertex w reachable from u such that texpl(w) is maximum.

Since u is reachable from itself, we have

$$\text{texpl}(u) \leq \text{texpl}(\phi(u)). \tag{4.2}$$

Clearly, by the definition of forefather

$$\text{if } v \text{ is reachable from } u, \text{ then } \text{texpl}(\phi(v)) \leq \text{texpl}(\phi(u)). \tag{4.3}$$

The next lemma gives a justification for the term 'forefather'.

Lemma 4.4.3 *In any execution DFS on a digraph D, every vertex $u \in V(D)$ is a descendant of its forefather $\phi(u)$.*

Proof: If $\phi(u) = u$, this lemma is trivially true. Thus, assume that $\phi(u) \neq u$ and consider the time tvisit(u) of DFS for D. Look at the status of $\phi(u)$. The vertex $\phi(u)$ cannot be already explored as that would mean texpl$(\phi(u)) <$ texpl(u), which is impossible. If $\phi(u)$ is already visited but not explored, then, by Corollary 4.1.2, u is a descendant of $\phi(u)$ and the lemma is proved.

It remains to show that $\phi(u)$ has been indeed visited before time tvisit(u). Assume it is not true and consider a $(u, \phi(u))$-path P. If every vertex of P except for u has not been visited yet (at the time tvisit(u)), then by Proposition 4.1.3 $\phi(u)$ is a descendant of u, i.e. texpl$(\phi(u)) <$ texpl(u), which is impossible. Suppose now that there is a vertex x in P apart from u which has been visited. Assume that x is the last such vertex in P (going from u towards $\phi(u)$). Clearly, x has not been explored yet (as x dominates an unvisited vertex). By Proposition 4.1.3 applied to $P[x, \phi(u)]$, $\phi(u)$ is a descendant of x. Thus, texpl$(\phi(u)) <$ texpl(x), which contradicts the definition of $\phi(u)$.

Thus, $\phi(u)$ has been indeed visited before time tvisit(u), which completes the proof of this lemma. □

Lemma 4.4.4 *For every application of DFS to a digraph D and for every $u \in V(D)$, the vertices u and $\phi(u)$ belong to the same strong component of D.*

Proof: There is a $(u, \phi(u))$-path by the definition of forefather. The existence of a path from $\phi(u)$ to u follows from Lemma 4.4.3. □

Now we show a stronger version of Lemma 4.4.4.

Lemma 4.4.5 *For every application of DFS to a digraph D and for every pair $u, v \in V(D)$, the vertices u and v belong to the same strong component of D if and only if $\phi(u) = \phi(v)$.*

Proof: If u and v belong to the same strong component of D, then every vertex reachable from one of them is reachable from the other. Hence, $\phi(u) = \phi(v)$. By Lemma 4.4.4, u and v belong to the same strong components as their forefathers. Thus, $\phi(u) = \phi(v)$ implies that u and v are in the same strong component of D. □

Theorem 4.4.6 *The algorithm SCA correctly finds the strong components of a digraph D.*

Proof: We prove by induction on the number of DFS trees found in the execution of DFS on D' that the vertices of each of these trees induce a strong component of D. Each step of the inductive argument proves that the vertices of a DFS tree formed in D' induce a strong component of D provided the vertices of each of the previously formed DFS trees induce a strong component of D. The basis for induction is trivial, since the first tree obtained has no previous trees, and hence the assumption holds trivially. Recall that by the description of SCA, in the second application of DFS, we always start a new DFS tree from the vertex which currently has the highest value of texpl among vertices not yet in the DFS forest under construction.

Consider a DFS tree T with root r produced in DFS(D'). By the definition of a forefather $\phi(r) = r$. Indeed, r is reachable from itself and has the maximum texpl among the vertices reachable from r. Let $S(r) = \{v \in V(D) : \phi(v) = r\}$. We now prove that

$$V(T) = S(r). \tag{4.4}$$

By Lemmas 4.4.2 and 4.4.5, every vertex in $S(r)$ is in the same DFS tree. Since $r \in S(r)$ and r is the root of T, every vertex in $S(r)$ belongs to T. To complete the proof of (4.4), it remains to show that, if $u \in V(T)$, then $u \in S(r)$, namely, if $\mathrm{texpl}(\phi(x)) \neq \mathrm{texpl}(r)$, then x is not placed in T. Suppose that $\mathrm{texpl}(\phi(x)) \neq \mathrm{texpl}(r)$ for some vertex x. By induction hypothesis, we may assume that $\mathrm{texpl}(\phi(x)) < \mathrm{texpl}(r)$, since otherwise x is placed in the tree with root $\phi(x) \neq r$. If x was placed in T, then r would be reachable from x. By (4.3) and $\phi(r) = r$, this would mean $\mathrm{texpl}(x) \geq \mathrm{texpl}(\phi(r)) = \mathrm{texpl}(r)$, a contradiction. □

4.5 Line Digraphs

For a directed pseudograph D, the **line digraph** $Q = L(D)$ has vertex set $V(Q) = A(D)$ and arc set

$$A(Q) = \{ab : a, b \in V(Q),\ \text{the head of } a \text{ coincides with the tail of } b\}.$$

A directed pseudograph H is a **line digraph** if there is a directed pseudograph D such that $H = L(D)$. See Figure 4.4. Clearly, line digraphs do not have parallel arcs; moreover, the line digraph $L(D)$ has a loop at a vertex $a \in A(D)$ if and only if a is a loop in D.

The following theorem provides a number of equivalent characterizations of line digraphs. Of these characterizations, (ii) is due to Harary and Norman [403], (iii) to Heuchenne [425], and (iv) and (v) to Richards [634]; conditions (ii) and (iii) have each been rediscovered several times, see the survey

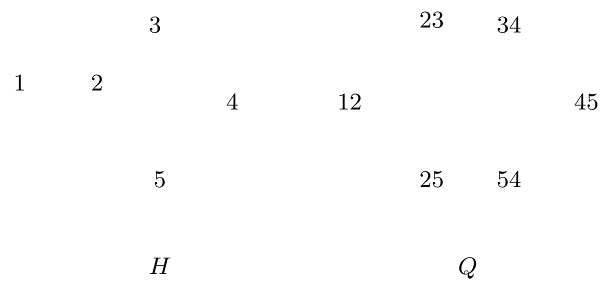


Figure 4.4 A digraph H and its line digraph $Q = L(H)$.

[419] by Beineke and Hemminger. The proof presented here is adapted from [419]. For an $n \times n$-matrix $M = [m_{ik}]$, a row i is **orthogonal** to a row j if $\sum_{k=1}^{n} m_{ik}m_{jk} = 0$. One can give a similar definition of orthogonal columns.

Theorem 4.5.1 *Let D be a directed pseudograph with vertex set $\{1, 2, \ldots, n\}$ and with no parallel arcs and let $M = [m_{ij}]$ be its adjacency matrix (i.e., the $n \times n$-matrix such that $m_{ij} = 1$, if $ij \in A(D)$, and $m_{ij} = 0$, otherwise). Then the following assertions are equivalent:*

(i) D is a line digraph;
(ii) there exist two partitions $\{A_i\}_{i\in I}$ and $\{B_i\}_{i\in I}$ of $V(D)$ such that $A(D) = \cup_{i\in I} A_i \times B_i$[4];
(iii) if vw, uw and ux are arcs of D, then so is vx;
(iv) any two rows of M are either identical or orthogonal;
(v) any two columns of M are either identical or orthogonal.

Proof: We show the following implications and equivalences: (i) $\Leftrightarrow$ (ii), (ii) $\Rightarrow$ (iii), (iii) $\Rightarrow$ (iv), (iv) $\Leftrightarrow$ (v), (iv) $\Rightarrow$ (ii).

(i) $\Rightarrow$ (ii). Let $D = L(H)$. For each $v_i \in V(H)$, let A_i and B_i be the sets of in-coming and out-going arcs at v_i, respectively. Then the arc set of the subdigraph of D induced by $A_i \cup B_i$ equals $A_i \times B_i$. If $ab \in A(D)$, then there is an i such that $a = v_jv_i$ and $b = v_iv_k$. Hence, $ab \in A_i \times B_i$. The result follows.

(ii) $\Rightarrow$ (i). Let Q be the directed pseudograph with ordered pairs (A_i, B_i) as vertices, and with $|A_j \cap B_i|$ arcs from (A_i, B_i) to (A_j, B_j) for each i and j (including $i = j$). Let σ_{ij} be a bijection from $A_j \cap B_i$ to this set of arcs (from (A_i, B_i) to (A_j, B_j)) of Q. Then the function σ defined on $V(D)$ by taking σ to be σ_{ij} on $A_j \cap B_i$ is a well-defined function of $V(D)$ into $V(L(Q))$, since $\{A_j \cap B_i\}_{i,j\in I}$ is a partition of $V(D)$. Moreover, σ is a bijection since every σ_{ij} is a bijection. Furthermore, it is not difficult to see that σ is an isomorphism from D to $L(Q)$ (this is left as Exercise 4.4).

[4] Recall that $X \times Y = \{(x, y) : x \in X, y \in Y\}$.

(ii) $\Rightarrow$ (iii). If vw, uw and ux are arcs of D, then there exist i, j such that $\{u, v\} \subseteq A_i$ and $\{w, x\} \subseteq B_j$. Hence, $(v, x) \in A_i \times B_j$ and $vx \in D$.

(iii) $\Rightarrow$ (iv). Assume that (iv) does no hold. This means that some rows, say i and j, are neither identical nor orthogonal. Then there exist k, h such that $m_{ik} = m_{jk} = 1$ and $m_{ih} = 1, m_{jh} = 0$ (or vice versa). Hence, ik, jk, ih are in $A(D)$ but jh is not. This contradicts (iii).

(iv) $\Leftrightarrow$ (v). Both (iv) and (v) are equivalent to the statement:

$$\text{for all } i, j, h, k, \text{ if } m_{ih} = m_{ik} = m_{jk} = 1, \text{ then } m_{jh} = 1.$$

(iv) $\Rightarrow$ (ii). For each i and j with $m_{ij} = 1$, let $A_{ij} = \{h : \ m_{hj} = 1\}$ and $B_{ij} = \{k : \ m_{ik} = 1\}$. Then, by (iv), A_{ij} is the set of vertices in D whose row vectors in M are identical to the ith row vector, whereas B_{ij} is the set of vertices in D whose column vectors in M are identical to the jth column vector (we use the previously proved fact that (iv) and (v) are equivalent). Thus, $A_{ij} \times B_{ij} \subseteq A(D)$, and moreover $A(D) = \cup\{A_{ij} \times B_{ij} : \ m_{ij} = 1\}$. By the orthogonality condition, A_{ij} and A_{hk} are either equal or disjoint, as are B_{ij} and B_{hk}. For zero row vector i in M, let A_{ij} be the set of vertices whose row vector in M is the zero vector, and let $B_{ij} = \emptyset$. Doing the same with the zero column vectors of M completes the partition as in (ii). □

The characterizations (ii)-(v) all imply polynomial algorithms to verify whether a given directed pseudograph is a line digraph. This fact is obvious regarding (iii)-(v); it is slightly more difficult to see that (ii) can be used to construct a very effective polynomial algorithm. We actually design such an algorithm for acyclic digraphs (as a pair of procedures illustrated by an example) just after Proposition 4.5.3. The criterion (iii) also provides the following characterization of line digraphs in terms of forbidden induced subdigraphs. Its proof is left as Exercise 4.5.

Corollary 4.5.2 *A directed pseudograph D is a line digraph if and only if D does not contain, as an induced subdigraph, any directed pseudograph that can be obtained from one of the directed pseudographs in Figure 4.5 (dotted arcs are missing) by adding zero or more arcs (other than the dotted ones).*

Observe that the digraph of order 4 in Figure 4.5 corresponds to the case of distinct vertices in Part (iii) of Theorem 4.5.1, and the two directed pseudographs of order 2 correspond to the cases $x = u \neq v = w$ and $u = w \neq v = x$, respectively.

Clearly, Theorem 4.5.1 implies a set of characterizations of the line digraphs of digraphs (without parallel arcs and loops). This can be found in [419]. Several characterizations of special classes of line digraphs and iterated line digraphs can be found in surveys by Hemminger and Beineke [419] and Prisner [614].

Many applications of line digraphs deal with the line digraphs of special families of digraphs, for example regular digraphs, in general, and complete

Figure 4.5 Forbidden directed pseudographs.

digraphs, in particular, see e.g., the papers [207] by Du, Lyuu and Hsu and [236] by Fiol, Yebra and Alegre. In Section 4.7, we need the following characterization, due to Harary and Norman, of the line digraphs of acyclic directed multigraphs. It is a specialization of Parts (i) and (ii) of Theorem 4.5.1. The proof is left as (an easy) Exercise 4.6.

Proposition 4.5.3 *[403] A digraph D is the line digraph of an acyclic directed multigraph if and only if D is acyclic and there exist two partitions $\{A_i\}_{i\in I}$ and $\{B_i\}_{i\in I}$ of $V(D)$ such that $A(D) = \cup_{i\in I} A_i \times B_i$.* $\square$

We will now show how Proposition 4.5.3 can be used to recognize very effectively whether a given acyclic digraph R is the line digraph of another acyclic directed multigraph H, i.e., $R = L(H)$. The two procedures, which we construct and illustrate by Figure 4.8 can actually be used to recognize and represent (that is, to construct H such that $R = L(H)$) arbitrary line digraphs (see Theorem 4.5.1(i) and (ii)).

We first use Proposition 4.5.3 to check whether H above exists. The following procedure **Check-H** can be applied. Initially, all arcs and vertices of R are not marked. At every iteration, we choose an arc uv in R, which is not marked yet, and mark all vertices in $N^+(u)$ by 'B', all vertices in $N^-(v)$ by 'A' and all arcs in $(N^-(v), N^+(u))_R$ by 'C'. If $(N^-(v), N^+(u))_R \neq N^-(v) \times N^+(u)$ or if we mark a certain vertex or arc twice (starting from another arc $u'v'$) by the same symbol, then this procedure stops as there is no H such that $L(H) = R$. (We call these conditions **obstructions**.) If this procedure is performed to the end (i.e. every vertex and arc received a mark), then such H exists. It is not difficult to see, using Proposition 4.5.3, that Check-H correctly verifies whether H exists or not.

To illustrate Check-H, consider the digraph R_0 of Figure 4.8(a). Suppose that we choose the arc ab first. Then ab is marked, at the first iteration, together with the arcs af and ag. The vertex a receives 'A', the vertices b, f, g get 'B'. Suppose that fi is chosen at the second iteration. Then the arcs fh, fi, gh, gi are all marked at this iteration. The vertices f, g receive 'A', the vertices h, i 'B'. Suppose that bc is chosen at the third iteration. We see that this arc is the only arc marked at this iteration. The vertex b receives 'A', the vertex c 'B'. Finally, say, ce is chosen. Then both cd and ce are marked. The vertex c gets 'A', the vertices d, e receive 'B'. Thus, all arcs became marked with no obstruction happened. This means that there exists a digraph H_0 such that $H_0 = L(R_0)$.

Suppose now that H does exist. The following procedure **Build-H** constructs such a directed multigraph H. By Proposition 4.5.3, if H exists, then all arcs of R can be partitioned into arc sets of bipartite tournaments with partite sets A_i and B_i and arc sets $A_i \times B_i$. Let us denote these digraphs by $T_1, \ldots, T_k$. (They can be computed by Check-H if we mark every $(N^-(v), N^+(u))_R$ not only by 'C' but also by a second mark 'i' starting from 1 and increasing by 1 at each iteration of the procedure.) We construct H as follows. The vertex set of H is $\{t_0, t_1, \ldots, t_k, t_{k+1}\}$. The arcs of H are obtained by the following procedure. For each vertex v of R, we append one arc a_v to H according to the rules below:

(a) If $d_R(v) = 0$, then $a_v := (t_0, t_{k+1})$;
(b) If $d_R^+(v) > 0, d_R^-(v) = 0$, then $a_v := (t_0, t_i)$, where i is the index of T_i such that $v \in A_i$;
(c) If $d_R^+(v) = 0, d_R^-(v) > 0$, then $a_v := (t_j, t_{k+1})$, where j is the index of T_j such that $v \in B_j$;
(d) If $d_R^+(v) > 0, d_R^-(v) > 0$, then $a_v := (t_i, t_j)$, where i and j are the indices of T_i and T_j such that $v \in A_j \cap B_i$.

It is straightforward to verify that $R = L(H)$. Note that Build-H always constructs H with only one vertex of in-degree zero and only one vertex of out-degree zero.

To illustrate Build-H, consider R_0 of Figure 4.8 once again. Earlier we showed that there exists H_0 such that $R_0 = L(H_0)$. Now we will construct H_0. The previous procedure applied to verify the existence of H_0 has implicitly constructed the digraphs $T_1 = (\{a, b, f, g\}, \{ab, af, ag\})$, $T_2 = (\{f, g, h, i\}, \{fh, fi, gh, gi\})$, $T_3 = (\{b, c\}, \{bc\})$, $T_4 = (\{c, d, e\}, \{cd, ce\})$. Thus, H_0 has vertices $t_0, \ldots, t_5$. Considering the vertices of R_0 in the lexicographic order, we obtain the following arcs of H_0 (in this order):

$$t_0t_1, t_1t_3, t_3t_4, t_4t_5, t_4t_5, t_1t_2, t_1t_2, t_2t_5, t_2t_5.$$

The directed multigraph H_0 is depicted in Figure 4.8(c). It is easy to check that $R_0 = L(H_0)$.

The iterated line digraphs are defined recursively: $L^1(D) = L(D)$, $L^{k+1}(D) = L(L^k(D))$, $k \geq 1$. It is not difficult to prove by induction (Exercise 4.8) that $L^k(D)$ is isomorphic to the digraph H, whose vertex set consists of walks of D of length k and a vertex $v_0v_1\dots v_k$ (which is a walk in D) dominates the vertex $v_1v_2\dots v_kv_{k+1}$ for every $v_{k+1} \in V(D)$ such that $v_kv_{k+1} \in A(D)$. New characterizations of line digraphs and iterated line digraphs are given by Liu and West [518].

The following proposition can be proved by induction on $k \geq 1$ (Exercise 4.10).

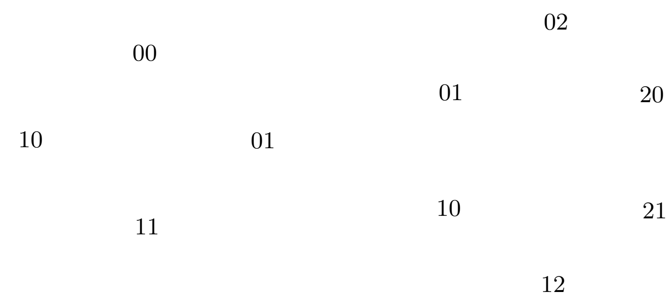
Proposition 4.5.4 *Let D be a strong d-regular digraph ($d > 1$) of order n and diameter t. Then $L^k(D)$ is of order d^kn and diameter $t + k$.* □

4.6 The de Bruijn and Kautz Digraphs and their Generalizations

The following problem is of importance in network design. Given positive integers n and d, construct a digraph D of order n and maximum out-degree at most d such that $\mathrm{diam}(D)$ is as small as possible and the vertex-strong connectivity $\kappa(D)$ is as large as possible. So we have a 2-objective optimization problem. For such a problem, in general, no solution can maximize/minimize both objective functions. However, for this specific problem, there are solutions, which (almost) maximize/minimize both objective functions. The aim of this section is to introduce these solutions, the de Bruijn and Kautz digraphs, as well as some of their generalizations. For more information on the above classes of digraphs, the reader may consult the survey [204] by Du, Cao and Hsu. For applications of these digraphs in design of parallel architectures and large packet radio networks, see e.g. the papers [113] by Bermond and Hell, [114] by Bermond and Peyrat and [649] by Samatan and Pradhan.

Let V be the set of vectors with t coordinates, $t \geq 2$, each taken from $\{0, 1, \dots, d-1\}$, $d \geq 2$. The **de Bruijn digraph** $D_B(d, t)$ is the directed pseudograph with vertex set V such that $(x_1, x_2, \dots, x_t)$ dominates $(y_1, y_2, \dots, y_t)$ if and only if $x_2 = y_1, x_3 = y_2, \dots, x_t = y_{t-1}$. See Figure 4.6 (a). Let $D_B(d, 1)$ be the complete digraph of order d with loop at every vertex.

These directed pseudographs are named after de Bruijn who was the first to consider them in [185]. Clearly, $D_B(d, t)$ has d^t vertices and the out-pseudodegree and in-pseudodegree of every vertex of $D_B(d, t)$ equal d. This directed pseudograph has no parallel arcs and contains a loop at every vertex for which all coordinates are the same. It is natural to call $D_B(d, t)$



(a) (b)

Figure 4.6 (a) The de Bruijn digraph $D_B(2,2)$; (b) The Kautz digraph $D_K(2,2)$.

***d*-pseudoregular** (recall that in the definition of semi-degrees we do not count loops).

Since $D_B(d,t)$ has loops at some vertices, the vertex-strong connectivity of $D_B(d,t)$ is at most $d-1$ (indeed, the loops can be deleted without the vertex-strong connectivity being changed). Imase, Soneoka and Okada [444] proved that $D_B(d,t)$ is $(d-1)$-strong, and moreover, for every pair $x \neq y$ of vertices there exist $d-1$ internally disjoint (x,y)-paths of length at most $t+1$. To prove this result we will use the following two lemmas. The proof of the first lemma, due to Fiol, Yebra and Alegre, is left as Exercise 4.11.

Lemma 4.6.1 *[236] For $t \geq 2$, $D_B(d,t)$ is the line digraph of $D_B(d,t-1)$.* □

Lemma 4.6.2 *Let x,y be distinct vertices of $D_B(d,t)$ such that $x \to y$. Then, there are $d-2$ internally disjoint (x,y)-paths different from xy, each of length at most $t+1$.*

Proof: Let $x = (x_1, x_2, \ldots, x_t)$ and $y = (x_2, \ldots, x_t, y_t)$. Consider the walk W_k given by $W_k = (x_1, x_2, \ldots, x_t), (x_2, \ldots, x_t, k), (x_3, \ldots, x_t, k, x_2), \ldots, (k, x_2, \ldots, x_t), (x_2, \ldots, x_t, y_t)$, where $k \neq x_1, y_t$. For each k, every internal vertex of W_k has coordinates forming the same multiset $M_k = \{x_2, \ldots, x_t, k\}$. Since for different k, the multisets M_k are different, the walks W_k are internally disjoint. Each of these walks is of length $t+1$. Therefore, by Proposition 1.4.1, $D_B(d,t)$ contains $d-2$ internally disjoint (x,y)-paths P_k with $A(P_k) \subseteq A(W_k)$. Since $k \neq x_1, y_t$, we may form the paths P_k such that none of them coincides with xy. □

Theorem 4.6.3 *[444] For every pair x,y of distinct vertices of $D_B(d,t)$, there exist $d-1$ internally disjoint (x,y)-paths, one of length at most t and the others of length at most $t+1$.*

Proof: By induction on $t \geq 1$. Clearly, the claim holds for $t = 1$ since $D_B(d, 1)$ contains, as spanning subdigraph, $\overleftrightarrow{K}_d$. For $t \geq 2$, by Lemma 4.6.1, we have that

$$D_B(d, t) = L(D_B(d, t-1)). \tag{4.5}$$

Let x, y be a pair of distinct vertices in $D_B(d, t)$ and let e_x, e_y be the arcs of $D_B(d, t-1)$ corresponding to vertices x, y due to (4.5). Let u be the head of e_x and let v be the tail of e_y.

If $u \neq v$, by the induction hypothesis, $D_B(d, t-1)$ has $d-1$ internally disjoint (u, v)-paths, one of length at most $t-1$ and the others of length at most t. The arcs of these paths together with arcs e_x and e_y correspond to $d-1$ internally disjoint (x, y)-paths in $D_B(d, t)$, one of length at most t and the others of length at most $t+1$.

If $u = v$, we have $x \rightarrow y$ in $D_B(d, t-1)$. It suffices to apply Lemma 4.6.2 to see that there are $d-1$ internally disjoint (x, y)-paths in $D_B(d, t)$, one of length one and the others of length at most $t+1$. $\square$

By this theorem and Corollary 7.3.2, we conclude that $\kappa(D_B(d, t)) = d-1$. From Theorem 4.6.3 and Proposition 2.4.3, we obtain immediately the following simple, yet important property.

Proposition 4.6.4 *The de Bruijn digraph $D_B(d, t)$ achieves the minimum value t of diameter for directed pseudographs of order d^t and maximum out-degree at most d.* $\square$

For $t \geq 2$, the **Kautz digraph** $D_K(d, t)$ is obtained from $D_B(d+1, t)$ by deletion of all vertices of the form $(x_1, x_2, \ldots, x_t)$ such that $x_i = x_{i+1}$ for some i. See Figure 4.6 (b). Define $D_K(d, 1) := \overleftrightarrow{K}_{d+1}$. Clearly, $D_K(d, t)$ has no loops and is a d-regular digraph. Since we have $d+1$ choices for the first coordinate of a vertex in $D_K(d, t)$ and d choices for each of the other coordinates, the order of $D_K(d, t)$ is $(d+1)d^{t-1} = d^t + d^{t-1}$. It is easy to see that Proposition 4.6.4 holds for the Kautz digraphs as well.

The following lemmas are analogous to Lemmas 4.6.1 and 4.6.2. Their proofs are left as Exercises 4.12 and 4.13.

Lemma 4.6.5 *For $t \geq 2$, the Kautz digraph $D_K(d, t)$ is the line digraph of $D_K(d, t-1)$.* $\square$

Lemma 4.6.6 *Let xy be an arc in $D_K(d, t)$. There are $d-1$ internally disjoint (x, y)-paths different from xy, one of length at most $t+2$ and the others of length at most $t+1$.* $\square$

The following result due to Du, Cao and Hsu [204] shows that the Kautz digraphs are better, in a sense, than de Bruijn digraphs from the local vertex-strong connectivity point of view. This theorem can be proved similarly to Theorem 4.6.3 and is left as Exercise 4.14.

Theorem 4.6.7 *[204] Let x, y be distinct vertices of $D_K(d,t)$. Then there are d internally disjoint (x,y)-paths in $D_K(d,t)$, one of length at most t, one of length at most $t+2$ and the others of length at most $t+1$.* $\square$

This theorem implies that $D_K(d,t)$ is d-strong.

The de Bruijn digraphs were generalized independently by Imase and Itoh [441] and Reddy, Pradhan and Kuhl [624] in the following way. We can transform every vector $(x_1, x_2, \ldots, x_t)$ with coordinates from $Z_d = \{0, 1, \ldots, d-1\}$ into an integer from $Z_{d^t} = \{0, 1, \ldots, d^t - 1\}$ using the polynomial $P(x_1, x_2, \ldots, x_t) = x_1 d^{t-1} + x_2 d^{t-2} + \ldots + x_t$. It is easy to see that this polynomial provides a bijection from Z_d^t to Z_{d^t}. Moreover, for $i, j \in Z_{d^t}$, $i \rightarrow j$ in $D_B(d,t)$ if and only if $j \equiv di + k \pmod{d^t}$ for some $k \in Z_d$.

Let d, n be two natural numbers such that $d < n$. The **generalized de Bruijn digraph** $D_G(d,n)$ is a directed pseudograph with vertex set Z_n and arc set

$$\{(i, di + k \pmod{n}) : \ i, k \in Z_d\}.$$

For example, $V(D_G(2,5)) = \{0,1,2,3,4\}$ and $A(D_G(2,5)) = \{(0,0),(0,1),(1,2),(1,3),(2,4),(2,0),(3,1),(3,2),(4,3),(4,4)\}$.

Clearly, $D_G(d,n)$ is d-pseudoregular. It is not difficult to show that $\mathrm{diam}(D_G(d,n)) \leq \lceil \log_d n \rceil$. By Proposition 2.4.3, a digraph of maximum out-degree at most $d \geq 2$ and order n has a diameter at least $\lfloor \log_d n(d-1)+1 \rfloor$. Thus, the generalized de Bruijn digraphs are of optimal or almost optimal diameter. It was proved, by Imase, Soneoka and Okada [443], that $D_G(d,n)$ is $(d-1)$-strong. It follows from these results that the generalized de Bruijn digraphs have almost minimum diameter and almost maximum vertex-strong connectivity.

The Kautz digraphs were generalized by Imase and Itoh [442]. Let n, d be two natural numbers such that $d < n$. The Imase-Itoh digraph $D_I(d,n)$ is the digraph with vertex set Z_n such that $i \rightarrow j$ if and only if $j \equiv -d(i+1)+k \pmod{n}$ for some $k \in Z_d$. It has been shown (for a brief account, see the paper [204]) by Du, Cao and Hsu, that $D_I(d,n)$ are of (almost) optimal diameter and vertex-strong connectivity.

Du, Hsu and Hwang [206] suggested a concept of digraphs extending both generalized the de Bruijn digraphs and the Imase-Ito digraphs. Let d, n be two natural numbers such that $d < n$. Given $q \in Z_n - \{0\}$ and $r \in Z_n$, **consecutive-*d* digraph** $D(d,n,q,r)$ is the directed pseudograph with vertex set Z_n such that $i \rightarrow j$ if and only if $j \equiv qi + r + k \pmod{n}$ for some $k \in Z_d$. Several results on diameter, vertex- and arc-strong connectivity and other properties of consecutive-d digraphs are given in [204]. In Section 5.11, we provide results on hamiltonicity of consecutive-d digraphs.

4.7 Series-Parallel Digraphs

In this section we study vertex series-parallel digraphs and arc series-parallel directed multigraphs. Vertex series-parallel digraphs were introduced by Lawler [510], and Monma and Sidney [568] as a model for scheduling problems. While vertex series-parallel digraphs continue to play an important role for the design of efficient algorithms in scheduling and sequencing problems, they have been extensively studied in their own right as well as in relations to other optimization problems (cf. the papers [36] by Baffi and Petreschi, [116] by Bertolazzi, Cohen, Di Battista, Tamassia and Tollis, [633] by Rendl and [682] by Steiner). Arc series-parallel directed multigraphs were introduced even earlier (than vertex series-parallel digraphs) by Duffin [209] as a mathematical model of electrical networks.

For an acyclic digraph D, let F_D (I_D) be the set of vertices of D of out-degree (in-degree) zero. To define vertex series-parallel digraphs, we first introduce **minimal vertex series-parallel (MVSP) digraphs** recursively.

The digraph of order one with no arc is an MVSP digraph. If $D_1 = (V, A)$, $D_2 = (U, B)$ is a pair of MVSP digraphs, so are the acyclic digraphs constructed by each of the following operations (see Figure 4.7):

(a) **Parallel composition:** $P = (V \cup U, A \cup B)$;
(b) **Series composition:** $S = (V \cup U, A \cup B \cup (F_D \times I_H))$.

It is interesting to note that we can embed every MVSP digraph D into the Cartesian plane such that if vertices u, v have coordinates (x_u, y_u) and (x_v, y_v), respectively, then there is a (u, v)-path in D if and only if $x_u \leq x_v$ and $y_u \leq y_v$. The proof of this non-difficult fact is given in the paper [726] by Valdes, Tarjan, and Lawler; see Exercise 4.15. See also Figure 4.9.

An acyclic digraph D is a **vertex series-parallel (VSP)** digraph if the transitive reduction of D is an MVSP digraph (see Subsection 4.3 for the definition of the transitive reduction). See Figure 4.8.

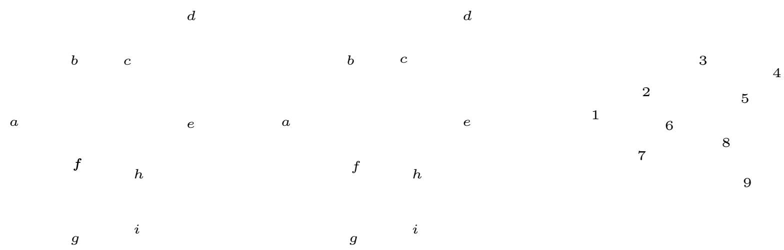
The following class of acyclic directed multigraphs, **arc series-parallel (ASP) directed multigraphs**, is related to VSP digraphs. The digraph $\vec{P}_2$ is an ASP directed multigraph. If D_1, D_2 is a pair of ASP directed multigraphs, then so are acyclic directed multigraphs constructed by each of the following operations (see Figure 4.10):

(a) **Two-terminal parallel composition:** Choose a vertex u_i of out-degree zero in D_i and a vertex v_i of in-degree zero in D_i for $i = 1, 2$. Identify u_1 with u_2 and v_1 with v_2;
(b) **Two-terminal series composition:** Choose $u \in F_{D_1}$ and $v \in I_{D_2}$ and identify u with v.

Figure 4.7 (De)construction of an MVSP digraph R_0 by series and parallel (de)compositions.

We refer the reader to the book [97] by Battista, Eades, Tamassia and Tollis for several algorithms for drawing graphs nicely, in particular drawing of ASP digraphs.

The next result shows a relation between the classes of digraphs introduced above.



(a) (b) (c)

Figure 4.8 Series-parallel directed multigraphs: (a) an MVSP digraph R_0, (b) a VSP digraph R_1, (c) an AVSP directed multigraph H_0.

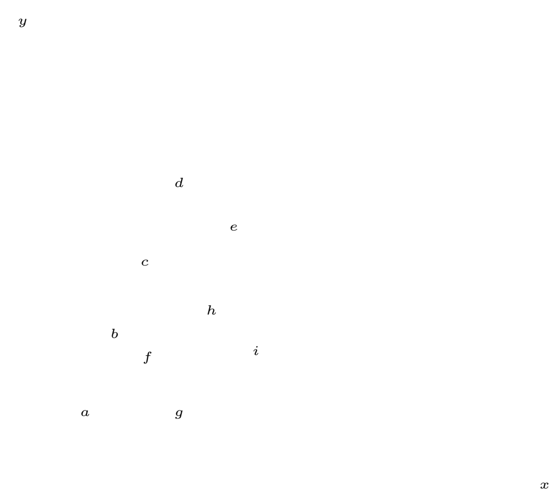


Figure 4.9 The MVSP digraph R_0 of Figure 4.7 embedded into the Cartesian plane such that for every (u, v)-path in R_0 we have $x_u \leq x_v$ and $y_u \leq y_v$ (and vice versa).

Theorem 4.7.1 *An acyclic directed multigraph D with a unique vertex of out-degree zero and a unique vertex of in-degree zero is ASP if and only if $L(D)$ is an MVSP digraph.*

Proof: This can be proved easily by induction on $|A(D)|$ using the following two facts:

(i) $L(\vec{P}_2) = \vec{P}_1$, which is an MVSP digraph;

(ii) The line digraph of the two-terminal series (parallel) composition of D_1 and D_2 is the series (parallel) composition of $L(D_1)$ and $L(D_2)$. □

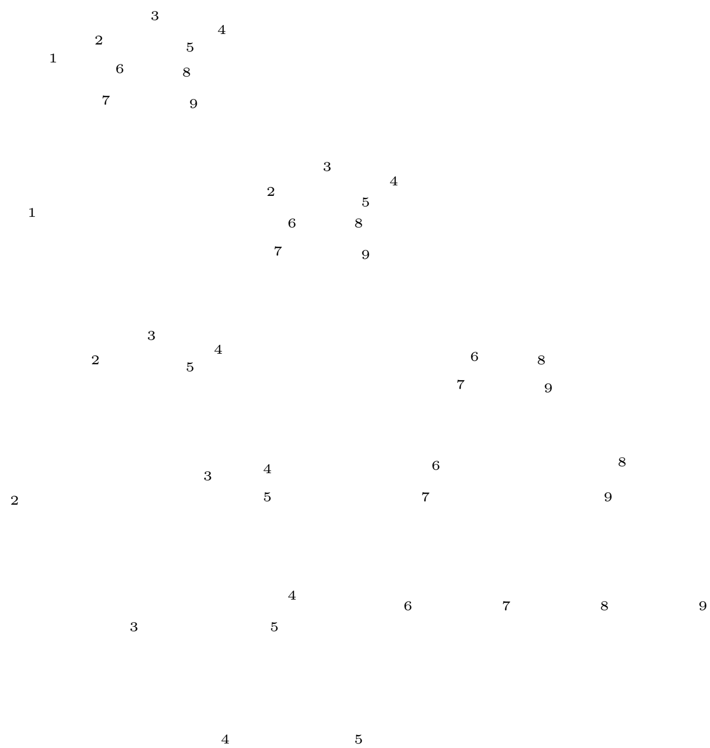


Figure 4.10 (De)construction of an ASP directed multigraph H_0 by two-terminal series and parallel (de)compositions.

It is easy to check that $L(H_0) = R_0$ for directed multigraphs H_0 and R_0 depicted in Figure 4.8. The following operations in a directed multigraph D are called **reductions:**

(a) **Series reduction:** Replace a path uvw, where $d^+_D(v) = d^-_D(v) = 1$ by the arc uw;
(b) **Parallel reduction:** Replace a pair of parallel arcs from u to v by just one arc from u to v.

The following proposition due to Duffin (see also the paper [726] by Valdes, Lawler and Tarjan) gives a characterization of ASP directed multigraphs. Its proof is left as Exercise 4.16.

Proposition 4.7.2 *[209] A directed multigraph is ASP if and only if it can be reduced to $\vec{P}_2$ by a sequence of series and parallel reductions.* □

The reader is advised to apply a sequence of series and parallel reductions to the directed multigraph H_0 of Figure 4.8 to obtain a digraph isomorphic to

$\vec{P}_2$. ¿From the algorithmic point of view, it is important that *every* sequence of series and parallel reductions transforms a directed multigraph to the same digraph. Indeed, this implies an obvious polynomial algorithm to verify if a given directed multigraph is ASP. The proof of the following result, due to Harary, Krarup and Schwenk, is left as Exercise 4.17.

Proposition 4.7.3 *[401] For every acyclic directed multigraph D, the result of application of series and parallel reductions until one can apply such reductions is a unique digraph H.* □

In [726], Valdes, Tarjan and Lawler showed how to construct a linear-time algorithm to recognize ASP directed multigraphs, which is based on Propositions 4.7.2 and 4.7.3. They also presented a more complicated linear-time algorithm to recognize VSP digraphs. Since we are limited in space, we will not discuss the details of the linear-time algorithms. Instead, we will consider the following simplified polynomial algorithm to recognize VSP digraphs.

VSP recognition algorithm:
Input: An acyclic digraph D.
Output: YES if D is VSP and NO, otherwise.

1. Compute the transitive reduction R of D.
2. Try to compute an acyclic directed multigraph H with $|I_D| = |F_D| = 1$ such that $L(H) = R$. If there is no such H, then output NO.
3. Verify whether H is an ASP directed multigraph. If it is so, then YES, otherwise, NO.

We prove first the correctness of this algorithm. If the output is YES, then, by Theorem 4.7.1, R is MVSP and thus D is VSP. If H is Step 2 is not found, then, by Theorem 4.7.1, R is not MVSP implying that D is not VSP. If H is not ASP, then R is not MVSP by the same theorem.

Now we prove that the algorithm is polynomial. Step 1 can be performed in polynomial time by Proposition 4.3.5. Step 2 can be implemented using Procedure Build-H described in the end of Section 4.5. This procedure implies that if there is an H such that $L(H) = R$, then there is such an H with additional property that $|I_D| = |F_D| = 1$. The procedure is polynomial. Finally, Step 3 is polynomial by the remark after Proposition 4.7.2.

4.8 Quasi-Transitive Digraphs

Quasi-transitive digraphs were introduced in Section 1.8. The aim of this section is to derive a recursive characterization of quasi-transitive digraphs which allows one to show that a number of problems for quasi-transitive digraphs including the longest path and cycle problems are polynomial time solvable (see Theorem 5.10.2). The characterization implies that every quasi-transitive digraph is totally Ψ-decomposable, where Ψ is the union of all transitive digraphs and all extended semicomplete digraphs. Our presentation is based on [79].

Proposition 4.8.1 *Let D be a quasi-transitive digraph. Suppose that $P = x_1x_2\dots x_k$ is a minimal (x_1, x_k)-path. Then the subdigraph induced by $V(P)$ is a semicomplete digraph and $x_j \rightarrow x_i$ for every $2 \leq i+1 < j \leq k$, unless $k = 4$, in which case the arc between x_1 and x_k may be absent.*

Proof: The cases $k = 2, 3, 4, 5$ are easily verified. As an example, let us consider the case $k = 5$. If x_i and x_j are adjacent and $2 \leq i+1 < j \leq 5$, then $x_j \rightarrow x_i$ since P is minimal. Since D is quasi-transitive, x_i and x_{i+2} are adjacent for $i = 1, 2, 3$. This and the minimality of P imply that $x_3 \rightarrow x_1, x_4 \rightarrow x_2$ and $x_5 \rightarrow x_3$. From these arcs and the minimality of P we conclude that $x_5 \rightarrow x_1$. Now the arcs x_4x_5 and x_5x_1 imply that $x_4 \rightarrow x_1$. Similarly, $x_5 \rightarrow x_1 \rightarrow x_2$ implies $x_5 \rightarrow x_2$.

The proof for the case $k \geq 6$ is by induction on k with the case $k = 5$ as the basis. By induction, each of $D\langle\{x_1, x_2, \dots, x_{k-1}\}\rangle$ and $D\langle\{x_2, x_3, \dots, x_k\}\rangle$ is a semicomplete digraph and $x_j \rightarrow x_i$ for any $1 < j - i \leq k - 2$. Hence x_3 dominates x_1 and x_k dominates x_3 and the minimality of P implies that x_k dominates x_1. □

Corollary 4.8.2 *If a quasi-transitive digraph D has an (x, y)-path but x does not dominate y, then either $y \rightarrow x$, or there exist vertices $u, v \in V(D) - \{x, y\}$ such that $x \rightarrow u \rightarrow v \rightarrow y$ and $y \rightarrow u \rightarrow v \rightarrow x$.*

Proof: This is easy to deduce by considering a minimal (x, y)-path and applying Proposition 4.8.1. □

Lemma 4.8.3 *Suppose that A and B are distinct strong components of a quasi-transitive digraph D with at least one arc from A to B. Then $A \mapsto B$.*

Proof: Suppose A and B are distinct strong components such that there exists an arc from A to B. Then for every choice of $x \in A$ and $y \in B$ there exists a path from x to y in D. Since A and B are distinct strong components, none of the alternatives in Corollary 4.8.2 can hold and hence $x \rightarrow y$. □

Lemma 4.8.4 *[79] Let D be a strong quasi-transitive digraph on at least two vertices. Then the following holds:*

(a) $\overline{UG(D)}$ is disconnected;
(b) If S and S' are two subdigraphs of D such that $\overline{UG(S)}$ and $\overline{UG(S')}$ are distinct connected components of $\overline{UG(D)}$, then either $S \mapsto S'$ or $S' \mapsto S$, or both $S \rightarrow S'$ and $S' \rightarrow S$ in which case $|V(S)| = |V(S')| = 1$.

Proof: The statement (b) can be easily verified from the definition of a quasi-transitive digraph and the fact that S and S' are completely adjacent in D (Exercise 4.18). We prove (a) by induction on $|V(D)|$. Statement (a) is trivially true when $|V(D)| = 2$ or 3. Assume that it holds when $|V(D)| < n$ where $n > 3$.

Suppose that there is a vertex z such that $D - z$ is not strong. Then there is an arc from (to) every terminal (initial) component of $D - z$ to (from) z. Since D is quasi-transitive, the last fact and Lemma 4.8.3 imply that $X \to Y$ for every initial (terminal) strong component X (Y) of $D - z$. Similar arguments show that each strong component of $D - z$ either dominates some terminal component or is dominated by some initial component of $D - z$ (intermediate strong components satisfy both). These facts imply that z is adjacent to every vertex in $D - z$. Therefore, $\overline{UG(D)}$ contains a component consisting of the vertex z, implying that $\overline{UG(D)}$ is disconnected and (a) follows.

Assume that there is a vertex v such that $D - v$ is strong. Since D is strong, D contains an arc vw from v to $D - v$. By induction, $\overline{UG(D - v)}$ is not connected. Let connected components S and S' of $\overline{UG(D - v)}$ be chosen such that $w \in S$, $S \mapsto S'$ in D (here we use (b) and the fact that $D - v$ is strong). Then v is completely adjacent to S' in D (as $v \to w$). Hence $\overline{UG(S')}$ is a connected component of $\overline{UG(D)}$ and the proof is complete. □

The following theorem completely characterizes quasi-transitive digraphs in recursive sense (see also Figure 4.11).

Theorem 4.8.5 (Bang-Jensen and Huang) *[79] Let D be a digraph which is quasi-transitive.*

(a) If D is not strong, then there exist a transitive oriented graph T with vertices $\{u_1, u_2, \dots, u_t\}$ and strong quasi-transitive digraphs $H_1, H_2, \dots, H_t$ such that $D = T[H_1, H_2, \dots, H_t]$, where H_i is substituted for u_i, $i = 1, 2, \dots, t$.

(b) If D is strong, then there exits a strong semicomplete digraph S with vertices $\{v_1, v_2, \dots, v_s\}$ and quasi-transitive digraphs $Q_1, Q_2, \dots, Q_s$ such that Q_i is either a vertex or is non-strong and $D = S[Q_1, Q_2, \dots, Q_s]$, where Q_i is substituted for v_i, $i = 1, 2, \dots, s$.

Proof: Suppose that D is not strong and let $H_1, H_2, \dots, H_t$ be the strong components of D. According to Lemma 4.8.3, if there is an arc between H_i and H_j, then either $H_i \mapsto H_j$ or $H_j \mapsto H_i$. Now if $H_i \mapsto H_j \mapsto H_k$ then, by quasi-transitivity, $H_i \mapsto H_k$. So by contracting each H_i to a vertex h_i, we get a transitive oriented graph T with vertices $h_1, h_2, \dots, h_t$. This shows that $D = T[H_1, H_2, \dots, H_t]$.

Suppose now that D is strong. Let $Q_1, Q_2, \dots, Q_s$ be the subdigraphs of D such that each $\overline{UG(Q_i)}$ is a connected component of $\overline{UG(D)}$. According to Lemma 4.8.4(a), each Q_i is either non-strong or just a single vertex. By

Figure 4.11 A decomposition of a non-strong quasi-transitive digraph. Big arcs between different boxed sets indicate that there is a complete domination in the direction shown.

Lemma 4.8.4(b) we obtain a strong semicomplete digraph S if each Q_i is contracted to a vertex. This shows that $D = S[Q_1, Q_2, \ldots, Q_s]$. $\square$

4.9 The Path-Merging Property and Path-Mergeable Digraphs

A digraph D is **path-mergeable**, if for any choice of vertices $x, y \in V(D)$ and any pair of internally disjoint (x, y)-paths P, Q, there exists an (x, y)-path R in D, such that $V(R) = V(P) \cup V(Q)$. We will see, in several places of this book, that the notion of a path-mergeable digraph is very useful for design of algorithms and proofs of theorems. This makes it worth while studying path-mergeable digraphs. The results presented in this section are adapted from [50], where the study of path-mergeable digraphs was initiated by Bang-Jensen.

We prove a characterization of path-mergeable digraphs, which implies that path-mergeable digraphs can be recognized efficiently.

Theorem 4.9.1 *A digraph D is path-mergeable if and only if for every pair of distinct vertices $x, y \in V(D)$ and every pair $P = xx_1 \ldots x_r y$, $P' = xy_1 \ldots y_s y$, $r, s \geq 1$ of internally disjoint (x, y)-paths in D, either there*

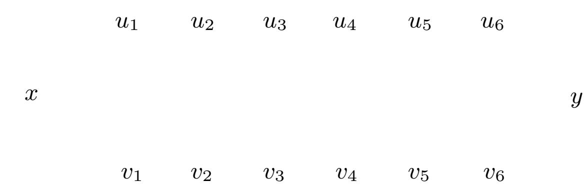


Figure 4.12 A digraph which is path-mergeable. The fat arcs indicate the path $xu_1u_2v_1v_2v_3u_3u_4u_5v_4v_5v_6u_6y$ from x to y which is obtained by merging the two (x, y)-paths $xu_1u_2u_3u_4u_5u_6y$ and $xv_1v_2v_3v_4v_5v_6y$.

exists an $i \in \{1, \ldots, r\}$, such that $x_i \rightarrow y_1$, or there exists a $j \in \{1, \ldots, s\}$, such that $y_j \rightarrow x_1$.

Proof: We prove 'only if' by induction on $r + s$. It is obvious for $r = s = 1$, so suppose that $r + s \geq 3$. If there is no arc between $\{x_1, \ldots, x_r\}$ and $\{y_1, \ldots, y_s\}$, then clearly P, P' cannot be merged into one path. Hence we may assume without loss of generality that there is an arc x_iy_j for some i, j, $1 \leq i \leq r, 1 \leq j \leq s$. If $j = 1$ then the claim follows. Otherwise apply induction to the paths $P[x, x_i]y_j$, $xP'[y_1, y_j]$.

The proof of 'if' is left to the reader. It is similar to the proof of Proposition 4.9.3 below. □

The proof of the following result is left as Exercise 4.23.

Corollary 4.9.2 *Path-mergeable digraphs can be recognized in polynomial time.* □

The next result shows that, if a digraph is path-mergeable, then the merging of paths can always be done in a particularly nice way.

Proposition 4.9.3 *Let D be a digraph which is path-mergeable and let $P = xx_1 \ldots x_ry$, $P' = xy_1 \ldots y_sy$, $r, s \geq 0$ be internally disjoint (x, y)-paths in D. The paths P and P' can be merged into one (x, y)-path P^* such that vertices from P (respectively, P') remain in the same order as on that path. Furthermore the merging can be done in at most $2(r + s)$ steps.*

Proof: We prove the result by induction on $r + s$. It is obvious if $r = 0$ or $s = 0$, so suppose that $r, s \geq 1$. By Theorem 4.9.1 there exists an i such that either $x_i \rightarrow y_1$ or $y_i \rightarrow x_1$. By scanning both paths forward one arc at a time, we can find i in at most $2i$ steps; suppose without loss of generality $x_i \rightarrow y_1$. By applying the induction hypothesis to the paths $P[x_i, x_r]y$ and $x_iP'[y_1, y_s]y$, we see that we can merge them into a single path Q in the required order-preserving way in at most $2(r + s - i)$ steps. The required path P^* is obtained by concatenating the paths $xP[x_1, x_i]$ and Q, and we have found it in at most $2(r+s)$ steps, as required. $\square$

4.10 Locally In-Semicomplete and Locally Out-Semicomplete Digraphs

A digraph D is **locally in-semicomplete** (**locally out-semicomplete**) if, for every vertex x of D, the in-neighbours (out-neighbours) of x induce a semicomplete digraph. Clearly, the converse of a locally in-semicomplete digraph is a locally out-semicomplete digraph and vice versa. A digraph D is **locally semicomplete** if it is both locally in- and locally out-semicomplete. See Figure 4.13. Clearly every semicomplete digraph is locally semicomplete. A locally in-semicomplete digraph with no 2-cycle is a **locally in-tournament digraph**. Similarly, one can define **locally out-tournament digraphs** and **locally tournament digraphs**. For convenience, we will sometimes refer to locally tournament digraphs as **local tournaments** and to locally in-tournament (out-tournament) digraphs as **local in-tournaments** (**local out-tournaments**).

(a) (b)

Figure 4.13 (a) A locally out-semicomplete digraph which is not locally in-semicomplete; (b) A locally semicomplete digraph.

Proposition 4.10.1 by Bang-Jensen shows that locally in-semicomplete and locally out-semicomplete digraphs form subclasses of the class of path-mergeable digraphs. In particular, this means that every tournament is path-mergeable. In many theorems and algorithms on tournaments this property is of essential use. In some other cases, the very use of this property allows one to simplify proofs of results on tournaments and their generalizations or speed up algorithms on those digraphs.

Proposition 4.10.1 *[50] Every locally in-semicomplete (out-semicomplete) digraph is path-mergeable.*

Proof: Let D be a locally out-semicomplete digraph and let $P = y_1y_2\dots y_k$, $Q = z_1z_2\dots z_t$ be a pair of internally disjoint (x, y)-paths (i.e., $y_1 = z_1 = x$

and $y_k = z_t = y$). We show that there exists an (x, y)-path R in D, such that $V(R) = V(P) \cup V(Q)$. Our claim is trivially true when $|A(P)| + |A(Q)| = 3$. Assume now that $|A(P)| + |A(Q)| \geq 4$. Since D is out-semicomplete, either $y_2 \rightarrow z_2$ or $z_2 \rightarrow y_2$ (or both) and the claim follows from Theorem 4.9.1.

The proposition holds for locally in-semicomplete digraphs as they are the converses of locally out-semicomplete digraphs. □

The path-mergeability can be generalized in a natural way as follows. A digraph D is **in-path-mergeable** if, for every vertex $y \in V(D)$ and every pair P, Q of internally disjoint paths with common terminal vertex y, there is a path R such that $V(R) = V(P) \cup V(Q)$, the path R terminates at y and starts at a vertex which is the initial vertex of either P or Q (or, possibly, both). Observe that, in this definition, the initial vertices of paths P and Q may coincide. Therefore, every in-path-mergeable digraph is path-mergeable. However, it is easy to see that not every path-mergeable digraph is in-path-mergeable (see Exercise 4.19). A digraph D is **out-path-mergeable** if the converse of D is in-path-mergeable. Clearly, every in-path-mergeable (out-path-mergeable) digraph is locally in-semicomplete (locally out-semicomplete). The converse is also true (hence this is another way of characterizing locally in-semicomplete digraphs). The proof of Proposition 4.10.2 is left as Exercise 4.20.

Proposition 4.10.2 *Every locally in-semicomplete (out-semicomplete, respectively) digraph is in-path-mergeable (out-path-mergeable, respectively).*

□

Some simple, yet very useful, properties of locally in-semicomplete digraphs are described in the following results (in [81], by Bang-Jensen, Huang and Prisner, these results were proved for locally tournament digraphs only, so the statements below are their slight generalizations first stated by Bang-Jensen and Gutin [65]). Observe that a locally out-semicomplete digraph, being the converse of a locally in-semicomplete digraph, has similar properties (see Exercise 4.26). The claim of Theorem 4.10.4 is illustrated in Figure 4.14.

Lemma 4.10.3 *Every connected locally in-semicomplete digraph D has an out-branching.*

Proof: By Proposition 1.6.1, it suffices to prove that D has only one initial strong component. Assume that D has a pair D_1, D_2 of initial strong components (i.e. no arc enters D_1 or D_2). Let $y_i \in V(D_i)$, $i = 1, 2$, and let $P = x_1 x_2 \ldots x_s$ be a shortest path between $V(D_1)$ and $V(D_2)$ in the underlying graph G of D. Since no arc enters D_1 or D_2, there is an index $k \leq s$ such that $x_1 x_2 \ldots x_{k-1}$ is a path in D, but $x_k \rightarrow x_{k-1}$. Since D is in-semicomplete, the vertices x_{k-2} and x_k are adjacent. However, this contradicts the fact that P is a shortest path between $V(D_1)$ and $V(D_2)$ in G. □

Theorem 4.10.4 *Let D be a locally in-semicomplete digraph.*

(i) Let A and B be distinct strong components of D. If a vertex $a \in A$ dominates some vertex in B, then $a \mapsto B$.
(ii) If D is connected, then $SC(D)$ has an out-branching.

Proof: Let A and B be strong components of D for which there is an arc (a, b) from A to B. Since B is strong, there is a (b', b)-path in B for every $b' \in V(B)$. By the definition of locally in-semicomplete digraphs and the fact that there is no arc from B to A, we can conclude that $a \rightarrow b'$. This proves (i).

Part (ii) follows from the fact that $SC(D)$ is itself a locally in-tournament digraph and Lemma 4.10.3. □

Figure 4.14 The strong decomposition of a non-strong locally in-semicomplete digraph. The big circles indicate strong components and a fat arc from a component A to a component B between two components indicates that there is at least one vertex $a \in A$ such that $a \mapsto B$.

4.11 Locally Semicomplete Digraphs

Locally semicomplete digraphs were introduced in 1990 by Bang-Jensen [44]. As shown in several places in our book, this class of digraphs has many nice properties in common with its proper subclass, semicomplete digraphs. The main aim of this section is to obtain a classification of locally semicomplete digraphs first proved by Bang-Jensen, Guo, Gutin and Volkmann [55]. In the process of deriving this classification, we will show several important properties of locally semicomplete digraphs. We start our consideration from round digraphs, a nice special class of locally semicomplete digraphs.

4.11.1 Round Digraphs

A digraph on n vertices is **round** if we can label its vertices $v_1, v_2, \ldots, v_n$ so that for each i, we have $N^+(v_i) = \{v_{i+1}, \ldots, v_{i+d^+(v_i)}\}$ and $N^-(v_i) = \{v_{i-d^-(v_i)}, \ldots, v_{i-1}\}$ (all subscripts are taken modulo n). We will refer to the ordering $v_1, v_2, \ldots, v_n$ as a **round labelling** of D. See Figure 4.15 for an example of a round digraph. Observe that every strong round digraph D is hamiltonian, since $v_1 v_2 \ldots v_n v_1$ form a hamiltonian cycle, whenever $v_1, v_2, \ldots, v_n$ is a round labelling. Round digraphs form a subclass of locally semicomplete digraphs. We will see below that round digraphs play an important role in the study of locally semicomplete digraphs.

2 3

1 4

6 5

R

Figure 4.15 A round digraph with a round labelling.

Proposition 4.11.1 *[438] Every round digraph is locally semicomplete.*

Proof: Let D be a round digraph and let $v_1, v_2, \ldots, v_n$ be a round labelling of D. Consider an arbitrary vertex, say v_i. Let x, y be a pair of out-neighbours of v_i. We show that x and y are adjacent. Assume without loss of generality that v_i, x, y appear in that circular order in the round labelling. Since $v_i \rightarrow y$ and the in-neighbours of y appear consecutively preceding y, we must have $x \rightarrow y$. Thus the out-neighbours of v_i are pairwise adjacent. Similarly, we can show that the in-neighbours of v_i are also pairwise adjacent. Therefore, D is locally semicomplete. $\square$

In the rest of this subsection, we will prove the following characterization of round digraphs due to Huang [438]. This characterization generalizes the corresponding characterizations of round local tournaments and tournaments, due Bang-Jensen [44] and Alspach and Tabib [22], respectively.

(a) (b) (c) (d)

Figure 4.16 Some forbidden digraphs in Huang's characterization

An arc xy of a digraph D is **ordinary** if yx is not in D. A cycle or path Q of a digraph D is **ordinary** if all arcs of Q are ordinary.

To prove Theorem 4.11.4 below, we need two lemmas due to Huang [438].

Lemma 4.11.2 *Let D be a round digraph then the following is true:*

(a) Every induced subdigraph of D is round.
(b) None of the digraphs in Figure 4.16 is an induced subdigraph of D.
(c) For each $x \in V(D)$, the subdigraphs induced by $N^+(x) - N^-(x)$ and $N^-(x) - N^+(x)$ are transitive tournaments.

Proof: Exercise 4.29. □

Lemma 4.11.3 *Let D be a round digraph. Then, for each vertex x of D, the subdigraph induced by $N^+(x) \cap N^-(x)$ contains no ordinary cycle.*

Proof: Suppose the subdigraph induced by some $N^+(x) \cap N^-(x)$ contains an ordinary cycle C. Let $v_1, v_2, \ldots, v_n$ be a round labelling of D. Without loss of generality, assume that $x = v_1$. Then C must contain an arc v_iv_j such that $v_jv_i \notin A(D)$ and $i > j$. We have $v_1 \in N^-(v_i)$ but $v_j \notin N^-(v_i)$, contradicting the assumption that $v_1, v_2, \ldots, v_n$ is a round labelling of D. □

Theorem 4.11.4 (Huang) *[438] A connected locally semicomplete digraph D is round if and only if the following holds for each vertex x of D:*

(a) $N^+(x) - N^-(x)$ and $N^-(x) - N^+(x)$ induce transitive tournaments and
(b) $N^+(x) \cap N^-(x)$ induces a (semicomplete) subdigraph containing no ordinary cycle.

Proof: The necessity follows from Lemmas 4.11.2(c) and 4.11.3. To prove the sufficiency, we consider two cases.

Case 1: *D* has an ordinary cycle. We start by proving that D contains an ordinary Hamilton cycle. Let $C = x_1x_2 \ldots x_kx_1$ be a longest ordinary cycle in D. Assume that $k \neq n$, the number of vertices in D. Since D is connected there is a vertex $v \in V(D) - V(C)$ such that v is adjacent to some vertex of C.

Suppose that there is an ordinary arc between v and some vertex, say x_1, of C. We may without loss of generality assume that the ordinary arc is x_1v (indeed, if necessary, we may consider the converse of D instead of D). The vertices v and x_2 are adjacent since they are out-neighbours of x_1. The arc between v and x_2 must be ordinary since D does not contain as induced subdigraph the digraph depicted in Figure 4.16 (a). Since C is a longest ordinary cycle, v cannot dominate x_2. Thus, $x_2 \mapsto v$. Similarly, we can prove that $x_i \mapsto v$ for every $i = 3, 4, \ldots, k$. Hence, $N^-(v) - N^+(v)$ contains all vertices of C, which contradicts the assumption that $N^-(v) - N^+(v)$ induces a transitive tournament.

Since there is no ordinary arc between v and C, we may assume that vx_1v is a 2-cycle of D. Using the fact that D is locally semicomplete, it is easy to derive that $V(C) \subseteq N^+(v) \cap N^-(v)$. This contradicts the assumption that $N^+(v) \cap N^-(v)$ contains no ordinary cycle.

Thus, we have shown that D contains an ordinary Hamilton cycle. This implies that $N^+(x) - N^-(x) \neq \emptyset$ for every $x \in V(D)$.

We apply the following algorithm to find a round labelling of D. Start with an arbitrary vertex, say y_1, and, for each $i = 1, 2, \ldots$, let y_{i+1} be the vertex of in-degree zero in the (transitive) tournament induced by $N^+(y_i) - N^-(y_i)$. Let $y_1, y_2, \ldots, y_r$ be distinct vertices produced by the algorithm such that the vertex w of in-degree zero in the tournament induced by $N^+(y_r) - N^-(y_r)$ is in $\{y_1, y_2, \ldots, y_{r-2}\}$.

We show that $w = y_1$. If $w = y_j$ with $j > 1$, then $\{y_{j-1}, y_r\} \mapsto y_j$. Thus, y_{j-1} and y_r are adjacent by an ordinary arc (since the digraph in Figure 4.16(b) is forbidden). But either $y_{j-1} \mapsto y_r$ or $y_r \mapsto y_{j-1}$ contradicts the fact that y_j is the vertex of in-degree zero in the tournament induced by $N^+(y_{j-1}) - N^-(y_{j-1})$ or $N^+(y_r) - N^-(y_r)$. Thus, $w = y_1$ and $C' = y_1y_2 \ldots y_ry_1$ is an ordinary cycle.

We next show that $r = n$. Suppose $r < n$. Then, there is a vertex u, which is not in C' and is adjacent to some y_i of C'. Suppose first that $u \in N^+(y_i) - N^-(y_i)$. Then, being out-neighbours of y_i, the vertices y_{i+1} and u are adjacent. Since D contains no induced subdigraph isomorphic to the digraph in Figure 4.16 (a) and y_{i+1} is the vertex of in-degree zero in the subdigraph induced by $N^+(y_i) - N^-(y_i)$, we have $u \in N^+(y_{i+1}) - N^-(y_{i+1})$. This implies that u and y_{i+2} are adjacent. Similarly, we must have $u \in N^+(y_{i+2}) - N^-(y_{i+2})$. Continuing this way, we see that $u \in N^+(y_k) - N^-(y_k)$ for every $k = 1, 2, \ldots, r$. Hence, C' is contained in the subdigraph induced by $N^-(u) - N^+(u)$, a contradiction.

A similar argument applies for the case $u \in N^-(y_i) - N^+(y_i)$. So, we may assume that $u \in N^+(y_i) \cap N^-(y_i)$ and there is no ordinary arc between u and C'. Using the fact that D is locally semicomplete, it is easy to see that C' is contained in the subdigraph induced by $N^+(u) \cap N^-(u)$, a contradiction. Thus, $r = n$, i.e., the algorithm labels all vertices of D. To complete Case 1, it suffices to prove that $y_1, y_2, \ldots, y_n$ is a round labelling. Suppose not. Then, there are three vertices y_a, y_b, y_c listed in the circular order in the labelling such that, without loss of generality, we have

$$y_a \rightarrow y_c \text{ and } y_a \not\rightarrow y_b.$$

Assume that the tree vertices were chosen such that the number of vertices from y_b to y_c in the circular order is as small as possible. This implies that $c = b+1$. Since y_a and y_b are both in-neighbours of y_c, they are adjacent. Thus, $y_b \mapsto y_a$. Since we also have $y_b \mapsto y_c$ (recall that $y_c \in N^+(y_b) - N^-(y_b)$ by the definition of the labelling) and D contains no induced subdigraph isomorphic to the digraph given in Figure 4.16 (a), $y_a \mapsto y_c$. So, y_c is not the vertex of in-degree zero in the tournament induced by $N^+(y_b) - N^-(y_b)$, contradicting the choice of y_c.

Case 2: *D* contains no ordinary cycle. If D has no ordinary arc, D is complete. Thus, any labelling of $V(D)$ is round. So assume that D has an ordinary arc. Since D has an ordinary arc, but has no ordinary cycle, we claim that there is a vertex z_1 with

$$N^-(z_1) - N^+(z_1) = \emptyset \text{ and } N^+(z_1) - N^-(z_1) \neq \emptyset.$$

Indeed, let w_2w_1 be an ordinary arc in D. We may set $z_1 = w_2$ unless $N^-(w_2) - N^+(w_2) \neq \emptyset$. In the last case there is an ordinary arc whose head is w_2. Let w_3w_2 be such an arc. Again, either we may set $z_1 = w_3$ or there is an ordinary arc w_4w_3. Since D is finite and contains no ordinary cycle, the above process cannot repeat vertices and hence terminates at some vertex w_j such that we may set $z_1 = w_j$.

We apply the following algorithm to find a path in D. Begin with z_1 and, for each $i = 1, 2, \ldots,$ let z_{i+1} be the vertex of in-degree zero in the (transitive) tournament induced by $N^+(z_i) - N^-(z_i)$ unless this set is empty. Since D has no ordinary cycle, this produces a path $P = z_1z_2 \ldots z_s$ with $N^+(z_s) - N^-(z_s) = \emptyset$. Applying an argument similar to that used above, we can show that $z_1, z_2, \ldots, z_s$ is a round labelling of the subdigraph induced by $V(P)$. Thus, if P contains all vertices of D, then a round labelling of D is established. So assume that there is a vertex v not in P, which is adjacent to some vertex of P. It is easy to see that there is no ordinary arc between v and P. This implies that $v \in N^+(z_i) \cap N^-(z_i)$ for each $i = 1, 2, \ldots, s$. In fact, it is not hard to see that the same is true for every vertex $v \in V(D) - V(P)$. Therefore, if we apply the above algorithm starting from an appropriate ('z_1-type') vertex not in P, we obtain a new ordinary path Q and $V(Q) \cap V(P) = \emptyset$. By applying the above algorithm as many times as possible, we obtain a collection of vertex-disjoint ordinary paths $P^k = z_1^k z_2^k \ldots z_{m_k}^k$, $k = 1, 2, \ldots, t$. Let $z_1^{t+1}, \ldots, z_{m_{t+1}}^{t+1}$ be the remaining vertices (these form a complete digraph). It is easy to verify that labelling the vertices according to the ordering

$$z_1^1, z_2^1, \ldots, z_{m_1}^1, z_1^2, z_2^2, \ldots, z_{m_2}^2, \ldots, z_1^{t+1}, z_2^{t+1}, \ldots, z_{m_{t+1}}^{t+1}$$

results in a round labelling of D. In fact the proof above implies that if we let D_i, $i = 1, 2, \ldots, t+1$, be the subdigraph induced by the vertices

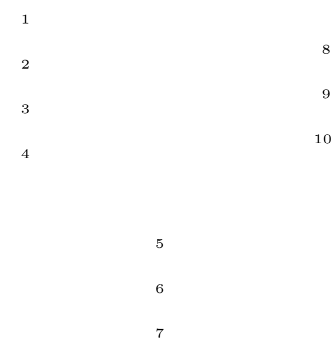


Figure 4.17 An example of a round digraph containing 2-cycles. Undirected edges are used to indicate 2-cycles and fat edges between two boxes indicate a complete connection in both directions between the corresponding vertices.

with superscript i above, then we have $D = \overleftrightarrow{K}_{t+1} [D_1, D_2, \ldots, D_t, D_{t+1}]$ (see Figure 4.17). $\square$

It is left as an exercise to show that this proof implies a polynomial algorithm to decide whether a digraph D is round and to find a round labelling of D if D is round.

Corollary 4.11.5 (Bang-Jensen) *[44] A connected local tournament D is round if and only if, for each vertex x of D, $N^+(x)$ and $N^-(x)$ induce transitive tournaments.* $\square$

4.11.2 Non-Strong Locally Semicomplete Digraphs

The most basic properties of strong components of a connected non-strong locally semicomplete digraph are given in the following result, due to Bang-Jensen.

Theorem 4.11.6 *[44] Let D be a connected locally semicomplete digraph that is not strong. Then the following holds for D.*

(a) If A and B are distinct strong components of D with at least one arc between them, then either $A \mapsto B$ or $B \mapsto A$.
(b) If A and B are strong components of D, such that $A \mapsto B$, then A and B are semicomplete digraphs.
(c) The strong components of D can be ordered in a unique way $D_1, D_2, \ldots, D_p$ such that there are no arcs from D_j to D_i for $j > i$, and D_i dominates D_{i+1} for $i = 1, 2, \ldots, p-1$.

Proof: Recall that a locally semicomplete digraph is a locally in-semicomplete digraph as well as a locally out-semicomplete digraph. Part (a) of this theorem follows immediately from Part (i) of Theorem 4.10.4 and its analogue for locally out-semicomplete digraphs. Part (b) can be easily obtained from the definition of a locally semicomplete digraph. Finally, Part (c) follows from the fact proved in Theorem 4.10.4 (and its analogue for locally out-semicomplete digraphs) that $SC(D)$ has an out-branching and an in-branching. Indeed, a digraph which is both out-branching and in-branching is merely a hamiltonian path. □

A locally semicomplete digraph D is **round decomposable** if there exists a round local tournament R on $r \geq 2$ vertices such that $D = R[S_1, \ldots, S_r]$, where each S_i is a strong semicomplete digraph. We call $R[S_1, \ldots, S_r]$ a **round decomposition** of D. The following consequence of Theorem 4.11.6, whose proof is left as Exercise 4.30, shows that connected, but not strongly connected locally semicomplete digraphs are round decomposable.

Figure 4.18 A round decomposable locally semicomplete digraph D. The big circles indicate the sets that correspond to the sets $W_1, W_2, \ldots, W_6$ in the decomposition $D = R[W_1, W_2, \ldots, W_6]$, where R is the round locally semicomplete digraph one obtains by replacing each circled set by one vertex. Fat arcs indicate that there is a complete domination in the direction shown.

Corollary 4.11.7 *[44] Every connected, but not strongly connected locally semicomplete digraph D has a unique round decomposition $R[D_1, D_2, \ldots, D_p]$, where $D_1, D_2, \ldots, D_p$ is the acyclic ordering of strong components of D and R is the round local tournament containing no cycle which one obtains by taking one vertex from each D_i.* □

Now we describe another kind of decomposition theorem for locally semicomplete digraphs due to Guo and Volkmann. The proof of this theorem is left as Exercise 4.31. The statement of the theorem is illustrated in Figure 4.19.

Theorem 4.11.8 *[349, 351] Let D be a connected locally semicomplete digraph that is not strong and let $D_1, \ldots, D_p$ be the acyclic ordering of strong components of D. Then D can be decomposed into $r \geq 2$ induced subdigraphs $D'_1, D'_2, \ldots, D'_r$ as follows:*

$$D'_1 = D_p, \quad \lambda_1 = p,$$

$$\lambda_{i+1} = \min\{\, j \mid N^+(D_j) \cap V(D'_i) \neq \emptyset\},$$

$$\textit{and} \quad D'_{i+1} = D\langle V(D_{\lambda_{i+1}}) \cup V(D_{\lambda_{i+1}+1}) \cup \cdots \cup V(D_{\lambda_i - 1})\rangle.$$

The subdigraphs $D'_1, D'_2, \ldots, D'_r$ satisfy the properties below:

(a) D'_i consists of some strong components of D and is semicomplete for $i = 1, 2, \ldots, r$
(b) D'_{i+1} dominates the initial component of D'_i and there exists no arc from D'_i to D'_{i+1} for $i = 1, 2, \ldots, r-1$
(c) if $r \geq 3$, then there is no arc between D'_i and D'_j for i, j satisfying $|j-i| \geq 2$. □

For a connected, but not strongly connected locally semicomplete digraph D, the unique sequence $D'_1, D'_2, \ldots, D'_r$ defined in Theorem 4.11.8 is called the **semicomplete decomposition** of D.

4.11.3 Strong Round Decomposable Locally Semicomplete Digraphs

In the previous subsection we saw that every connected non-strong locally semicomplete digraph is round decomposable. This property does not hold for strong locally semicomplete digraphs (see Lemma 4.11.14). The following assertions, due to Bang-Jensen, Guo, Gutin and Volkman, provide some important properties concerning round decompositions of strong locally semicomplete digraphs.

Proposition 4.11.9 *[55] Let $R[H_1, H_2, \ldots, H_\alpha]$ be a round decomposition of a strong locally semicomplete digraph D. Then, for every minimal separating set S, there are two integers i and $k \geq 0$ such that $S = V(H_i) \cup \ldots \cup V(H_{i+k})$.*

Proof: We will first prove that

$$\text{if } V(H_i) \cap S \neq \emptyset, \text{ then } V(H_i) \subseteq S. \tag{4.6}$$

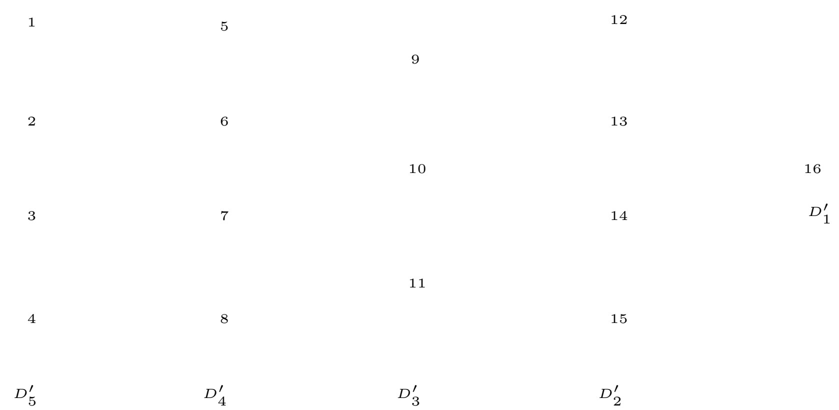


Figure 4.19 The semicomplete decomposition of a non-strong locally semicomplete digraph with 16 strong components (numbered 1-16 corresponding to the acyclic ordering). Each circle indicates a strong component and each box indicates a semicomplete subdigraph formed by consecutive components all of which dominate the first component in the previous layer. For clarity arcs inside components as well as some arcs between components inside a semicomplete subdigraph D'_i (all going from top to bottom) are omitted.

Assume that there exists H_i such that $V(H_i) \cap S \neq \emptyset \neq V(H_i) - S$. Using this assumption we shall prove that $D - S$ is strong, contradicting the definition of S.

Let $s' \in V(H_i) \cap S$. To show that $D - S$ is strong, we consider a pair of different vertices x and y of $D - S$ and prove that $D - S$ has an (x, y)-path. Since S is a minimal separating set, $D' = D - (S - s')$ is strong. Consider a shortest (x, y)-path P in D' among all (x, y)-paths using at most two vertices from each H_j. The existence of such a path follows from the fact that R is strong. Since the vertices of H_i in D' have the same in- and out-neighbourhoods, P contains at most one vertex from H_i, unless $x, y \in V(H_i)$ in which case P contains only these two vertices from H_i. If s' is not on P, we are done. Thus, assume that s' is on P. Then, since P is shortest possible, neither x nor y belongs to H_i. Now we can replace s' with a vertex in $V(H_i) - S$. Therefore, $D - S$ has an (x, y)-path, so (4.6) is proved.

Suppose that S consists of disjoint sets $T_1, \ldots, T_\ell$ such that

$$T_i = V(H_{j_i}) \cup \ldots \cup V(H_{j_i + k_i}) \quad \text{and} \quad (V(H_{j_i - 1}) \cup V(H_{j_i + k_i + 1})) \cap S = \emptyset$$

for $i \in \{1, \ldots, \ell\}$. If $\ell \geq 2$, then $D - T_i$ is strong and hence it follows from the fact that R is round that $H_{j_i - 1}$ dominates $H_{j_i + k_i + 1}$ for every $i = 1, \ldots, \ell$. Therefore, $D - S$ is strong; a contradiction. □

Corollary 4.11.10 *[55] If a locally semicomplete digraph D is round decomposable, then it has a unique round decomposition $D = R[D_1, D_2, \ldots, D_\alpha]$.*

Proof: Suppose that D has two different round decompositions: $D = R[D_1, \ldots, D_\alpha]$ and $D = R'[H_1, \ldots, H_\beta]$.

By Corollary 4.11.7, we may assume that D is strong. By the definition of a round decomposition, this implies that $\alpha, \beta \geq 3$. Let S be a minimal separating set of D. By Proposition 4.11.9, we may assume without loss of generality that $S = V(D_1 \cup \ldots \cup D_i) = V(H_1 \cup \ldots \cup H_j)$ for some i and j. Since $D - S$ is non-strong, by Corollary 4.11.7, $D_{i+1} = H_{j+1}, \ldots, D_\alpha = H_\beta$ (in particular, $\alpha - i = \beta - j$). Now it suffices to prove that

$$D_1 = H_1, \ldots, D_i = H_j \text{ (in particular, } i = j). \tag{4.7}$$

If $D\langle S \rangle$ is non-strong, then (4.7) follows by Corollary 4.11.7. If $D\langle S \rangle$ is strong, then first consider the case $\alpha = 3$. Then $S = V(D_1)$, because $D - S$ is non-strong and $\alpha = 3$. Assuming that $j > 1$, we obtain that the subdigraph of D induced by S has a strong round decomposition. This contradicts the fact that R' is a local tournament, since the in-neighbourhood of the vertex r'_{j+1} in R' contains a cycle (where r'_p corresponds to H_p, $p = 1, \ldots, \beta$). Therefore, (4.7) is true for $\alpha = 3$. If $\alpha > 3$, then we can find a separating set in $D\langle S \rangle$ and conclude by induction that (4.7) holds. □

Proposition 4.11.9 allows us to construct a polynomial algorithm for checking whether a locally semicomplete digraph is round decomposable.

Proposition 4.11.11 *[55] There exists a polynomial algorithm to decide whether a given locally semicomplete digraph D has a round decomposition and to find this decomposition if it exists.*

Proof: We only give a sketch of such an algorithm. Find a minimal separating set S in D starting with $S' = N^+(x)$ for a vertex $x \in V(D)$ and deleting vertices from S' until a minimal separating set is obtained. Construct the strong components of $D\langle S \rangle$ and $D - S$ and label these $D_1, D_2, \ldots, D_\alpha$, where $D_1, \ldots, D_p$, $p \geq 1$, form an acyclic ordering of the strong components of $D\langle S \rangle$ and $D_{p+1}, \ldots, D_\alpha$ form an acyclic ordering of the strong components of $D - S$. For every pair D_i and D_j $(1 \leq i \neq j \leq \alpha)$, we check the following: if there exist some arcs between D_i and D_j, then either $D_i \mapsto D_j$ or $D_j \mapsto D_i$. If we find a pair for which the above condition is false, then D is not round decomposable. Otherwise, we form a digraph $R = D\langle \{x_1, x_2, \ldots, x_\alpha\} \rangle$, where $x_i \in V(D_i)$ for $i = 1, 2, \ldots, \alpha$. We check whether R is round using Corollary 4.11.5. If R is not round, then D is not round decomposable. Otherwise, D is round decomposable and $D = R[D_1, \ldots, D_\alpha]$.

It is not difficult to verify that our algorithm is correct and polynomial. □

4.11.4 Classification of Locally Semicomplete Digraphs

We start this subsection with a lemma on minimal separating sets of locally semicomplete digraphs. It will be shown in Lemma 7.13.4 that for a strong locally semicomplete digraph D and a minimal separating set S in D, we have that $D-S$ is connected.

Lemma 4.11.12 *[55] If a strong locally semicomplete digraph D is not semicomplete, then there exists a minimal separating set $S \subset V(D)$ such that $D-S$ is not semicomplete. Furthermore, if $D_1, D_2, \ldots, D_p$ is the acyclic ordering of the strong components of D and $D'_1, D'_2, \ldots, D'_r$ is the semicomplete decomposition of $D-S$, then $r \geq 3$, $D\langle S \rangle$ is semicomplete and we have $D_p \mapsto S \mapsto D_1$.*

Proof: Suppose $D-S$ is semicomplete for every minimal separating set S. Then $D-S$ is semicomplete for all separating sets S. Hence D is semicomplete, because any pair of non-adjacent vertices can be separated by some separating set S. This proves the first claim of the lemma.

Let S be a minimal separating set such that $D-S$ is not semicomplete. Clearly, if $r=2$ (in Theorem 4.11.8), then $D-S$ is semicomplete. Thus, $r \geq 3$. By the minimality of S every vertex $s \in S$ dominates a vertex in D_1 and is dominated by a vertex in D_p. Thus if some $x \in D_p$ was dominated by $s \in S$, then, by the definition of a locally semicomplete digraph, we would have $D_1 \mapsto D_p$, contradicting the fact that $r \geq 3$. Hence (using that D_p is strongly connected) we get that $D_p \mapsto S$ and similarly $S \mapsto D_1$. From the last observation it follows that S is semicomplete. □

Now we consider strongly connected locally semicomplete digraphs which are not semicomplete and not round decomposable. We first show that the semicomplete decomposition of $D-S$ has exactly three components, whenever S is a minimal separating set such that $D-S$ is not semicomplete.

Lemma 4.11.13 *[55] Let D be a strong locally semicomplete digraph which is not semicomplete. Either D is round decomposable, or D has a minimal separating set S such that the semicomplete decomposition of $D-S$ has exactly three components D'_1, D'_2, D'_3.*

Proof: By Lemma 4.11.12, D has a minimal separating set S such that the semicomplete decomposition of $D-S$ has at least three components.

Assume now that the semicomplete decomposition of $D-S$ has more than three components $D'_1, \ldots, D'_r$ ($r \geq 4$). Let $D_1, D_2, \ldots, D_p$ be the acyclic ordering of strong components of $D-S$. According to Theorem 4.11.8 (c), there is no arc between D'_i and D'_j if $|i-j| \geq 2$. It follows from the definition of a locally semicomplete digraph that

$$N^+(D'_i) \cap S = \emptyset \text{ for } i \geq 3 \text{ and } N^-(D'_j) \cap S = \emptyset \text{ for } j \leq r-2. \qquad (4.8)$$

By Lemma 4.11.12, $D\langle S \rangle$ is semicomplete and $S = N^+(D_p)$. Let $D_{p+1}, \ldots, D_{p+q}$ be the acyclic ordering of the strong components of $D\langle S \rangle$. Using (4.8) and the assumption $r \geq 4$, it is easy to check that if there is an arc between D_i and D_j $(1 \leq i \neq j \leq p+q)$, then $D_i \mapsto D_j$ or $D_j \mapsto D_i$. Let $R = D\langle\{x_1, x_2, \ldots, x_{p+q}\}\rangle$ with $x_i \in V(D_i)$ for $i = 1, 2, \ldots, p+q$. Now it suffices to prove that R is a round local tournament.

Since R is a subdigraph of D and no pair D_i, D_j induces a strong digraph, we see that R is a local tournament. By Corollary 4.11.7 each of the subdigraphs $R' = R - \{x_{p+1}, \ldots, x_{p+q}\}$, $R'' = R - V(R) \cap V(D'_{r-1})$ and $R''' = R - V(R) \cap V(D'_2)$ is round. Since $N^+(v) \cap V(R)$ (as well as $N^-(v) \cap V(R)$) is completely contained in one of the sets $V(R'), V(R'')$ and $V(R''')$ for every $v \in V(R)$, we see that R is round.

Thus if $r \geq 4$, then D is round decomposable. $\square$

Our next result is a characterization of locally semicomplete digraphs which are not semicomplete and not round decomposable. This characterization was proved for the first time by Guo in [341]. A weaker form was obtained earlier by Bang-Jensen in [49]. Here we give the proof of this result from [55].

Lemma 4.11.14 *Let D be a strong locally semicomplete digraph which is not semicomplete. Then D is not round decomposable if and only if the following conditions are satisfied:*

(a) There is a minimal separating set S such that $D - S$ is not semicomplete and for each such S, $D\langle S\rangle$ is semicomplete and the semicomplete decomposition of $D - S$ has exactly three components D'_1, D'_2, D'_3;
(b) There are integers α, β, μ, ν with $\lambda_2 \leq \alpha \leq \beta \leq p-1$ and $p+1 \leq \mu \leq \nu \leq p+q$ such that

$$N^-(D_\alpha) \cap V(D_\mu) \neq \emptyset \quad \textit{and} \quad N^+(D_\alpha) \cap V(D_\nu) \neq \emptyset,$$

$$\text{or} \quad N^-(D_\mu) \cap V(D_\alpha) \neq \emptyset \quad \textit{and} \quad N^+(D_\mu) \cap V(D_\beta) \neq \emptyset,$$

where $D_1, D_2, \ldots, D_p$ and $D_{p+1}, \ldots, D_{p+q}$ are the acyclic orderings the strong components of $D - S$ and $D\langle S\rangle$, respectively, and D_{λ_2} is the initial component of D'_2.

Proof: If D is round decomposable and satisfies (a), then we must have $D = R[D_1, D_2, \ldots, D_{p+q}]$, where R is the digraph obtained from D by contracting each D_i into one vertex. This follows from Corollary 4.11.7 and the fact that each of the digraphs $D - S$ and $D - V(D'_2)$ has a round decomposition that agrees with this structure. Now it is easy to see that D does not satisfy (b).

Suppose now that D is not round decomposable. By Lemmas 4.11.12 and 4.11.13, D satisfies (a), so we only have to prove that it also satisfies (b).

If there are no arcs from S to D'_2, then it is easy to see that D has a round decomposition. If there exist components D_{p+i} and D_j with $V(D_j) \subseteq V(D'_2)$, such that there are arcs in both directions between D_{p+i} and D_j, then D satisfies (b). So we can assume that for every pair of sets from the collection $D_1, D_2, \dots, D_{p+q}$, either there are no arcs between these sets, or one set completely dominates the other. Then, by Corollary 4.11.5, D is round decomposable, with round decomposition $D = R[D_1, D_2, \dots, D_{p+q}]$ as above, unless we have three subdigraphs $X, Y, Z \in \{D_1, D_2, \dots, D_{p+q}\}$ such that $X \mapsto Y \mapsto Z \mapsto X$ and there exists a subdigraph $W \in \{D_1, D_2, \dots, D_{p+q}\} - \{X, Y, Z\}$ such that either $W \mapsto X, Y, Z$ or $X, Y, Z \mapsto W$.

One of the subdigraphs X, Y, Z, say without loss of generality X, is a strong component of $D\langle S\rangle$. If we have $V(Y) \subseteq S$ also, then $V(Z) \subseteq V(D'_2)$ and W is either in $D\langle S\rangle$ or in D'_2 (there are four possible positions for W satisfying that either $W \mapsto X, Y, Z$ or $X, Y, Z \mapsto W$). In each of these cases it is easy to see that D satisfies (b). For example, if W is in $D\langle S\rangle$ and $W \mapsto X, Y, Z$, then any arc from W to Z and from Z to X satisfies the first part of (b). The proof is similar when $V(Y) \subseteq V(D'_3)$. Hence we can assume that $V(Y) \subseteq V(D'_2)$. If $Z = D_p$, then W must be either in $D\langle S\rangle$ and $X, Y, Z \mapsto W$, or $V(W) \subseteq V(D'_2)$ and $W \mapsto X, Y, Z$ (which means that $W = D_i$ and $Y = D_j$ for some $\lambda_2 \leq i < j < p$). In both cases it is easy to see that D satisfies (b). The last case $V(Y), V(Z) \subseteq V(D'_2)$ can be treated similarly. □

We can now state a classification of locally semicomplete digraphs.

Theorem 4.11.15 (Bang-Jensen, Guo, Gutin, Volkmann) *[55] Let D be a connected locally semicomplete digraph. Then exactly one of the following possibilities holds.*

(a) D is round decomposable with a unique round decomposition given by $D = R[D_1, D_2, \dots, D_\alpha]$, where R is a round local tournament on $\alpha \geq 2$ vertices and D_i is a strong semicomplete digraph for $i = 1, 2, \dots, \alpha$;
(b) D is not round decomposable and not semicomplete and it has the structure as described in Lemma 4.11.14;
(c) D is a semicomplete digraph which is not round decomposable. □

We finish this section with the following useful proposition, whose proof is left as Exercise 4.35.

Proposition 4.11.16 *[55] Let D be a strong non-round decomposable locally semicomplete digraph and let S be a minimal separating set of D such that $D - S$ is not semicomplete. Let $D_1, \dots, D_p$ be the acyclic ordering of the strong components of $D - S$ and $D_{p+1}, \dots, D_{p+q}$ be the acyclic ordering of the strong components of $D\langle S\rangle$. Suppose that there is an arc $s \to v$ from S to D'_2 with $s \in V(D_i)$ and $v \in V(D_j)$, then*

$$D_i \cup D_{i+1} \cup \dots \cup D_{p+q} \mapsto D'_3 \mapsto D_{\lambda_2} \cup \dots \cup D_j.$$

□

4.12 Totally Φ_i-Decomposable Digraphs

Theorem 4.8.5 is a very important starting point for construction of polynomial algorithms for hamiltonian paths and cycles in quasi-transitive digraphs (see Chapter 5) and solving more general problems in this class of digraphs. This theorem shows that quasi-transitive digraphs are totally Φ-decomposable, where Φ is the union of extended semicomplete and transitive digraphs. Since both extended semicomplete digraphs and transitive digraphs are special subclasses of much wider classes of digraphs, it is natural to study totally Φ-decomposable digraphs, where Φ is a much more general class of digraphs than the union of extended semicomplete and transitive digraphs. However, our choice of candidates for the class Φ should be restricted in such a way that we can still construct polynomial algorithms for some important problems such as the hamiltonian cycle problem using properties of digraphs in Φ.

This idea was first used by Bang-Jensen and Gutin [62] to introduce the following three classes of digraphs:

(a) Φ_0 is the union of all semicomplete multipartite digraphs, all connected extended locally semicomplete digraphs and all acyclic digraphs,
(b) Φ_1 is the union of all semicomplete bipartite digraphs, all connected extended locally semicomplete digraphs and all acyclic digraphs, and
(c) Φ_2 is the union of all connected extended locally semicomplete digraphs and all acyclic digraphs.

The aim of this section is to show that totally Φ_i-decomposable digraphs can be recognized in polynomial time for $i = 0, 1, 2$. (If these recognition problems were not polynomial, then the study of the properties of totally Φ_i-decomposable digraphs would be of much less interest.)

A set Φ of digraphs is **hereditary** if $D \in \Phi$ implies that every induced subdigraph of D is in Φ. Observe that every Φ_i, $i = 0, 1, 2$ is a hereditary set.

Lemma 4.12.1 *Let Φ be a hereditary set of digraphs. If a given digraph D is totally Φ-decomposable, then every induced subdigraph D' of D is totally Φ-decomposable. In other words, total Φ-decomposability is a hereditary property.*

Proof: By induction on the number of vertices of D. The claim is obviously true if D has less than 3 vertices.

If $D \in \Phi$, then our claim follows from the fact that Φ is hereditary. So we may assume that $D = R[H_1, \ldots, H_r]$, $r \geq 2$, where $R \in \Phi$ and each of $H_1, \ldots, H_r$ is totally Φ-decomposable.

Let D' be an induced subdigraph of D. If there is an index i so that $V(D') \subset V(H_i)$, then D' is totally Φ-decomposable by induction. Otherwise, $D' = R'[T_1, \ldots, T_{r'}]$, where $r' \geq 2$ and $R' \in \Phi$, is the subdigraph of R induced by those vertices i of R, whose H_i has a non-empty intersection with

$V(D')$ and the T_j's are the corresponding H_i's restricted to the vertices of D'. Observe that $R' \in \Phi$, since Φ is hereditary. Moreover, by induction, each T_j is totally Φ-decomposable, hence so is D'. □

Lemma 4.12.2 *There exists an $O(mn+n^2)$-algorithm for checking if a digraph D with n vertices and m arcs has a decomposition $D = R[H_1, \ldots, H_r]$, $r \geq 2$, where H_i is an arbitrary digraph and the digraph R is either acyclic or semicomplete multipartite or semicomplete bipartite or connected extended locally semicomplete.*

Proof: If D is not connected and $D_1,\ldots,D_c$ are its components, then $D = \overline{K}_c[D_1, \ldots, D_c]$. Hence, in the rest of the proof we may assume that D is connected. We consider the different possibilities for R we are interested in, one by one.

Check whether *R* can be acyclic: First find the strong components $D_1, \ldots, D_k$ of D. If $k = 1$ then R cannot be acyclic and we can stop verifying that possibility. So suppose $k \geq 2$.

If we find two strong components D_i and D_j such that there is an arc between them but there are non-adjacent vertices $x \in D_i$ and $y \in D_j$, then we replace D_i and D_j by their union. This is justified because D_i and D_j cannot be in different sets H_s and H_t in a possible decomposition. Repeat this step but now check also the possibility for a pair D' and D'' of new 'components' to have arcs between D' and D'' in different directions. In the last case we also replace D' and D'' by their union. Continue this procedure until all remaining sets satisfy that either there is no arc between them, or there are all possible arcs from one to the other. Let $V_1, \ldots, V_r$, $r \geq 1$ denote the distinct vertex sets of the obtained 'components'. If $r = 1$, then we cannot find an acyclic graph as R. Otherwise $D = R[V_1, \ldots, V_r]$, $r \geq 2$, and we obtain R by taking one vertex from each V_i.

Check whether *R* can be a semicomplete multipartite digraph: Find the connected components $\overline{G}_1, \ldots, \overline{G}_c$, $c \geq 1$, of the complement of the underlying graph $UG(D)$ of D. If $c = 1$, then R cannot be semicomplete multipartite. So we may assume that $c \geq 2$ below. Let G_j be the subgraph of $UG(D)$ induced by the vertices V_j of the jth component $\overline{G}_j$ of the complement of $UG(D)$. Furthermore, let $G_{j1}, \ldots, G_{jn_j}$, $n_j \geq 1$, be the connected components of G_j. Denote $V_{jk} = V(G_{jk})$.

Starting with the collection $W = \{V_1, \ldots, V_c\}$, we identify two of the sets V_i and V_j if there exist V_{ia} and V_{jb} $a \in \{1, \ldots, n_i\}$, $b \in \{1, \ldots, n_j\}$ such that we have none of the possibilities $V_{ia} \mapsto V_{jb}$, $V_{jb} \mapsto V_{ia}$ or $V_{ia} \rightarrow V_{jb}$ and $V_{jb} \rightarrow V_{ia}$. Clearly the obtained set $V_i \cup V_j$ induces a connected subdigraph of D. Let $Q_1, \ldots, Q_r$ denote the sets obtained, by repeating this process until no more changes occur. If $r = 1$, then R cannot be semicomplete multipartite. Otherwise, R is the semicomplete multipartite digraph obtained by set-contracting each connected component of Q_i into a vertex.

Checking whether R can be a semicomplete bipartite digraph or a connected extended locally semicomplete digraph is left as Exercise 4.38.

It is not difficult to see that, for every R being either acyclic or semicomplete multipartite, the procedures above can be realized as an $O(nm + n^2)$-algorithm. The same complexity is proved for semicomplete bipartite digraphs and extended locally semicomplete digraphs in Exercise 4.38. □

Theorem 4.12.3 *[62] There exists an $O(n^2m+n^3)$-algorithm for checking if a digraph with n vertices and m arcs is totally Φ_i-decomposable for $i = 0, 1, 2$.*

Proof: We describe a recursive algorithm to check Φ_i-decomposability. We have shown in Lemma 4.12.2 how to verify whether $D = R[H_1, \ldots, H_r]$, $r \geq 2$, where R is acyclic, semicomplete multipartite, semicomplete bipartite or connected extended locally semicomplete. Whenever we find an R that could be used, the algorithm checks total Φ_i-decomposability of $H_1, \ldots, H_r$ in recursive calls.

Notice how the algorithm exploits the fact that total Φ_i-decomposability is a hereditary property (see Lemma 4.12.1): if some R is found appropriate, then R can be used, because if D is totally Φ_i-decomposable, then each of $H_1, \ldots, H_r$ (being an induced subdigraph of D) must also be totally Φ_i-decomposable. Since there are $O(n)$ recursive calls, the complexity of the algorithm is $O(n^2m + n^3)$. □

4.13 Intersection Digraphs

Let U and V be sets and let $\mathcal{F} = \{(S_v, T_v) : S_v, T_v \subseteq U \text{ and } v \in V\}$ be a family of ordered subsets of U (one for each $v \in V$). The **intersection digraph** corresponding to $\mathcal{F}$ is the digraph $D_{\mathcal{F}} = (V, A)$ such that $vw \in A$ if and only if $S_v \cap T_w \neq \emptyset$. The set U is called the **universal set** for $D_{\mathcal{F}}$. The above family of pairs form a **representation** of D. The concept of an intersection digraph is a natural analogue of the notion of an intersection graph and was introduced by Beineke and Zamfirescu [101] and Sen, Das, Roy and West [661]. Since an arc is an ordered pair of vertices, every line digraph $L(D)$ is the intersection digraph of the family $A(D')$, where D' is the converse of D. It follows from the definition of an intersection digraph that every digraph D is the intersection digraph of the family $\{(A^+(v), A^-(v)) : v \in V(D)\}$, where $A^+(v)$ $(A^-(v))$ is the set of arcs leaving v (entering v). Here the universal set is $A(D)$.

Clearly, a digraph can be represented as the intersection digraph of various families of ordered pairs. It is quite natural to ask how large the universal set U has to be. For a digraph D the minimum number of elements in U such that $D = D_{\mathcal{F}}$ for some family $\mathcal{F}$ of ordered pairs of subsets of U is called the **intersection number**, $\mathrm{in}(D)$ of D. Sen, Das, Roy and West [661] prove the following theorem for the intersection number of an arbitrary digraph D.

For a digraph $D = (V, A)$, a set $B \subseteq A$ is **one-way** if there is a pair of sets $X, Y \subset V$ (called a **generating pair**) such that $B = (X, Y)_D$, that is, B is the set of arcs from X to Y.

Theorem 4.13.1 *[661] The intersection number of a digraph $D = (V, A)$ equals the minimum number of one-way sets required to cover A.*

Proof: Let $B_1, \ldots, B_k$ be a minimum collection of one-way sets covering A and let $(X_1, Y_1), \ldots, (X_k, Y_k)$ be the corresponding generating pairs. Let $S_v = \{i : \ v \in X_i\}$, and $T_v = \{i : \ v \in Y_i\}$. Then $S_v \cap T_w \neq \emptyset$ if and only if $vw \in A$, showing that $\mathrm{in}(D) \leq k$.

Now let U be a universal set of cardinality $u = \mathrm{in}(D)$ such that D has a representation by a set of ordered pairs (S_v, T_v) of subsets of U. We may assume that $U = \{1, 2, \ldots, u\}$. Define u one-way sets covering A as follows: $v \in X_i$ if and only if $i \in S_v$ and $v \in Y_i$ if and only if $i \in T_v$. Then $vw \in A$ if and only if $v \in X_i, \ w \in Y_i$ for some i. Thus, $k \leq \mathrm{in}(D)$. $\square$

A **subtree intersection digraph** is a digraph representable as the intersection digraph of a family of ordered pairs of subtrees in an undirected tree. A **matching diagram digraph** is digraph representable as the intersection digraph of a family of ordered pairs of straight-line segments between two parallel lines. An **interval digraph** is a digraph representable as the intersection digraph of a family of ordered pairs of closed intervals on the real line. Subtree intersection digraphs, matching diagram digraphs and interval digraphs are 'directed' analogues of chordal graphs, permutation graphs and interval graphs, respectively, where subtrees, straight-line segments and real line intervals are also used for representation (see the book [331] by Golumbic). While chordal graphs form a special family of undirected graphs, Harary, Kabell and McMorris showed that every digraph is a subtree intersection digraph.

Proposition 4.13.2 *[400] Every digraph is a subtree intersection digraph.*

Proof: Let $D = (V, A)$ be an arbitrary digraph. Let $G = (U, E), U = V \cup \{x\}$, $E = \{\{x, v\} : \ v \in V\}$, $x \notin V$. Clearly, G is an undirected tree. Setting $S_v = G\langle\{v\}\rangle$ and $T_v = G\langle\{x\} \cup \{w : \ wv \in A\}\rangle$ provides the required representation. $\square$

The following construction by Müller shows that every interval digraph is a matching diagram digraph [576]. Let $\{([a_v, b_v], [c_v, d_v] : \ v \in V(D)\}$ be a representation of an interval digraph D. To obtain a representation $\{(S_v, T_v) : \ v \in V(D)\}$ of D as a matching diagram digraph we set S_v to be the line segment between points $(a_v, 0)$ and $(b_v, 1)$ in the plane, and T_v to be the line segment connecting the points $(c_v, 1)$ and $(d_v, 0)$.

There are several characterizations of interval digraphs, see, e.g., the papers [650] by Sanyal and Sen and [736] by West. We restrict ourselves to just one of them.

Theorem 4.13.3 *[661] A digraph D is an interval digraph if and only if there exist independent row and column permutations of the adjacency matrix $M(D)$ of D which result in a matrix M' satisfying the following property: the zero entries of M' can be labeled R or C such that every position above and to the right of an R is an R and every position below and to the left of a C is a C.* □

None of the characterizations given in [650, 736] implies a polynomial algorithm to recognize interval digraphs. Müller [576] obtained such an algorithm. A polynomial algorithm is also given in [576] to recognize unit interval digraphs, i.e., interval digraphs who have interval representations, where all intervals are of the same length.

4.14 Planar Digraphs

We now discuss planar (di)graphs, i.e. (di)graphs that can be drawn without crossings between (arcs) edges (except at endpoints). Clearly this property does not depend on the orientation of the arcs and hence we can ignore the orientation below when we give a formal definition. Furthermore, most of the results and definitions in this section are for undirected graphs, but are valid also for planar digraphs as far as their underlying graphs are concerned.

An undirected graph $G = (V, E)$ is **planar** if there exists a mapping f which maps G to $\mathcal{R}^2$ in the following way:

> Each vertex is mapped to a point in $\mathcal{R}^2$ and distinct vertices are mapped to distinct points.
> Each edge $uv \in E$ is mapped to a simple (that is, not self-intersecting) curve C_{uv} from $f(u)$ to $f(v)$ and no two curves corresponding to distinct edges intersect, except possibly at their endpoints.

For algorithmic purposes as well as for arguing about planar graphs, it is inconvenient to allow arbitrary curves in the embeddings of planar graphs. A **polygonal curve** from u to v is a piecewise linear curve consisting of finitely many lines such that the first line starts at u, the last line ends at v and each other line starts at the last point of the previous line. Since we can approximate any simple curve arbitrarily well by a polygonal curve we may assume that the curves used in the embedding are always polygonal curves.

A planar graph G may have many different embeddings in the plane (each embedding corresponds to a mapping f as above). Sometimes we wish to refer to properties of a specific embedding f of G. In this case we say that G is **plane** (that is, already embedded) with planar embedding f. A plane graph G partitions $\mathcal{R}^2$ into a finite number of (topologically) connected regions called **faces**. Precisely one of these faces is unbounded and we call this the **outer face**. It is easy to see that, for any fixed face F of G, we may reembed G in

$\mathcal{R}^2$ in such a way that F becomes the outer face. The boundary of a face F is denoted by $bd(F)$ and we normally describe a face by listing the vertices in clockwise order around the face (for the unbounded face this corresponds to listing the vertices on the boundary in the anti-clockwise order). See Figure 4.20 for an illustration of the definitions.

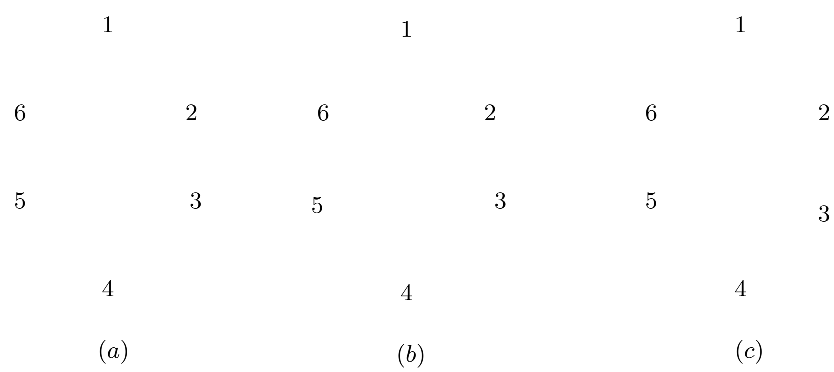


Figure 4.20 (a) shows a non-planar embedding of a graph H; (b) shows a planar embedding of H; (c) shows a planar embedding of H where all curves are polygonal. With respect to the embedding in (c), the faces are 12341, 14561, 16321 and 36543. The outer face is 36543.

Observe that, if we add the edge 25 to the graph H in Figure 4.20, then the resulting graph, which is isomorphic to $K_{3,3}$, is no longer planar. In fact planar graphs have a famous characterization, due to Kuratowski:

Theorem 4.14.1 (Kuratowski's theorem) *[507] A graph has a planar embedding if and only if it does not contain a subdivision[5] of K_5 or $K_{3,3}$.* □

Based on this it is possible to show that planar graphs (and hence also planar digraphs) can be recognized efficiently. In fact Hopcroft and Tarjan [432] showed that it can be done in linear time and if the graph is planar, one can find a planar embedding in the same time.

The following relation between the number of vertices, edges and faces in a plane graph, known as **Euler's formula**, is easy to prove by induction on the number of faces.

Theorem 4.14.2 *If G is a connected plane graph on n vertices and m edges, then*

$$n - m + \phi = 2,$$

where ϕ denotes the number of faces in the embedding on G. In particular the number of faces is the same in every embedding of G. □

[5] A **subdivision** H' of a graph H is any graph that can be obtained from H by replacing each edge by a path all of whose internal vertices have degree 2 in H'.

We leave it to the reader to derive the following easy consequence of Theorem 4.14.2 (see Exercise 4.42):

Corollary 4.14.3 *For every planar graph on $n \geq 3$ vertices and m edges we have $m \leq 3n - 6$.* □

If we allow multiple edges, then we cannot bound the number of edges as we did above. However for planar digraphs we have the following easy consequence:

Corollary 4.14.4 *No planar digraph on $n \geq 3$ vertices has more than $6n-12$ arcs.* □

For much more information about drawings of graphs (in particular embeddings of planar graphs) we refer the reader to the recent book [97] by Battista, Eades, Tamassia and Tollis. This book also contains a number of results on how to use digraph techniques (in particular network flows) to obtain nice drawings of (di)graphs.

4.15 Application: Gaussian Elimination

In many applications, such as modeling a problem by a system of differential equations and then solving this system by numerical methods (cf. the book [208] by Duff, Erisman and Reid), the final step of the solution of the problem under consideration consists of solving a system of linear equations: $Ax = b$, where $A = [a_{ij}]$ is an $n \times n$ matrix of coefficients, b is a given vector of dimension n and x is a vector of unknowns. In a considerable number of applications the matrix A is sparse, i.e., most entries of A are zero. The system $Ax = b$ is often solved by the Gaussian elimination method. To use this method, the only requirement is that all diagonal elements a_{ii} of matrix A can be made non-zero row and column permutations.

In many cases in practice, a sparse matrix A has some special structure, which allows one to solve the system much faster than just using Gaussian elimination directly. One of the most important such structures is block-triangular structure. Let $n_1, n_2, \ldots, n_k$ be natural numbers such that $1 \leq n_1 < n_2 < \ldots < n_k = n$ and let $n_0 = 0$. We call the submatrices $A^{(p)} = [a_{i_p, j_p}]$, with $n_{p-1} + 1 \leq i_p, j_p \leq n_p$, the **main $(n_1, \ldots, n_p)$-blocks** (or just main blocks). We say that A has **$(n_1, \ldots, n_p)$-block-triangular structure** (or just block-triangular structure) if all entries of A below the main blocks are zero. (More precisely, one should call this structure upper block-triangular [208], but since we do not consider lower block-triangular structure here, we will omit the word 'upper'.) The matrix

$$\begin{bmatrix} 3 & 2 & 4 & 1 \\ 5 & 6 & 0 & 0 \\ 3 & 0 & 7 & 9 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

has $(3,1)$-block-triangular structure. See also Figure 4.21.

n_0

n_1

n_2

n_3

n_4

Figure 4.21 An (n_1, n_2, n_3, n_4)-block-triangular structure. White space consists of entries equal zero.

If A has block-triangular structure, we solve first the system $A^{(p)}x^{(p)} = b^{(p)}$, where $x^{(p)}$ ($b^{(p)}$) is the vector consisting of n_p last coordinates of x (b). The values of coordinates of $x^{(p)}$, which we found, equal the values of the corresponding unknowns in the system $Ax = b$ since in the last n_p rows of A all coefficients except for some in the last n_p columns are zero. Taking into consideration that the values of coordinates of $x^{(p)}$ are already found, we can compute the values of coordinates of $x^{(p-1)}$ using the block $A^{(p-1)}$. Similarly, using all blocks of A (in the decreasing order of their indices) we can compute all coordinates in x.

However, quite often the block-triangular structure of A is **hidden**, i.e. A has no block-triangular structure, but A can be transformed into a matrix with block-triangular structure after certain permutations π and τ of its rows and columns, respectively. Here we are interested in using the Gaussian elimination method and thus we assume that all diagonal entries of A are non-zero (when it is possible, one can find permutations of rows and columns of A, which bring non-zero diagonal to A using perfect matchings in bipartite graphs, see [208]). Therefore, we do not wish to change the diagonal entries of A. This can be achieved by using only simultaneous permutations of rows and columns of A, i.e. $\pi = \tau$.

To reveal hidden block-triangular structure of A, the following approach can be used. Let us replace all non-zero entries of A by 1. We obtain matrix $B = [b_{ij}]$, which can be viewed as the adjacency matrix of some directed pseudograph D with vertex set $\{v_1, \ldots, v_n\}$, i.e. $b_{ij} = 1$ if and only if $v_i \rightarrow v_j$ in D. (Clearly, D has no parallel arcs, but due to the assumption on the diagonal elements it has a loop at every vertex.) Suppose that D is not strong, $D_1, \ldots, D_p$ is the acyclic ordering of the strong components of D (i.e. there is no arc from D_j to D_i if $j > i$) and the vertices of D are ordered $v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)}$ such that

$$V(D_i) = \{v_{\pi(n_i+1)}, v_{\pi(n_i+2)}, \ldots, v_{\pi(n_{i+1})}\}.$$

It is easy to see that B has $(n_1, \ldots, n_p)$-block-triangular structure. This implies that A has block-triangular structure. The above observation suggests the following procedure to reveal hidden block-triangular structure of A.

1. Replace every non-zero entry of A by 1 to obtain a $(0, 1)$-matrix B.
2. Construct a directed pseudograph D with vertex set $\{v_1, \ldots, v_n\}$ such that B is the adjacency matrix of D.
3. Find the strong components of D. If D is strong, then B (and thus A) does not have hidden block-triangular structure[6]. If D is not strong, let $D_1, \ldots, D_p$ be the strong components of D (in acyclic order). Find a permutation π on $\{1, \ldots, n\}$ such that

$$V(D_i) = \{v_{\pi(n_i+1)}, v_{\pi(n_i+2)}, \ldots, v_{\pi(n_{i+1})}\}.$$

   This permutation reveals hidden block-triangular structure of B (and thus A). Use π to permute rows and columns of A and coordinates of x and b.

To perform Step 3 one may use Tarjan's algorithm in Section 4.4.

We will illustrate the procedure above by the following example. Suppose we wish to solve the system:

$$\begin{aligned} x_1 \qquad\quad + 3x_3 + 8x_4 &= 2,\\ x_2 \qquad\quad + 5x_4 &= 1,\\ 2x_1 + 2x_2 + 4x_3 + 9x_4 &= 6,\\ 3x_2 \qquad\quad + 2x_4 &= 3. \end{aligned}$$

We first construct the matrix B and the directed pseudograph D. We have $V(D) = \{v_1, v_2, v_2, v_4\}$ and

$$A(D) = \{v_1v_3, v_1v_4, v_2v_4, v_3v_1, v_3v_2, v_3v_4, v_4v_2\} \cup \{v_iv_i : \ i = 1, 2, 3, 4\}.$$

The digraph D has strong components $D^{(1)}$ and $D^{(2)}$, which are subdigraphs of D induced by $\{v_1, v_3\}$ and $\{v_2, v_4\}$, respectively. These components suggest the following permutation π, $\pi(i) = i$ for $i = 1, 4$, $\pi(2) = 3$ and $\pi(3) = 2$, of rows and columns of A as well as elements of x and b, the right-hand side. As a result, we obtain the following:

$$\begin{aligned} x'_1 + 3x'_2 \qquad\quad + 8x'_4 &= 2,\\ 2x'_1 + 4x'_2 + 2x'_3 \quad 9x'_4 &= 6,\\ x'_3 + 5x'_4 &= 1,\\ 3x'_3 + 2x'_4 &= 3, \end{aligned}$$

[6] Provided we do not change the set of entries of the diagonal of A

where $x'_i = x_i$ for $i = 1, 4$, $x'_2 = x_3$ and $x'_3 = x_2$.

Solving the last two equations separately, we obtain $x'_3 = 1,\ x'_4 = 0$. Now solving the first two equations, we see that $x'_1 = 2,\ x'_2 = 0$. Hence, $x_1 = 2,\ x_2 = 1,\ x_3 = x_4 = 0$.

A discussion on practical experience with revealing and exploiting block-triangular structures is given in [208].

4.16 Exercises

4.1. Let $\phi(u)$ be the forefather of a vertex u as defined in Section 4.4. Combining (4.2) and (4.3), prove that $\phi(\phi(u)) = \phi(u)$.

4.2. Prove Proposition 4.3.1.

4.3. Prove Lemma 4.4.1.

4.4. In part (ii) $\Rightarrow$ (i) of Theorem 4.5.1, prove that $\sigma(D) = L(Q)$.

4.5. Derive Corollary 4.5.2 from Theorem 4.5.1 (iii).

4.6. (−) Prove Proposition 4.5.3 using Theorem 4.5.1 (i) and (ii).

4.7. Prove the following simple properties of line digraphs:
(i) $L(D) \cong \vec{P}_{n-1}$ if and only if $D \cong \vec{P}_n$;
(ii) $L(D) \cong \vec{C}_n$ if and only if $D \cong \vec{C}_n$.

4.8. Let D be a digraph. Show by induction that $L^k(D)$ is isomorphic to the digraph H, whose vertex set consists of walks of D of length k and a vertex $v_0v_1 \ldots v_k$ dominates the vertex $v_1v_2 \ldots v_kv_{k+1}$ for every $v_{k+1} \in V(D)$ such that $v_kv_{k+1} \in A(D)$.

4.9. Using the results in Exercise 4.7, prove the following elementary properties of iterated line digraphs: Let D be a digraph. Then
(i) $L^k(D)$ is a digraph with no arcs, for some k, if and only if D is acyclic;
(ii) if D has a pair of cycles joined by a path (possibly of length 0), then

$$\lim_{k \longrightarrow \infty} n_k = \infty,$$

where n_k is the order of $L^k(D)$;
(iii) if no pair of cycles of D is joined by a path, then for all sufficiently large values of k, each connected component of $L^k(D)$ has at most one cycle.

4.10. Prove by induction on $k \geq 1$ Proposition 4.5.4.

4.11. Prove Lemma 4.6.1.

4.12. Prove Lemma 4.6.5.

4.13. Prove Lemma 4.6.6.

4.14. Prove Theorem 4.6.7.

4.15. **Upwards embeddings of MVSP digraphs.** Prove that one can embed every MVSP digraph D into the Cartesian plane such that, if vertices u, v have coordinates (x_u, y_u) and (x_v, y_v), respectively, and there is a (u, v)-path in D, then $x_u \leq x_v$ and $y_u \leq y_v$. Hint: consider series composition and parallel composition separately.

4.16. Prove Proposition 4.7.2. Hint: use induction on the number of reductions applied for the 'if' part and the number of arcs for the 'only if' part.

4.17. Prove Proposition 4.7.3.

4.18. Prove part (b) of Lemma 4.8.4. Hint: if u and v are in S then there is a path from u to v in $\overline{UG(S)}$. Similarly, if x and y are in S'. Use these paths (corresponding to sequences of non-adjacent vertices in D) to show that if xu and vy are arcs, then $u = v$ and $x = y$ must hold if D is quasi-transitive.

4.19. $(-)$ Construct an infinite family of path-mergeable digraphs, which are not in-path-mergeable.

4.20. Prove Proposition 4.10.2.

4.21. $(-)$ Show that the following 'claim' is wrong. Let D be a locally in-semicomplete digraph and let D contain internally disjoint paths P_1, P_2 such that P_i is an (x_i, y)-path $(i = 1, 2)$ and $x_1 \neq x_2$. Then x_1 and x_2 are adjacent.

4.22. **Orientations of path-mergeable digraphs.** Prove that every orientation of a path-mergeable digraph is a path-mergeable oriented graph.

4.23. $(+)$ Prove Corollary 4.9.2.

4.24. **Path-mergeable digraphs which are neither locally in-semicomplete nor locally out-semicomplete.** Show by a construction that there exists an infinite class of path-mergeable digraphs, none of which is locally in-semicomplete or locally out-semicomplete. Then extend your construction to arbitrary degrees of vertex-strong connectivity. Hint: consider extensions.

4.25. $(-)$ **Path-mergeable transitive digraphs.** Prove that a transitive digraph $D = (V, A)$ is path-mergeable if and only if for every $x, y \in V$ and every pair xuy, xvy of (x, y)-path of length 2 either $u{\rightarrow}v$ or $v{\rightarrow}u$ holds.

4.26. Reformulate Lemma 4.10.3 and Theorem 4.10.4 for locally out-semicomplete digraphs.

4.27. **Orientations of locally in-semicomplete digraphs**. Prove that every orientation of a digraph which is locally in-semicomplete is a locally in-tournament digraph.

4.28. **Strong orientations of strong locally in-semicomplete digraphs**. Prove that every strong locally in-semicomplete digraph on at least 3 vertices has a strong orientation.

4.29. Prove Lemma 4.11.2.

4.30. Prove Corollary 4.11.7.

4.31. Prove Theorem 4.11.8.

4.32. **Recognition of round digraphs.** Show that the proof of Theorem 4.11.4 implies a polynomial algorithm to decide whether a digraph D is round and to find a round labelling of D (if D is round).

4.33. **(+)** Using Lemma 4.11.13, show that, if D is a non-round decomposable locally semicomplete digraph, then the independence number of $UG(D)$ is at most two.

4.34. **(−)** Give an example of a locally semicomplete digraph on 4 vertices with no 2-king.

4.35. Prove Proposition 4.11.16.

4.36. Prove the assertion stated in Exercise 4.33 using Lemma 4.11.14 and Proposition 4.11.16.

4.37. **Extending in-path-mergeability.** Prove that, if P, Q are internally disjoint (x,z)- and (y,z)-paths in an extended locally in-semicomplete digraph D and no vertex on $P-z$ is similar to a vertex of $Q-z$, then there is a path R from either x or y to z in D such that $V(R) = V(P) \cup V(Q)$.

4.38. Prove that there exists an $O(mn+n^2)$-algorithm for checking if a digraph D with n vertices and m arcs has a decomposition $D = R[H_1, \ldots, H_r]$, $r \geq 2$, where H_i is an arbitrary digraph and the digraph R is either semicomplete bipartite or connected extended locally semicomplete.

4.39. **(−)** Let D be a connected digraph which is both quasi-transitive and locally semicomplete. Prove that D is semicomplete.

4.40. **(−)** Let D be a connected digraph which is both quasi-transitive and locally in-semicomplete. Prove that the diameter of $UG(D)$ is at most 2.

4.41. **(−)** Prove that the intersection number $\mathrm{in}(D) \leq n$ for every digraph D of order n. Show that this upper bound is sharp (Sen, Das, Roy and West [661]).

4.42. Prove Corollary 4.14.3. Hint: use that each edge is on the boundary of precisely two faces and that each face has at least 3 edges.

4.43. **(−)** Check which of the following 4×4-matrices $A = [a_{ij}]$ have hidden block-triangular structure (the entries not specified equal zero). Only simultaneous permutations of rows and columns are allowed.

(a) $a_{1i} = i+1$ for $i = 1,2,3$, $a_{2i} = a_{3i} = i$ for $i = 2,3$, and $a_{4i} = 2$ for $i = 2,3,4$;

(b) $a_{12} = a_{21} = a_{14} = a_{41} = a_{34} = a_{43} = 2$ and $a_{ii} = 1$ for $i = 1,2,3,4$.

5. Hamiltonicity and Related Problems

In this chapter we will consider the hamiltonian path and cycle problems for digraphs as well as some related problems such as the longest path and cycle problems and the minimum path factor problem. We describe and prove a number of results in the area as well as formulate several open questions.

We recall that a k-path factor of a digraph D is a collection of k vertex-disjoint paths covering $V(D)$. Recall that the minimum positive integer k such that D has a k-path factor is the path covering number of D, denoted by $\mathrm{pc}(D)$. A $\mathrm{pc}(D)$-path factor of D is also called a **minimum path factor** of D. Recall also that a digraph is traceable if it contains a hamiltonian path.

For arbitrary digraphs the hamiltonian path and hamiltonian cycle problems are very difficult and both are $\mathcal{NP}$-complete (see, e.g. the book [303] by Garey and Johnson). For convenience of later referencing we state these results as theorems.

Theorem 5.0.1 *The problem to check whether a given digraph has a hamiltonian cycle is $\mathcal{NP}$-complete.* □

Theorem 5.0.2 *The problem to check whether a given digraph has a hamiltonian path is $\mathcal{NP}$-complete.* □

It is worthwhile mentioning that the hamiltonian cycle and path problems are $\mathcal{NP}$-complete even for some special classes of digraphs. Garey, Johnson and Tarjan showed [305] that the problem remains $\mathcal{NP}$-complete even for planar 3-regular digraphs. It follows easily from Theorems 5.0.1 and 5.0.2 that the problem to determine the minimum path factor as well as the longest path and cycle problems are $\mathcal{NP}$-hard as optimization problems for arbitrary digraphs. This is also true for several special classes of digraphs. However, for some important special classes of digraphs these problems are polynomial time solvable. One such class is the class of acyclic digraphs (see Theorem 2.3.5 and Section 5.3). The reader will see in this chapter that many more such classes can be found.

In Section 5.1, some powerful necessary conditions, due to Gutin and Yeo, are considered for a digraph to be hamiltonian. These conditions can be used for the hamiltonian path problem due to the following simple observation:

Proposition 5.0.3 *A digraph D has a Hamilton path if and only if the digraph D^*, obtained from D by adding a new vertex x^* such that x^* dominates every vertex of D and is dominated by every vertex of D, is hamiltonian.* □

In Section 5.2 we prove that the path covering number of an arbitrary digraph is never more than its independence number. In Section 5.3 we show that the minimum path factor problem for acyclic digraphs can be solved quite efficiently. Furthermore, we show that algorithms for finding minimum path factors in acyclic digraphs are useful in a number of applications.

In Section 5.4, we obtain necessary and sufficient conditions by Bang-Jensen for a path-mergeable digraph to be hamiltonian. Since locally in-semicomplete and out-semicomplete digraphs are proper subclasses (see Proposition 4.10.1) of path-mergeable digraphs, we may use these conditions, in Section 5.5, to derive a characterization of hamiltonian locally in-semicomplete and out-semicomplete digraphs. As corollaries, we obtain the corresponding results for locally semicomplete digraphs. Digraphs with restricted degrees are considered in Section 5.6. There, a number of degree-related sufficient conditions for a digraph to be hamiltonian are described. In that section, we also consider a recently introduced and powerful proof technique, called multi-insertion, that can be applied to prove many theorems on hamiltonian digraphs.

In the last decade quite a number of papers were devoted to studying the structure of longest cycles and paths of semicomplete multipartite digraphs. In Section 5.7, we consider the most important results obtained in this area so far including some striking results by Yeo. The proofs in that section provide further illustrations of the multi-insertion technique. In Section 5.8, we discuss generalizations of characterizations of hamiltonian and traceable extended semicomplete digraphs to extended locally semicomplete digraphs.

Sections 5.9 and 5.10 are devoted to quasi-transitive digraphs. We present two interesting methods to tackle the hamiltonian path and cycle problems, and the longest path and cycle problems, respectively, in this class of digraphs. The second method by Bang-Jensen and Gutin allows one to find even vertex-heaviest paths and cycles in quasi-transitive digraphs in polynomial time (where the weights are on the vertices). The last section is devoted to results on hamiltonian paths and cycles in some classes of digraphs not considered in the previous sections. The proof of Theorem 5.11.2 by Thomassen illustrates how the properties of tournaments can be used to prove results on more general digraphs.

For additional information on hamiltonian and traceable digraphs, see e.g. the surveys [61, 66] by Bang-Jensen and Gutin, [126] by Bondy, [368] by Gutin and [728, 729] by Volkmann.

5.1 Necessary Conditions for Hamiltonicity of Digraphs

An obvious condition for a digraph to be hamiltonian is to be strong. Another obvious and, yet, quite powerful necessary condition for a digraph to be hamiltonian is the existence of a cycle factor[1]. Both conditions can be verified in polynomial time (see Sections 4.4 and 3.11.4). The purpose of this section is to describe a series of more powerful conditions, called k-quasi-hamiltonicity, which were recently introduced by Gutin and Yeo in [379]. An equivalent form of 1-quasi-hamiltonicity, pseudo-hamiltonicity, was actually investigated earlier by Babel and Woeginger in [35] for undirected graphs.

We prove that every $(k+1)$-quasi-hamiltonian digraph is also k-quasi-hamiltonian (however, there are digraphs which are k-quasi-hamiltonian, but not $(k+1)$-quasi-hamiltonian). We introduce an algorithm that checks k-quasi-hamiltonicity of a given digraph with n vertices and m arcs in time $O(nm^k)$. Hence, these conditions can be efficiently verified for small values of k. Thus, they can be incorporated in software systems which investigate properties of digraphs (or graphs); one such system is described by Delorme, Ordaz and Quiroz in [189]. We prove that $(n-1)$-quasi-hamiltonicity coincides with hamiltonicity and 1-quasi-hamiltonicity is equivalent to pseudo-hamiltonicity.

5.1.1 Path-Contraction

In this section we consider, for technical reasons, directed multigraphs. We use a variation of the operation of contraction of a set of vertices in a directed multigraph. This operation is called **path-contraction** and is defined as follows. Let P be an (x,y)-path in a directed multigraph $D=(V,A)$. Then $D//P$ stands for the directed multigraph with vertex set $V(D//P) = V \cup \{z\} - V(P)$, where $z \notin V$, and $\mu_{D//P}(uv) = \mu_D(uv)$, $\mu_{D//P}(uz) = \mu_D(ux)$, $\mu_{D//P}(zv) = \mu_D(yv)$ for all distinct $u, v \in V - V(P)$. In other words, $D//P$ is obtained from D by deleting all vertices of P and adding a new vertex z such that every arc with head x (tail y) and tail (head) in $V - V(P)$ becomes an arc with head (tail) z and the same tail (head). Observe that a path-contraction in a digraph results in a digraph (no parallel arcs arise). We will often consider path-contractions of paths of length one, i.e. arcs e. Clearly, a directed multigraph D has a k-cycle ($k \geq 3$) through an arc e if and only if $D//e$ has a cycle through z. Observe that the obvious analogue of path-contraction for undirected multigraphs does not have this nice property which is of use in this section. The difference between (ordinary) contraction (which is also called **set-contraction**) and path-contraction is reflected in Figure 5.1.

---

[1] Häggkvist [387] posed a problem to find classes of digraphs for which strong connectivity and the existence of a cycle factor are sufficient for hamiltonicity. In this chapter we consider some classes with this property.

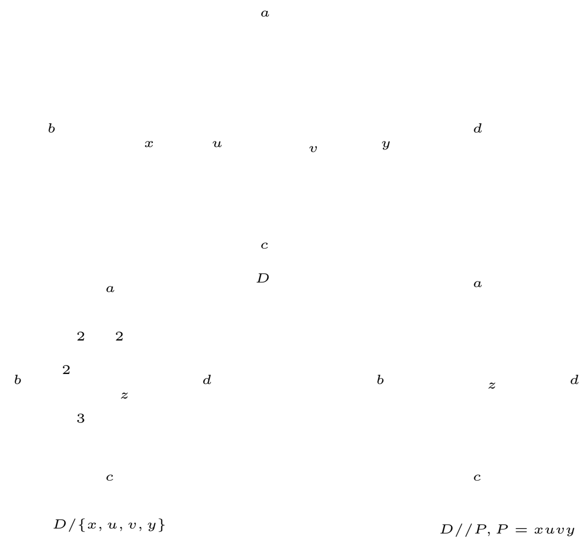


Figure 5.1 The two different kinds of contraction, set-contraction and path-contraction. The integers 2 and 3 indicate the number of corresponding parallel arcs.

As for set-contraction, for vertex-disjoint paths $P_1, P_2, \ldots, P_t$ in D, the path-contraction $D//\{P_1, \ldots, P_t\}$ is defined as the directed multigraph $(\ldots((D//P_1)//P_2)\ldots)//P_t$; clearly, the result does not depend on the order of $P_1, P_2, \ldots, P_t$.

5.1.2 Quasi-Hamiltonicity

The results in the remainder of this section are due to Gutin and Yeo. Let $D = (V, A)$ be a directed multigraph. Let $QH_1(D) = (V, A_1)$ be the directed multigraph with arc set

$$A_1 = \{e \in A : \ e \text{ is contained in a cycle factor of } D\}.$$

For $k \geq 2$, $QH_k(D) = (V, A_k)$ is the directed multigraph with arc set $A_k = \{e \in A : \ QH_{k-1}(D//e) \text{ is strong}\}$. For $k \geq 1$, a directed multigraph D is ***k*-quasi-hamiltonian**, if $QH_k(D)$ is strong. We assume (by definition) that every directed multigraph is 0-quasi-hamiltonian. The quasi-hamiltonicity number of a directed multigraph D of order n, $\mathrm{qhn}(D)$, is the maximum integer $k(< n)$ such that D is k-quasi-hamiltonian.

Figure 5.2 illustrates the notion of quasi-hamiltonicity. The directed multigraph H is 0-quasi-hamiltonian, but not 1-quasi-hamiltonian ($QH_1(H) = H - \{(3,4),(4,3)\}$ is not strong). Hence, $\mathrm{qhn}(H) = 0$. The directed multigraph D is 1-quasi-hamiltonian as $QH_1(D) = D$ is strong (every arc of D

belongs to a cycle factor of D). However, D is not 2-quasi-hamiltonian since $QH_2(D)$ is not strong (indeed, $QH_1(D//(3,4)) = QH_1(L)$ is not strong). Thus, $\mathrm{qhn}(D) = 1$.

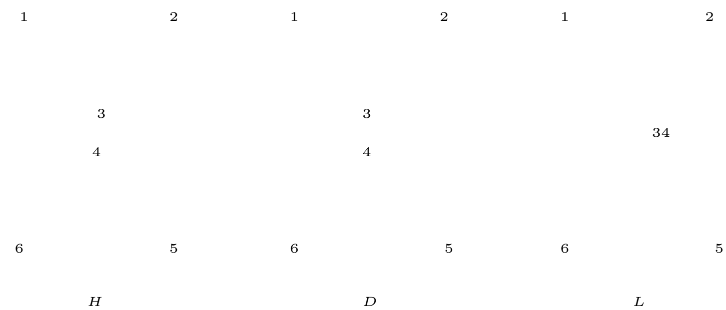


Figure 5.2 Digraphs.

We start with some basic facts on k-quasi-hamiltonicity.

Proposition 5.1.1 *[379] Let D be a directed multigraph of order $n(\geq 2)$ and let $k \in \{2, 3, \dots, n-1\}$. Then $A(QH_k(D)) \subseteq A(QH_{k-1}(D))$. In particular, if D is k-quasi-hamiltonian, it is $(k-1)$-quasi-hamiltonian.*

Proof: We prove the claim by induction on k. Let $e \in A(QH_2(D))$. Thus, $QH_1(D//e)$ is strong which, in particular, means that $D//e$ has a cycle factor. Hence, $e \in A(QH_1(D))$. Let now $k \geq 3$ and let $e \in A(QH_k(D))$. Then, $QH_{k-1}(D//e)$ is strong. By the induction hypothesis, $QH_{k-2}(D//e)$ is also strong. Hence, $e \in A(QH_{k-1}(D))$. □

Theorem 5.1.2 *[379] A directed multigraph is hamiltonian if and only if it is $(n-1)$-quasi-hamiltonian.*

Proof: Clearly every hamiltonian directed multigraph of order 2 is 1-quasi-hamiltonian. Now assume that all hamiltonian directed multigraphs of order $n-1$ are $(n-2)$-quasi-hamiltonian, and let D be a hamiltonian digraph of order n. Whenever we contract an arc belonging to a hamiltonian cycle we obtain a hamiltonian digraph of order $n-1$, which therefore is $(n-2)$-quasi-hamiltonian. Hence, every arc on a Hamilton cycle lies in $QH_{n-1}(D)$, which implies that $QH_{n-1}(D)$ is strong, i.e. D is $(n-1)$-quasi-hamiltonian. Thus, the ‘only if’ part is proved.

We prove the ‘if’ part. Let D be a directed multigraph, such that $QH_{n-1}(D)$ is strong. Let e_1 be an arc in $QH_{n-1}(D)$. Since $QH_{n-2}(D//e_1)$ is strong there exists an arc e_2 in $QH_{n-2}(D//e_1)$. Since $QH_{n-3}((D//e_1)//e_2)$

is strong there exists an arc e_3 in $QH_{n-3}((D/e_1)//e_2)$. Continuing this procedure we obtain arcs $e_1, e_2, \ldots, e_{n-2}$, such that the directed multigraph $QH_1((((D//e_1)//e_2)\ldots)//e_{n-2})$ is strong. Let

$$D' = (((D//e_1)//e_2)\ldots)//e_{n-2},$$

and observe that, since $QH_1(D')$ is strong and D' has order 2, D' must be hamiltonian. By inserting the arcs $e_1, e_2, \ldots, e_{n-2}$ into a Hamilton cycle in D', we obtain a Hamilton cycle in D. □

We leave the proof of the following theorem as a non-trivial exercise (Exercise 5.1).

Theorem 5.1.3 *[379] For every $k \geq 0$, there exists a digraph D such that $qhn(D) = k < n$.* □

5.1.3 Pseudo-Hamiltonicity and 1-Quasi-Hamiltonicity

For a positive integer h, a sequence of vertices $Q = v_1v_2\ldots v_{hn}v_1$ in a directed multigraph D of order n is an ***h*-pseudo-hamiltonian walk** if every vertex of D appears h times in the sequence $v_1v_2\ldots v_{hn}$ and $v_iv_{i+1} \in A(D)$ for every $i = 1, 2, \ldots, hn$ ($v_{hn+1} = v_1$). A directed multigraph D possessing such a sequence is called ***h*-pseudo-hamiltonian** and the minimum h for which D is h-pseudo-hamiltonian is the **pseudo-hamiltonicity number** $\mathrm{ph}(D)$ of D. If D has no h-pseudo-hamiltonian walk for any positive integer h, then $\mathrm{ph}(D) = \infty$. A directed multigraph D is **pseudo-hamiltonian** if $\mathrm{ph}(D) < \infty$.

For example, in Figure 5.2, the digraph D is 2-pseudo-hamiltonian: 1212346565431 is a 2-pseudo-hamiltonian walk of D. This digraph is not 1-pseudo-hamiltonian as D is not hamiltonian. Thus, $\mathrm{ph}(D) = 2$. It is not difficult to see that the digraph H in Figure 5.2 is not pseudo-hamiltonian. We have already seen that D is 1-quasi-hamiltonian, but H is not. The above conclusions on pseudo-hamiltonicity of D and H can actually be obtained from Theorem 5.1.5.

Lemma 5.1.4 follows from the fact that every regular directed multigraph has a cycle factor (see Exercise 3.70), which implies that every h-regular directed multigraph can be decomposed into h cycle factors.

Lemma 5.1.4 *Every arc of a regular directed multigraph is included in a cycle factor.* □

Theorem 5.1.5 *[379] A directed multigraph is pseudo-hamiltonian if and only if it is 1-quasi-hamiltonian.*

Proof: Let D be a pseudo-hamiltonian directed multigraph, let Q be an h-pseudo-hamiltonian walk in D, and let $A(Q) = (v_1v_2, v_2v_3, \ldots, v_{hn-1}v_{hn}, v_{hn}v_1)$ be the sequence of arcs in Q. Construct a new directed multigraph $H(D,Q)$ from D by replacing, for every pair x, y with $\mu_D(xy) > 0$, all arcs from x to y in D by $t(\geq 0)$ parallel arcs from x to y, where t is the number of appearances of xy in $A(Q)$. By the definition of an h-pseudo-hamiltonian walk, $H(D,Q)$ is an h-regular directed multigraph. Thus, by Lemma 5.1.4, every arc xy in $H(D,Q)$ is in a cycle factor. Therefore, $\mu_{H(D,Q)}(xy) > 0$ implies $\mu_{QH_1(D)}(xy) > 0$. Since $H(D,Q)$ is strong, we obtain that $QH_1(D)$ is also strong, i.e. D is 1-quasi-hamiltonian.

Now let D be a 1-quasi-hamiltonian directed multigraph, i.e. $QH_1(D)$ is strong. For each arc e in $QH_1(D)$ let $\mathcal{F}_e$ be a cycle factor in D including e. Let $D' = \cup_{e \in A(QH_1(D))} \mathcal{F}_e$. As the union of cycle factors, D' is regular. Since $QH_1(D)$ is strong, D' is also strong. Therefore, D' has a eulerian trail, which corresponds to a pseudo-hamiltonian walk in D. □

The following theorem provides a sharp upper bound for the pseudo-hamiltonicity number of a digraph.

Theorem 5.1.6 *[379] For a pseudo-hamiltonian digraph D,* $\mathrm{ph}(D) \leq (n-1)/2$. *For every integer $n \geq 3$, there exists a digraph H_n of order n such that* $\mathrm{ph}(H_n) = \lfloor (n-1)/2 \rfloor$.

Proof: Exercise 5.2. □

5.1.4 Algorithms for Pseudo- and Quasi-Hamiltonicity

It is easy to check whether a digraph is 1-quasi-hamiltonian (i.e., by Theorem 5.1.5 is pseudo-hamiltonian). Indeed, checking whether $QH_1(D)$ is strong can be done in time $O(n+m)$ (see Section 4.4). Hence, it suffices to show how to verify for each arc xy if this arc is on some cycle factor. We can merely replace xy by a path xzy, where z is not in D, and check whether the new digraph has a cycle factor. This can be done in time $O(\sqrt{n}m)$ by Corollary 3.11.7. Thus, we obtain the total time of $O(\sqrt{n}m^2)$. This complexity bound was improved by Gutin and Yeo [379] as follows.

Theorem 5.1.7 *We can check whether a directed multigraph D is pseudo-hamiltonian in $O(nm)$ time.*

Proof: Exercise 5.3. □

The following theorem implies that one can check k-quasi-hamiltonicity for a constant k in polynomial time.

Theorem 5.1.8 *[379] In $O(nm^k)$ time, one can check if a directed multigraph is k-quasi-hamiltonian.*

Proof: In this proof, we describe an algorithm $\mathcal{A}$ that, in time $T(k)$, checks whether a directed multigraph D is k-quasi-hamiltonian. We will show that $T(k) = O(nm^k)$.

If $k = 1$, the algorithm $\mathcal{A}$ uses the algorithm $\mathcal{B}$ of Theorem 5.1.7. Thus, $T(1) = O(nm)$. If $k \geq 2$ then, for each arc e in D, $\mathcal{A}$ verifies whether $D//e$ is $(k-1)$-quasi-hamiltonian. The algorithm $\mathcal{A}$ forms $QH_k(D)$ from all arcs e such that $D//e$ is $(k-1)$-quasi-hamiltonian. Finally, $\mathcal{A}$ checks whether $QH_k(D)$ is strong (in time $O(m)$). This implies that, for $k \geq 2$,

$$T(k) \leq mT(k-1) + O(m).$$

Since $T(1) = O(nm)$, we obtain that $T(k) = O(nm^k)$. □

5.2 Path Covering Number

The following attainable lower bound for the path covering number of a digraph D is quite trivial: $\mathrm{pcc}(D) \leq \mathrm{pc}(D)$. We will see later in this chapter that $\mathrm{pcc}(D) = \mathrm{pc}(D)$ for acyclic digraphs and semicomplete multipartite digraphs D. The aim of this short section is to obtain a less trivial attainable upper bound for $\mathrm{pc}(D)$. This bound is of use in several applications (see, e.g., Section 5.3).

Recall that the independence number $\alpha(D)$ of a digraph D is the cardinality of a maximum independent set of vertices of D (a set $X \subseteq V(D)$ is independent if no pair of vertices in X is adjacent). Rédei's theorem (Theorem 1.4.5) can be rephrased as saying that every digraph with independence number 1 has a hamiltonian path and hence path covering number equal 1. Gallai and Milgram generalized this as follows.

Theorem 5.2.1 (Gallai-Milgram theorem) *[298] For every digraph D, $\mathrm{pc}(D) \leq \alpha(D)$.*

This theorem is an immediate consequence of the following lemma by Bondy [126]:

Lemma 5.2.2 *Let D be a digraph and let $\mathcal{P} = P_1 \cup P_2 \cup \ldots \cup P_s$ be an s-path factor of D. Let $i(\mathcal{P})$ ($t(\mathcal{P})$) denote the set of initial (terminal) vertices of the paths in $\mathcal{P}$. Suppose that $s > \alpha(D)$. Then there exists an $(s-1)$-path factor $\mathcal{P}'$ of D such that $i(\mathcal{P}') \subset i(\mathcal{P})$ and $t(\mathcal{P}') \subset t(\mathcal{P})$.*

Proof: The proof is by induction on n, the order of D. The case $n = 1$ holds vacuously. Let $\mathcal{P}$ be as described in the lemma. Let the path P_j in $\mathcal{P}$ be denoted by $x_{j1}x_{j2}\ldots x_{jr_j}$, $j = 1, 2, \ldots, s$. Since $s > \alpha(D)$ the subdigraph $D\langle i(\mathcal{P})\rangle$ must contain an arc $x_{k1}x_{j1}$ for some $k \neq j$ $(1 \leq k, j \leq s)$.

If $r_k = 1$, then we can replace P_k, P_j by the path $x_{k1}P_j$ and obtain the desired path factor. So suppose that $r_k > 1$. Now consider $D^* = D - x_{k1}$ and the path factor $\mathcal{P}^*$ which we obtain from $\mathcal{P}$ by deleting x_{k1} from the path P_k. Clearly $\alpha(D^*) \leq \alpha(D)$ and we have $i(\mathcal{P}^*) = i(\mathcal{P}) - x_{k1} + x_{k2}$, $t(\mathcal{P}^*) = t(\mathcal{P})$. Thus it follows by the induction hypothesis that D^* has an $(s-1)$-path factor $\mathcal{Q}$ such that $t(\mathcal{Q}) \subset t(\mathcal{P}^*)$, $i(\mathcal{Q}) \subset i(\mathcal{P}^*)$.

If $x_{k2} \in i(\mathcal{Q})$, let Q_p be the path of $\mathcal{Q}$ whose initial vertex is x_{k2}. Replacing Q_p with $x_{k1}Q_p$ we obtain a path factor in D with the desired properties. So suppose that x_{k2} is not an initial vertex of any of the paths in $\mathcal{Q}$. Then x_{j1} must belong to $i(\mathcal{Q})$ and we obtain the desired path factor by replacing the path Q_r of $\mathcal{Q}$ which starts at x_{j1} by the path $x_{k1}Q_r$. □

The following theorem due to Erdős and Szekeres [596] follows easily from Theorem 5.2.1.

Theorem 5.2.3 *Let n, p, q be positive integers with $n > pq$, and let $I = (i_1, i_2, \ldots, i_n)$ be a sequence of n distinct integers. Then there exists either a decreasing subsequence of I with more than p integers or an increasing subsequence of I with more than q integers.*

Proof: Let $D = (V, A)$ be the digraph with $V = \{i_1, i_2, \ldots, i_n\}$ and $A = \{i_m i_k : \ m < k \text{ and } i_m < i_k\}$. Observe the obvious correspondence between independent sets of D and decreasing subsequences of I (respectively, paths of D and increasing subsequences of I). Let $\mathcal{F} = P_1 \cup \ldots \cup P_s$ be a minimal path factor of D. By Theorem 5.2.1, $s \leq \alpha(D)$. Hence, $\alpha(D) \cdot \max_{j=1}^{s} |P_j| \geq n > pq$. Thus, either $\alpha(D) > p$, i.e., there exists a decreasing subsequence with $\alpha(D) > p$ integers, or $\max_{j=1}^{s} |P_j| > q$, i.e., there exists an increasing subsequence with more than q integers. □

Very recently, the following improvement on Theorem 5.2.1 in the case of strong digraphs was proved by Thomassé. This was originally conjectured by Las Vergnas (see [107]).

Theorem 5.2.4 *[695] If a digraph D is strong, then* $\mathrm{pc}(D) \leq \max\{\alpha(D) - 1, 1\}$.

Las Vergnas (see [106]) proved the following generalization of Theorem 5.2.1.

Theorem 5.2.5 *Every digraph D of finite out-radius has an out-branching with at most $\alpha(D)$ vertices of out-degree zero.* □

Theorem 5.2.5 implies Theorem 5.2.1 (Exercise 5.7).

5.3 Path Factors of Acyclic Digraphs with Applications

For acyclic digraphs it turns out that the minimum path factor problem can be solved quite efficiently. This is important since this problem has many practical applications. One such example is as follows.

A news agency wishes to cover a set of events $E_1, E_2, \ldots, E_n$ which take place within the coming week starting at a prescribed time T_i. For each event E_i its duration time t_i and geographical site O_i is known. The news agency wishes to cover each of these events by having one reporter present for the full duration of the event. At the same time it wishes to use as few reporters as possible. Assuming that the travel time t_{ij} from O_i to O_j is known for each $1 \leq i, j \leq n$, we can model this problem as follows. Form a digraph $D = (V, A)$ by letting $V = \{v_1, v_2, \ldots, v_n\}$ and for every choice of $i \neq j$ put an arc from v_i to v_j if $T_j \geq T_i + t_i + t_{ij}$. It is easy to see that D is acyclic. Furthermore, if the events can be covered by k reporters then D has a k-path factor (just follow the routes travelled by the reporters). It is also easy to see that the converse also holds. Hence having an algorithm for the minimum path factor problem for acyclic digraphs will provide a solution to this and a large number of similar problems (such as airline and tanker scheduling, see Exercise 5.8).

Clearly, $\mathrm{pc}(D) = \mathrm{pcc}(D)$ for every acyclic digraph D. Using flows in networks, we can effectively find a $\mathrm{pcc}(D)$-path-cycle factor in any digraph D (see Exercises 3.59 and 3.7). Since a k-path-cycle factor in an acyclic digraph has no cycles, this implies that the minimum path factor problem for acyclic digraphs is easy (at least from an algorithmic point of view).

Theorem 5.3.1 *For acyclic digraphs the minimum path factor problem is solvable in time $O(\sqrt{n}m)$.* □

Another application of the path covering number of acyclic digraphs is for partial orders. A **partial order** consists of a set X and a binary relation '$\prec$' which is transitive (that is, $x \prec y, y \prec z$ implies $x \prec z$). Let $P = (X, \prec)$ be a partial order. Two elements $x, y \in X$ are **comparable** if either $x \prec y$ or $y \prec x$ holds. Otherwise x and y are **incomparable**. A **chain** in P is a totally ordered subset Y of X, that is, all elements in Y are pairwise comparable. An **antichain** on P is a subset Z of X, no two elements of which are comparable. Dilworth proved the following famous min-max result relating chains to antichains:

Theorem 5.3.2 (Dilworth's theorem) *[193] Let $P = (X, \prec)$ be a partial order. Then the minimum number of chains needed to cover X equals the maximum number of elements in an antichain.*

Proof: Given $P = (X, \prec)$, let $D = (X, A)$ be the digraph such that $xy \in A$ for $x \neq y \in X$ if and only if $x \prec y$. Clearly, D is transitive. Furthermore, a path (an independent set) in D corresponds to a chain (antichain) in P. We need to show that $\mathrm{pc}(D) = \alpha(D)$. By Theorem 5.2.1, $\mathrm{pc}(D) \leq \alpha(D)$. Let $\mathcal{F} = P_1 \cup P_2 \cup \ldots \cup P_k$ be a minimum path factor of D. By transitivity of D, each $V(P_i)$ induces a complete subgraph in $UG(D)$. Hence, $\alpha(D) = \alpha(UG(D)) \leq k = \mathrm{pc}(D)$. Thus, $\mathrm{pc}(D) = \alpha(D)$. □

The last theorem can obviously be reformulated as follows: $\alpha(D) = \mathrm{pc}(D)$ for every transitive oriented graph D. We conclude this section with an extension of the analogous result to extended semicomplete digraphs. Lemma 5.3.3 will be used in Section 6.11.

Lemma 5.3.3 *Let D be an acyclic extended semicomplete digraph with $\alpha(D) = k$, then the following holds:*

(a) $\mathrm{pc}(D) = k$.
(b) One can obtain a minimum path factor of D as follows: choose a longest path P in D, remove $V(P)$ and continue recursively.
(c) One can find a minimum path factor using the greedy algorithm in (b) in total time $O(n \log n)$ (using the adjacency matrix).

Proof: By Theorem 5.2.1 $\mathrm{pc}(D) \leq k$. On the other hand no path can contain two vertices from the same independent set as that would imply that D contains a cycle. Hence $\mathrm{pc}(D) = k$. To prove (b), let P be a longest path of D. By the argument above $\alpha(D - P) \geq k - 1$. On the other hand D can be written as $D = S[\overline{K}_{a_1}, \overline{K}_{a_2}, \ldots, \overline{K}_{a_s}]$, where S is a semicomplete digraph and $s = |V(S)|$. By Rédei's theorem (Theorem 1.4.5), S has a hamiltonian path P'. In D this path corresponds to a path Q which contains precisely one vertex from each maximal independent set. Hence Q is a longest path in D by the remark above and we have $\alpha(D - Q) = k - 1$. Now the second claim follows by induction on k. The third claim follows from the description of procedure MergeHamPathTour in Section 1.9.1, assuming that we have an adjacency matrix representation of D. Note that we delete the paths as we find them and hence the total complexity is still $O(n \log n)$. □

5.4 Hamilton Paths and Cycles in Path-Mergeable Digraphs

The class of path-mergeable digraphs was introduced in Section 4.9, where some of its properties were studied. In this section, we prove a characterization of hamiltonian path-mergeable digraphs due to Bang-Jensen [50].

We begin with a simple lemma which forms the basis for the proof of Theorem 5.4.2. For a cycle C, a ***C*-bypass** is a path of length at least two with both end-vertices on C and no other vertices on C.

Lemma 5.4.1 *[50] Let D be a path-mergeable digraph and let C be a cycle in D. If D has a C-bypass P, then there exists a cycle in D containing precisely the vertices $V(C) \cup V(P)$.*

Proof: Let P be an (x, y)-path. Then the paths P and $C[x, y]$ can be merged into one (x, y)-path R, which together with $C[y, x]$ forms the desired cycle. □

Theorem 5.4.2 (Bang-Jensen) *[50] A path-mergeable digraph D of order $n \geq 2$ is hamiltonian if and only if D is strong and $UG(D)$ is 2-connected.*

Proof: 'Only if' is obvious; we prove 'if'. Suppose that D is strong, $UG(D)$ is 2-connected and D is not hamiltonian. Let $C = u_1u_2\dots u_pu_1$ be a longest cycle in D. Observe that, by Lemma 5.4.1, there is no C-bypass. For each $i \in \{1,\dots,p\}$ let X_i (respectively Y_i) be the set of vertices of $D - V(C)$ that can be reached from u_i (respectively, from which u_i can be reached) by a path in $D - (V(C) - u_i)$. Since D is strong,

$$X_1 \cup \dots \cup X_p = Y_1 \cup \dots \cup Y_p = V(D) - V(C).$$

Since there is no C-bypass, every path starting at a vertex in X_i and ending at a vertex in C must end at u_i. Thus, $X_i \subseteq Y_i$. Similarly, $Y_i \subseteq X_i$ and, hence, $X_i = Y_i$. Since there is no C-bypass, the sets X_i are disjoint. Since we assumed that D is not hamiltonian, at least one of these sets, say X_1, is non-empty. Since $UG(D)$ is 2-connected, there is an arc with one end-vertex in X_1 and the other in $V(D) - (X_1 \cup u_1)$, and no matter what its orientation is, this implies that there is a C-bypass, a contradiction. □

Using the proof of this theorem, Lemma 5.4.1 and Proposition 4.9.3, it is not difficult to show the following (Exercise 5.10):

Corollary 5.4.3 *[50] There is an $O(nm)$-algorithm to decide whether a given strong path-mergeable digraph has a hamiltonian cycle and find one if it exists.*

Clearly, Theorem 5.4.2 and Corollary 5.4.3 imply an obvious characterization of longest cycles in path-mergeable digraphs and a polynomial algorithm to find a longest cycle. Neither a characterization nor the complexity of the hamiltonian path problem for path-mergeable digraphs is currently known. The following problem was posed by Bang-Jensen and Gutin:

Problem 5.4.4 *[65] Characterize traceable path-mergeable digraphs. Is there a polynomial algorithm to decide whether a path-mergeable digraph is traceable?*

For a related result, see Proposition 6.3.2. This result may be considered as a characterization of traceable path-mergeable digraphs. However, this characterization seems of not much value from the complexity point of view.

5.5 Hamilton Paths and Cycles in Locally In-Semicomplete Digraphs

According to Proposition 4.10.1, every locally in-semicomplete digraph is path-mergeable. By Exercise 5.12, every strong locally in-semicomplete digraph has a 2-connected underlying graph. Thus, Theorem 5.4.2 implies the following characterization of hamiltonian locally in-semicomplete digraphs[2].

Theorem 5.5.1 (Bang-Jensen, Huang and Prisner) *[81] A locally in-semicomplete digraph D of order $n \geq 2$ is hamiltonian if and only if D is strong.* □

This theorem generalizes Camion's theorem on strong tournaments (Theorem 1.5.2). Bang-Jensen and Hell [75] showed that for the class of locally in-semicomplete digraphs Corollary 5.4.3 can be improved to the following result.

Theorem 5.5.2 *[75] There is an $O(m + n \log n)$-algorithm for finding a hamiltonian cycle in a strong locally in-semicomplete digraph.*

In Section 5.4, we remarked that the Hamilton path problem for path-mergeable digraphs is unsolved so far. For a subclass of this class, locally in-semicomplete digraphs, an elegant characterization, due to Bang-Jensen, Huang and Prisner, exists.

Theorem 5.5.3 *[81] A locally in-semicomplete digraph is traceable if and only if it contains an in-branching.*

Proof: Since a Hamilton path is an in-branching, it suffices to show that every locally in-semicomplete digraph D with an in-branching T is traceable. We prove this claim by induction on the number b of vertices of T of in-degree zero.

For $b = 1$, the claim is trivial. Let $b \geq 2$. Consider a pair of vertices x, y of in-degree zero in T. By the definition of an in-branching there is a vertex z in T such that T contains both (x, z)-path P and (y, z)-path Q. Assume that the only common vertex of P and Q is z.

By Proposition 4.10.2, there is a path R in D that starts at x or y and terminates at z and $V(R) = V(P) \cup V(Q)$. Using this path, we may replace T with an in-branching with $b - 1$ vertices of in-degree zero and apply the induction hypothesis of the claim. □

Clearly, Theorem 5.5.3 implies that a locally out-semicomplete digraph is traceable if and only if it contains an out-branching. By Proposition 1.6.1, we have the following:

Corollary 5.5.4 *A locally in-semicomplete digraph is traceable if and only if it contains only one terminal strong component.* □

[2] Actually, this characterization, as well as the other results of this section, were originally proved only for oriented graphs. However, as can be seen from Exercises 4.27 and 4.28, the results for oriented graphs immediately imply the results of this section.

Using Corollary 5.5.4, Bang-Jensen and Hell [75] proved the following:

Theorem 5.5.5 *A longest path in a locally in-semicomplete digraph D can be found in time $O(m + n \log n)$.* □

Corollary 5.5.4 and Lemma 4.10.3 imply the following:

Corollary 5.5.6 (Bang-Jensen) *[44] A locally semicomplete digraph has a hamiltonian path if and only if it is connected.* □

Notice that there is a nice direct proof of this corollary (using Proposition 4.10.2), which is analogous to the classical proof of Rédei's theorem displayed in procedure HamPathTour in Section 1.9.1. See Exercise 5.14.

5.6 Hamilton Cycles and Paths in Degree-Constrained Digraphs

In Subsection 5.6.1 we formulate certain sufficient degree-constrained conditions for hamiltonicity of digraphs. Several of these conditions do not follow from the others, i.e. there are certain digraphs that can be proved to be hamiltonian using some condition but none of the others. (The reader will be asked to show this in the exercises.)

In Subsection 5.6.3 we provide proofs to some of these conditions to illustrate the power of a recently introduced approach, which we call the **multi-insertion technique**. (This technique can be traced back to Ainouche [9] for undirected graphs and to Bang-Jensen [48] for digraphs, see also the paper [68] by Bang-Jensen, Gutin and Huang). The technique itself is introduced in Subsection 5.6.2. The strength of the multi-insertion technique lies in the fact that we can prove the existence of a hamiltonian cycle without actually exhibiting it. Moreover, our hamiltonian cycles may have quite a complicated structure. For example, compare the hamiltonian cycles in the proof of Theorem 5.6.1 to the hamiltonian paths constructed in the inductive proof of Theorem 1.4.5. The multi-insertion technique is used in some other parts of this book, see e.g. Section 5.7.

Let x, y be a pair of distinct vertices in a digraph D. The pair $\{x, y\}$ is **dominated by a vertex** z if $z \rightarrow x$ and $z \rightarrow y$; in this case we say that the pair $\{x, y\}$ is **dominated**. Likewise, $\{x, y\}$ **dominates a vertex** z if $x \rightarrow z$ and $y \rightarrow z$; we call the pair $\{x, y\}$ **dominating**.

5.6.1 Sufficient Conditions

Considering the converse digraph and using Theorem 5.5.1, we see that a locally out-semicomplete digraph is hamiltonian if and only if it is strong. This can be generalized as follows. We prove Theorem 5.6.1 in Subsection 5.6.3.

Theorem 5.6.1 (Bang-Jensen, Gutin and Li) *[69] Let D be a strong digraph of order $n \geq 2$. Suppose that, for every dominated pair of non-adjacent vertices $\{x, y\}$, either $d(x) \geq n$ and $d(y) \geq n-1$ or $d(x) \geq n-1$ and $d(y) \geq n$. Then D is hamiltonian.*

The following example shows the sharpness of the conditions of Theorem 5.6.1 (and Theorem 5.6.5), see Figure 5.3. Let G and H be two disjoint transitive tournaments such that $|V(G)| \geq 2, |V(H)| \geq 2$. Let w be the vertex of out-degree 0 in G and w' the vertex of in-degree 0 in H. Form a new digraph by identifying w and w' to one vertex z. Add four new vertices x, y, u, v and the arcs $\{xv, yv, ux, uy\} \cup \{xz, zx, yz, zy\} \cup \{rg : \ r \in \{x, y, v\}, g \in V(G) - w\} \cup \{hs : \ h \in V(H) - w', s \in \{u, x, y\}\}$. Denote the resulting digraph by Q_n, where n is the order of Q_n. It is easy to check that Q_n is strong and non-hamiltonian (Exercise 5.17). Also x, y is the only pair of non-adjacent vertices which is dominating (dominated, respectively). An easy computation shows that

$$d(x) = d(y) = n - 1 = d^+(x) + d^-(y) = d^-(x) + d^+(y).$$

v y x u

z

$G - w$ $H - w'$

Figure 5.3 The digraph Q_n. The two unoriented edges denote 2-cycles.

Combining Theorem 5.6.1 with Proposition 5.0.3 one can obtain sufficient conditions for a digraph to be traceable (see also Exercise 5.16). Theorem 5.6.1 also has the following immediate corollaries.

Corollary 5.6.2 (Ghouila-Houri) *[315] If the degree of every vertex in a strong digraph D of order n is at least n, then D is hamiltonian.* □

Corollary 5.6.3 *Let D be a digraph of order n. If the minimum semi-degree of D, $\delta^0(D) \geq n/2$, then D is hamiltonian.* □

It turns out that even a slight relaxation of Corollary 5.6.3 brings in non-hamiltonian digraphs. In particular, Darbinyan [177] proved the following:

Proposition 5.6.4 *Let D be a digraph of even order $n \geq 4$ such that the degree of every vertex of D is at least $n-1$ and $\delta^0(D) \geq n/2 - 1$. Then either D is hamiltonian or D belongs to a non-empty finite family of non-hamiltonian digraphs.* $\square$

By Theorem 5.5.1, a locally semicomplete digraph is hamiltonian if and only if it is strong [44]. This result was generalized by Bang-Jensen, Gutin and Li [69] as follows.

Theorem 5.6.5 *Let D be a strong digraph of order n. Suppose that D satisfies* $\min\{d^+(x) + d^-(y),\, d^-(x) + d^+(y)\} \geq n$ *for every pair of dominating non-adjacent and every pair of dominated non-adjacent vertices $\{x, y\}$. Then D is hamiltonian.*

We prove this theorem in Subsection 5.6.3. Theorem 5.6.5 implies Corollary 5.6.3 as well as the following theorem by Woodall [739]:

Corollary 5.6.6 *Let D be a strong digraph of order $n \geq 2$. If $d^+(x) + d^-(y) \geq n$ for all pairs of vertices x and y such that there is no arc from x to y, then D is hamiltonian.* $\square$

The following theorem generalizes Corollaries 5.6.2,5.6.3 and 5.6.6. The inequality of Theorem 5.6.7 is best possible: Consider $\overleftrightarrow{K}_{n-2}$ ($n \geq 5$) and fix a vertex u in this digraph. Construct the digraph H_n by adding to $\overleftrightarrow{K}_{n-2}$ a pair v, w of vertices such that both v and w dominates every vertex in $\overleftrightarrow{K}_{n-2}$ and are dominated by only u, see Figure 5.4. It is easy to see that H_n is strong and non-hamiltonian ($H_n - u$ is not traceable). However, v, w is the only pair of non-adjacent vertices in H_n and $d(v) + d(w) = 2n - 2$.

w v

u

$\overleftrightarrow{K}_{n-3}$

Figure 5.4 The digraph H_n.

Theorem 5.6.7 (Meyniel's theorem) *[564] Let D be a strong digraph of order $n \geq 2$. If $d(x) + d(y) \geq 2n - 1$ for all pairs of non-adjacent vertices in D, then D is hamiltonian.* $\square$

Short proofs of Meyniel's theorem were given by Overbeck-Larisch [597] and Bondy and Thomassen [128]. The second proof is slightly simpler than the first one and can also be found in the book [735] by West (see Theorem 8.4.38). Using Proposition 5.0.3 one can easily see that replacing $2n-1$ by $2n-3$ in Meyniel's theorem we obtain sufficient conditions for traceability. (Note that for traceability we do not require strong connectivity.) Darbinyan [180] proved that by weakening the degree condition in Meyniel's theorem only by one, we obtain a stronger result:

Theorem 5.6.8 *[180] Let D be a digraph of order $n \geq 3$. If $d(x) + d(y) \geq 2n-2$ for all pairs of non-adjacent vertices in D, then D contains a hamiltonian path in which the initial vertex dominates the terminal vertex.* □

Berman and Liu [111] extended Theorem 5.6.7 as formulated below. For a digraph D of order n, a set $M \subseteq V(D)$ is **Meyniel** if $d(x)+d(y) \geq 2n-1$ for every pair x, y of non-adjacent vertices in M. The proof of Theorem 5.6.9 in [111] is based on the multi-insertion technique.

Theorem 5.6.9 *[111] Let M be a Meyniel set of vertices of a strong digraph D of order $n \geq 2$. Then D has a cycle containing all vertices of M.* □

Another extension of Meyniel's theorem was given by Heydemann [428].

Theorem 5.6.10 *[428] Let h be a non-negative integer and let D be a strong digraph of order $n \geq 2$ such that, for every pair of non-adjacent vertices x and y, we have $d(x)+d(y) \geq 2n-2h+1$. Then D contains a cycle of length greater than or equal to $\lceil \frac{n-1}{h+1} \rceil + 1$.* □

Manoussakis [547] proved the following sufficient condition that involves triples rather than pairs of vertices. Notice that Theorem 5.6.11 does not imply either of Theorems 5.6.1, 5.6.5 and 5.6.7 [69].

Theorem 5.6.11 *[547] Suppose that a strong digraph D of order $n \geq 2$ satisfies the following conditions: for every triple $x, y, z \in V(D)$ such that x and y are non-adjacent*

(a) If there is no arc from x to z, then $d(x)+d(y)+d^+(x)+d^-(z) \geq 3n-2$.
(b) If there is no arc from z to x, then $d(x)+d(y)+d^-(x)+d^+(z) \geq 3n-2$.

Then D is hamiltonian. □

The next theorem resembles both Theorem 5.6.5 and Theorem 5.6.7. However, Theorem 5.6.12 does not imply any of these theorems. The sharpness of the inequality of Theorem 5.6.12 can be seen from the digraph H_n introduced before Theorem 5.6.7.

Theorem 5.6.12 (Zhao and Meng) *[758] Let D be a strong digraph of order $n \geq 2$. If*

$$d^+(x) + d^+(y) + d^-(u) + d^-(v) \geq 2n - 1$$

for every pair x, y of dominating vertices and every pair u, v of dominated vertices, then D is hamiltonian. □

Theorems 5.6.5 and 5.6.12 suggest that the following conjecture by Bang-Jensen, Gutin and Li, may be true.

Conjecture 5.6.13 *[69] Let D be a strong digraph of order $n \geq 2$. Suppose that $d(x)+d(y) \geq 2n-1$ for every pair of dominating non-adjacent and every pair of dominated non-adjacent vertices $\{x, y\}$. Then D is hamiltonian.*

Bang-Jensen, Guo and Yeo [57] proved that, if we replace the degree condition $d(x) + d(y) \geq 2n - 1$ with $d(x) + d(y) \geq \frac{5}{2}n - 4$ in Conjecture 5.6.13, then D is hamiltonian. They also provided additional support for Conjecture 5.6.13 by showing that every digraph satisfying the condition of Conjecture 5.6.13 has a cycle factor.

Perhaps Conjecture 5.6.13 can even be generalized to the following which was conjectured by Bang-Jensen, Gutin and Li:

Conjecture 5.6.14 *[69] Let D be a strong digraph of order $n \geq 2$. Suppose that, for every pair of dominated non-adjacent vertices $\{x, y\}$, $d(x) + d(y) \geq 2n - 1$. Then D is hamiltonian.*

Let F be the digraph obtained from the complete digraph $\overleftrightarrow{K}_{n-3}$ by adding three new vertices $\{x, y, z\}$ and the following arcs $\{xy, yx, yz, zy, zx\} \cup \{xu, ux, yu : u \in V(\overleftrightarrow{K}_{n-3})\}$, see Figure 5.5. Clearly F is strongly connected and the underlying undirected graph of F is 2-connected. However, F is not hamiltonian as all hamiltonian paths in $F - x$ start at z, but x does not dominate z. The only pairs of non-adjacent vertices in D are z and any vertex $u \in V(\overleftrightarrow{K}_{n-3})$ and here we have $d(z) + d(u) = 2n - 2$. Thus both conjectures above would be the best possible.

One of the oldest conjectures in the area of hamiltonian digraphs is the following conjecture by Nash-Williams.

Conjecture 5.6.15 *[586, 587] Let D be a digraph of order $n \geq 3$ satisfying the following conditions:*

(i) For every positive integer k less than $(n-1)/2$, the number of vertices of out-degree less than or equal to k is less than k.
(ii) The number of vertices of out-degree less than or equal to $(n-1)/2$ is less than or equal to $(n-1)/2$.

z

x y

$\overleftrightarrow{K}_{n-3}$

Figure 5.5 The digraph F.

(iii) For every positive integer k less than $(n-1)/2$, the number of vertices of in-degree less than or equal to k is less than k.
(iv) The number of vertices of in-degree less than or equal to $(n-1)/2$ is less than or equal to $(n-1)/2$.

Then D is hamiltonian.

Conjecture 5.6.15 seems to be very difficult (see comments by Nash-Williams in [587, 588]). This conjecture was inspired by the corresponding theorem by Pósa [610] on undirected graphs. Pósa's result implies that the assertion of this conjecture is true at least for symmetric digraphs, i.e. digraphs D such that $xy \in A(D)$ implies $yx \in A(D)$.

One may also try to obtain digraph analogues of various other sufficient degree conditions for graphs, such as Chvátal's theorem [159], which asserts that, if the degree sequence $d_1 \leq d_2 \leq \ldots \leq d_n$ of an undirected graph satisfies the condition $d_k \leq k < \frac{n}{2} \Rightarrow d_{n-k} \geq n-k$ for each k, then the graph is hamiltonian. Similarly, one may ask whether every strong digraph whose non-decreasing degree sequence $d_1 \leq d_2 \leq \ldots \leq d_n$ satisfies the following condition is hamiltonian:

$$d_k \leq 2k < n \Rightarrow d_{n-k} \geq 2(n-k), \;\; k = 1, 2, \ldots, n-1. \tag{5.1}$$

For a digraph D we can obtain the non-decreasing out-degree and in-degree sequences: $d_1^+ \leq d_2^+ \leq \ldots \leq d_n^+$ and $d_1^- \leq d_2^- \leq \ldots \leq d_n^-$ (orderings of vertices of D in these two sequences are usually different). Using the two sequences, one may suggest conditions similar to (5.1):

$$d_k^+ \leq k < \frac{n}{2} \Rightarrow d_{n-k}^+ \geq n-k \text{ and} \tag{5.2}$$

$$d^-_k \le k < \frac{n}{2} \Rightarrow d^-_{n-k} \ge n-k,\ 1 \le k \le (n-1)/2.$$

It is not difficult to construct an infinite family of non-hamiltonian strong digraphs that satisfy both (5.1) and (5.2) (Exercise 5.25). However, if we 'mix' the out-degrees with the in-degrees in (5.2), we obtain the following conjecture due to Nash-Williams:

Conjecture 5.6.16 *[588] If the non-decreasing out-degree and in-degree sequences of a strong digraph D satisfy the conditions*

$$d^+_k \le k < \frac{n}{2} \Rightarrow d^-_{n-k} \ge n-k \text{ and}$$
$$d^-_k \le k < \frac{n}{2} \Rightarrow d^+_{n-k} \ge n-k,\ 1 \le k \le (n-1)/2,$$

then D is hamiltonian.

One may expect that for oriented graphs (i.e., digraphs with no 2-cycles) a result much stronger than Corollary 5.6.3 holds. Häggkvist [387] proved the following theorem and made a much stronger conjecture. Notice that Häggkvist [387] constructed non-hamiltonian oriented graphs D with $\delta^0(D) \ge n/3$ (these oriented graphs do not even contain cycle factors).

Theorem 5.6.17 *[387] Let D be an oriented graph of order n and let $\delta^0(D) \ge (\frac{1}{2} - 2^{-18})n$. Then D is hamiltonian.* □

Conjecture 5.6.18 *[387] Let D be an oriented graph of order n and let $\delta^+(D) \ge (3n-2)/8$. Then D is hamiltonian.*

Jackson conjectured that for regular oriented graphs an even stronger assertion holds.

Conjecture 5.6.19 *[449] Every k-regular oriented graph of order at most $4k+1$, where $k \neq 2$, contains a Hamilton cycle.*

5.6.2 The Multi-Insertion Technique

Let $P = u_1u_2 \dots u_s$ be a path in a digraph D and let $Q = v_1v_2 \dots v_t$ be a path in $D - V(P)$. The path P **can be inserted into** Q if there is a subscript $i \in \{1, 2, \dots, t-1\}$ such that $v_i \rightarrow u_1$ and $u_s \rightarrow v_{i+1}$. Indeed, in this case the path Q can be extended to a new (v_1, v_t)-path $Q[v_1, v_i]PQ[v_{i+1}, v_t]$. The path P **can be multi-inserted into** Q if there are integers $i_1 = 1 < i_2 < \dots < i_m = s+1$ such that, for every $k = 2, 3, \dots, m$, the subpath $P[u_{i_{k-1}}, u_{i_k - 1}]$ can be inserted into Q. The sequence of subpaths $P[u_{i_{k-1}}, u_{i_k-1}]$, $k = 2, \dots, m$, is a **multi-insertion partition** of P. Similar definitions can be given for the case when Q is a cycle.

The complexity of algorithms in this subsection is measured in terms of the number of queries to the adjacency matrix of a digraph. In this subsection we prove several simple results, which are very useful while applying the multi-insertion technique. Some of these results are used in this section, others will be applied in other parts of this book. The following lemma is a simple extension of a lemma by Bang-Jensen, Gutin and Li [69].

Lemma 5.6.20 *Let P be a path in D and let $Q = v_1v_2\dots v_t$ be a path (a cycle, respectively) in $D - V(P)$. If P can be multi-inserted into Q, then there is a (v_1, v_t)-path R (a cycle, respectively) in D so that $V(R) = V(P) \cup V(Q)$. Given a multi-insertion partition of P, the path R can be found in time $O(|V(P)||V(Q)|)$.*

Proof: We consider only the case when Q is a path, as the other case (Q is a cycle) can be proved analogously. Let $P = u_1u_2\dots u_s$. Suppose that integers $i_1 = 1 < i_2 < \dots < i_m = s+1$ are such that the subpaths $P[u_{i_{k-1}}, u_{i_k - 1}]$, $k = 2, 3, \dots, m$, form a multi-insertion partition of P.

We proceed by induction on m. If $m = 2$ then the claim is obvious, hence assume that $m \geq 3$. Let $xy \in A(Q)$ be such that the subpath $P[u_{i_1}, u_{i_2 - 1}]$ can be inserted between x and y on Q. Choose r as large as possible such that $u_{i_r - 1} \rightarrow y$. Clearly, $P[u_{i_1}, u_{i_r - 1}]$ can be inserted into Q to give a (v_1, v_t)-path Q^*. Thus, if $r = m$ we are done. Otherwise apply the induction hypothesis to the paths $P[u_{i_r}, u_s]$ and Q^* (observe that by the choice of r none of the subpaths of the multi-insertion partition of $P[u_{i_r}, u_s]$ can be inserted between x and y in Q, and thus every such subpath can be inserted into Q^*).

If we postpone the actual construction of R till we have found a new multi-insertion partition $\mathcal{M}$ of P and all (distinct) pairs of vertices between which the subpaths of $\mathcal{M}$ can be inserted, then the complexity claim of this lemma follows easily. $\square$

The next two corollaries due to Bang-Jensen, Gutin and Huang, respectively, Yeo can easily be proved using Lemma 5.6.20; their proofs are left as an easy exercise (Exercise 5.21).

Corollary 5.6.21 *[68] Let D be a digraph. Suppose that $P = u_1u_2\dots u_r$ is a path in D and C is a cycle in $D - P$. Suppose that for each $i = 1, 2, \dots, r-1$, either the arc u_iu_{i+1} or the vertex u_i can be inserted into C, and, in addition, assume that u_r can be inserted into C. Then D contains a cycle Z with the vertex set $V(P) \cup V(C)$ and Z can be constructed in time $O(|V(P)||V(C)|)$.* $\square$

Corollary 5.6.22 *[744] Let D be a digraph. Suppose that $P = u_1u_2\dots u_r$ is a path in D and C is a cycle in $D - P$. Suppose also that for each odd index i the arc u_iu_{i+1} can be inserted into C, and if r is odd, u_r can be inserted* *into C. Then D contains a cycle Z with the vertex set $V(P) \cup V(C)$ and Z can be constructed in time $O(|V(P)||V(C)|)$.* □

Corollary 5.6.23 *[68] Let D be a digraph. Suppose that C is a cycle of even length in D and Q is a cycle in $D - C$. Suppose also that for each arc uv of C either the arc uv or the vertex u can be inserted into Q. Then D contains a cycle Z with the vertex set $V(Q) \cup V(C)$ and Z can be constructed in time $O(|V(Q)||V(C)|)$.*

Proof: If there is a vertex x on C that can be inserted into Q then apply Corollary 5.6.21 to $C[x^+, x]$ and Q. Otherwise, all the arcs of C can be inserted into Q and we can apply Corollary 5.6.22 to $C[y^+, y]$ and Q, where y is any vertex of C. □

5.6.3 Proofs of Theorems 5.6.1 and 5.6.5

The following lemma is a slight modification of a lemma by Bondy and Thomassen [128]; its proof is not too difficult and is left as an exercise to the reader (Exercise 5.18).

Lemma 5.6.24 *Let $Q = v_1 v_2 \dots v_t$ be a path in D and let w, w' be vertices of $V(D) - V(Q)$ (possibly $w = w'$). If there do not exist consecutive vertices v_i, v_{i+1} on Q such that $v_i w, w' v_{i+1}$ are arcs of D, then $d^-_Q(w) + d^+_Q(w') \leq t + \xi$, where $\xi = 1$ if $v_t \rightarrow w$ and 0, otherwise.* □

In the special case when $w' = w$ above, we get the following interpretation of the statement of Lemma 5.6.24.

Lemma 5.6.25 *Let $Q = v_1 v_2 \dots v_t$ be a path in D, and let $w \in V(D) - V(Q)$. If w cannot be inserted into Q, then $d_Q(w) \leq t + 1$. If, in addition, v_t does not dominate w, then $d_Q(w) \leq t$.* □

Let C be a cycle in D. Recall that an (x, y)-path P is a C-bypass if $|V(P)| \geq 3$, $x \neq y$ and $V(P) \cap V(C) = \{x, y\}$. The length of the path $C[x, y]$ is the **gap of P with respect to C**.

Proof of Theorem 5.6.1: Assume that D is non-hamiltonian and $C = x_1 x_2 \dots x_m x_1$ is a longest cycle in D. We first show that D contains a C-bypass. Assume D does not have one. Since D is strong, D must contain a cycle Z such that $|V(Z) \cap V(C)| = 1$. Without loss of generality, we may assume that $V(Z) \cap V(C) = \{x_1\}$. Let z be the successor of x_1 on Z. Since D has no C-bypass, z and x_2 are non-adjacent. Since z and x_2 are a dominated pair, $d(z) + d(x_2) \geq 2n - 1$. On the other hand, since D has no C-bypass, we have $d_{C-x_1}(z) = d_{Z-x_1}(x_2) = 0$ and $|(\{z, x_2\}, y) \cup (y, \{z, x_2\})| \leq 2$ for every $y \in V(D) - (V(C) \cup V(Z))$. Thus, $d(z) + d(x_2) \leq 2(n-1)$; a contradiction.

Let $P = u_1u_2 \dots u_s$ be a C-bypass ($s \geq 3$). Without loss of generality, let $u_1 = x_1,\ u_s = x_{\gamma+1},\ 0 < \gamma < m$. Suppose also that the gap γ of P is minimum among the gaps of all C-bypasses.

Since C is a longest cycle of D, $\gamma \geq 2$. Let $C' = C[x_2, x_\gamma]$, $C'' = C[x_{\gamma+1}, x_1]$, $R = D - V(C)$, and let x_j be any vertex in C' such that $x_1 \rightarrow x_j$. Let also x_k be an arbitrary vertex in C'.

We first prove that

$$d_{C''}(x_j) \geq |V(C'')| + 2. \tag{5.3}$$

Since C is a longest cycle and P has the minimum gap with respect to C, u_2 is not adjacent to any vertex on C', and there is no vertex $y \in V(R) - \{u_2\}$ such that either $u_2 \rightarrow y \rightarrow x_k$ or $x_k \rightarrow y \rightarrow u_2$. Therefore,

$$d_{C'}(x_k) + d_{C'}(u_2) \leq 2(|V(C')| - 1) \tag{5.4}$$

and

$$d_R(x_k) + d_R(u_2) \leq 2(n - m - 1). \tag{5.5}$$

By the maximality of C, u_2 cannot be inserted into C'', so by Lemma 5.6.25,

$$d_{C''}(u_2) \leq |V(C'')| + 1. \tag{5.6}$$

The fact that the pair of non-adjacent vertices $\{x_j, u_2\}$ is dominated by x_1 along with (5.4), (5.5) and (5.6), implies that

$$2n - 1 \leq d(x_j) + d(u_2) \leq d_{C''}(x_j) + 2n - |V(C'')| - 3.$$

This implies (5.3).

By (5.3) and Lemma 5.6.25, x_2 can be inserted into C''. Since C is a longest cycle, it follows from Lemma 5.6.20 that there exists $\beta \in \{3, \dots, \gamma\}$ so that the subpath $C[x_2, x_{\beta-1}]$ can be multi-inserted into C'', but $C[x_2, x_\beta]$ cannot. In particular, x_β cannot be inserted into C''. Thus, by (5.3) and Lemma 5.6.25, x_1 does not dominate x_β and $d_{C''}(x_\beta) \leq |V(C'')|$. This along with (5.4)-(5.6) gives $d(x_\beta) + d(u_2) \leq 2n - 3$. Since u_2 forms a dominated pair with x_2, we have that $d(u_2) \geq n - 1$. Hence,

$$d(x_\beta) \leq n - 2. \tag{5.7}$$

By the definition of multi-insertion, there are $\alpha \in \{2, 3, \dots, \beta - 1\}$ and $i \in \{\gamma + 1, \dots, m\}$ such that $x_i \rightarrow x_\alpha$ and $x_{\beta-1} \rightarrow x_{i+1}$. Observe that the pair $\{x_\beta, x_{i+1}\}$ is dominated by $x_{\beta-1}$. Thus, by (5.7) and the assumption of the theorem, either $x_\beta \rightarrow x_{i+1}$ or $x_{i+1} \rightarrow x_\beta$. If $x_\beta \rightarrow x_{i+1}$, then the path $P[x_2, x_\beta]$ can be multi-inserted into C'' which contradicts our assumption. Hence, $x_{i+1} \rightarrow x_\beta$. Considering the pair x_β, x_{i+2}, we conclude analogously that

$x_{i+2} \rightarrow x_\beta$. Continuing this process, we finally conclude that $x_1 \rightarrow x_\beta$, contradicting the conclusion above that the arc $x_1 x_\beta$ does not exist. □

Proof of Theorem 5.6.5: Assume that D is not hamiltonian and $C = x_1 x_2 \ldots x_m x_1$ is a longest cycle in D. Set $R = D - V(C)$. We first prove that D has a C-bypass with 3 vertices.

Since D is strong, there is a vertex y in R and a vertex x in C such that $y \rightarrow x$. If y dominates every vertex on C, then C is not a longest cycle, since a path P from a vertex x_i on C to y such that $V(P) \cap V(C) = \{x_i\}$ together with the arc $y \rightarrow x_{i+1}$ and the path $C[x_{i+1}, x_i]$ form a longer cycle in D. Hence, either there exists a vertex $x_r \in V(C)$ such that $x_r \rightarrow y \rightarrow x_{r+1}$, in which case we have the desired bypass, or there exists a vertex $x_j \in V(C)$ so that y and x_j are non-adjacent, but $y \rightarrow x_{j+1}$. Since the pair $\{y, x_j\}$ dominates x_{j+1}, $d^+(x_j) + d^-(y) \geq n$. This implies the existence of a vertex $z \in V(D) - \{x_j, x_{j+1}, y\}$ such that $x_j \rightarrow z \rightarrow y$. Since C is a longest cycle, $z \in V(C)$. So, $B = zyx_{j+1}$ is the desired bypass.

Without loss of generality, assume that $z = x_1$ and the gap j of B with respect to C is minimum among the gaps of all C-bypasses with three vertices. Clearly, $j \geq 2$.

Let $C' = C[x_2, x_j]$ and $C'' = C[x_{j+1}, x_1]$. Since C is a longest cycle, C' cannot be multi-inserted into C''. It follows from Lemma 5.6.24 that $d^+_{C''}(x_j) + d^-_{C''}(x_2) \leq |V(C'')| + 1$. By Lemma 5.6.25 and the maximality of C, $d_{C''}(y) \leq |V(C'')| + 1$. Analogously to the way we derived (5.4) in the previous proof, we get that $d_R(y) + d^+_R(x_j) + d^-_R(x_2) \leq 2(n - m - 1)$. Clearly, $d^+_{C'}(x_j) + d^-_{C'}(x_2) \leq 2|V(C')| - 2$. Since $d_{C'}(y) = 0$, the last four inequalities imply

$$d(y) + d^+(x_j) + d^-(x_2) \leq 2n - 2. \tag{5.8}$$

Since y is adjacent to neither x_2 nor x_j, the assumption of the theorem implies that $d^+(y) + d^-(x_2) \geq n$ and $d^-(y) + d^+(x_j) \geq n$, which contradicts (5.8). □

5.7 Longest Paths and Cycles in Semicomplete Multipartite Digraphs

While both Hamilton path and Hamilton cycle problems are polynomial time solvable for semicomplete multipartite digraphs (the latter was a difficult open problem for a while and was proved recently by Bang-Jensen, Gutin and Yeo [72] using several deep results on cycles and paths in semicomplete multipartite digraphs, see also [746]), only a characterization of traceable semicomplete multipartite digraphs is known. In Subsection 5.7.1, we give basic results on hamiltonian and longest paths and cycles in semicomplete multipartite digraphs. Several results of Subsection 5.7.1 are proved in Subsection

5.7.3 using the most important assertion of Subsection 5.7.2. In Subsection 5.7.4, we formulate perhaps the most important known result on Hamilton cycles in semicomplete multipartite digraphs, Yeo's Irreducible Cycle Subdigraph Theorem, and prove some interesting consequences of this powerful result. Due to the space limit our treatment of hamiltonian semicomplete multipartite digraphs is certainly restricted. The reader can find more information on the topic in the survey papers [65, 66] by Bang-Jensen and Gutin [368] by Gutin and [728] by Volkmann, the theses [345, 362, 692, 745], by Guo, Gutin, Tewes and Yeo respectively and the papers cited there.

5.7.1 Basic Results

We start by considering the longest path problem for semicomplete multipartite digraphs. The following characterization is proved in Subsection 5.7.3.

Theorem 5.7.1 (Gutin) *[358, 363] A semicomplete multipartite digraph D is traceable if and only if it contains a 1-path-cycle factor. One can verify whether D is traceable and find a hamiltonian path in D (if any) in time $O(n^{2.5})$.*

This theorem can be reformulated as $\mathrm{pc}(D) = 1$ if and only if $\mathrm{pcc}(D) = 1$ for a semicomplete multipartite digraph D. Using the result of Exercise 3.59, the last statement can be easily extended to the following result by Gutin:

Theorem 5.7.2 *[362] For a semicomplete multipartite digraph D, $\mathrm{pc}(D) = \mathrm{pcc}(D)$. The path covering number of D can be found in time $O(n^{2.5})$.* □

The non-complexity part of the next result by Gutin follows from Theorem 5.7.1. The complexity part is a simple consequence of Theorem 3.11.11.

Theorem 5.7.3 *[363] Let D be a semicomplete multipartite digraph of order n.*

(a) Let $\mathcal{F}$ be a 1-path-cycle subdigraph with maximum number of vertices in D. Then D contains a path P such that $V(P) = V(\mathcal{F})$.
(b) A longest path in D can be constructed in time $O(n^3)$.

□

We see from Theorem 5.7.1 that the hamiltonian path problem for semicomplete multipartite digraphs turns out to be relatively simple. The hamiltonian cycle problem for this class of digraphs seems to be much more difficult. One could guess that similarly to Theorem 5.7.1, a semicomplete multipartite digraph is hamiltonian if and only if it is strong and has a cycle factor. Even though these two conditions (strong connectivity and the existence of a cycle factor) are sufficient for semicomplete bipartite digraphs and extended semicomplete digraphs (see Theorems 5.7.4 and 5.7.5), they are not sufficient for semicomplete k-partite digraphs ($k \geq 3$) (see, e.g., an example later in this subsection). The following characterization was obtained independently by Gutin [353] and Häggkvist and Manoussakis [389].

Theorem 5.7.4 *A semicomplete bipartite digraph D is hamiltonian if and only if D is strong and contains a cycle factor. One can check whether D is hamiltonian and construct a Hamilton cycle of D (if one exists) in time $O(n^{2.5})$.*

Some sufficient conditions for the existence of a hamiltonian cycle in a bipartite tournament are described in the survey paper [368] by Gutin.

Theorem 5.7.5 *[359] An extended semicomplete digraph D is hamiltonian if and only if D is strong and contains a cycle factor. One can check whether D is hamiltonian and construct a Hamilton cycle of D (if one exists) in time $O(n^{2.5})$.*

These two theorems can be generalized as follows.

Theorem 5.7.6 (Gutin) *[357, 362] Let D be strong semicomplete bipartite digraph. The length of a longest cycle in D is equal to the number of vertices in a cycle subdigraph of D of maximum order. One can find a longest cycle in D in time $O(n^3)$.*

Theorem 5.7.7 *[362] Let D be a strong extended semicomplete digraph and let $\mathcal{F}$ be a cycle subdigraph of D. Then D has a cycle C which contains all vertices of $\mathcal{F}$. The cycle C can be found in time $O(n^3)$. In particular, if $\mathcal{F}$ is maximum, then $V(C) = V(\mathcal{F})$, i.e., C is a longest cycle of D.*

Proofs of the last two theorems are given in Subsection 5.7.3. One can see that the statement of Theorem 5.7.7 is stronger than Theorem 5.7.6. In fact, the analogue of Theorem 5.7.7 for semicomplete bipartite digraphs does not hold [362], see Exercise 5.29. The following strengthening of Theorem 5.7.7 is proved in [82]:

Theorem 5.7.8 (Bang-Jensen, Huang and Yeo) *[82] Let $D = (V, A)$ be a strong extended semicomplete digraph with decomposition given by $D = [H_1, H_2, \ldots, H_s]$, where $s = |S|$ and every $V(H_i)$ is a maximal independent set in V. Let m_i, $i = 1, 2, \ldots, s$, denote the maximum number of vertices from H_i which are contained in a cycle subdigraph of D. Then every longest cycle of D contains precisely m_i vertices from each H_i, $i = 1, 2, \ldots, t$.* □

One may ask whether there is any degree of strong connectivity, which together with a cycle factor is sufficient to guarantee a hamiltonian cycle in a semicomplete multipartite digraph (or a multipartite tournament). The answer is negative. In fact, there is no s such that every s-strong multipartite tournament with a cycle factor has a Hamilton cycle. Figure 5.6 shows a non-hamiltonian multipartite tournament T which is s-strong (s is the number of vertices in each of the sets A, B, C, D and X, Y, Z), and has a cycle factor. We leave it to the reader to verify that there is no Hamilton cycle in T (Exercise 5.28).

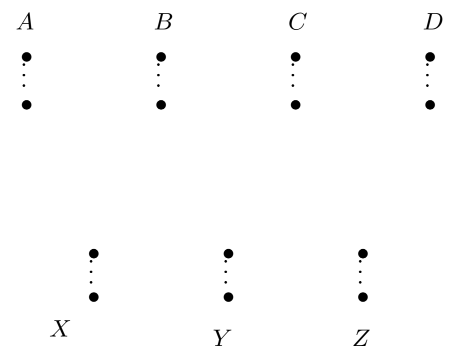


Figure 5.6 An s-strong non-hamiltonian multipartite tournament T with a cycle factor. Each of the sets A, B, C, D and X, Y, Z induces an independent set with exactly s vertices. All arcs between two sets have the direction shown.

We conclude the description of basic results on hamiltonian semicomplete digraphs by the following important result which we mentioned above.

Theorem 5.7.9 (Bang-Jensen, Gutin and Yeo) *[72] One can verify whether a semicomplete multipartite digraph D has a hamiltonian cycle and find one (if it exists) in time $O(n^7)$.* $\square$

Very recently Yeo [746] proved that the problem can be solved in time $O(n^5)$.

5.7.2 The Good Cycle Factor Theorem

The purpose of this subsection, based on the paper [68] by Bang-Jensen, Gutin and Huang, is to prove some sufficient conditions for a semicomplete multipartite digraph to be hamiltonian.

Let $\mathcal{F} = C_1 \cup C_2$ be a cycle factor or a 1-path-cycle factor in a digraph D, where C_1 is a cycle or a path in D and C_2 is a cycle. A vertex $v \in V(C_i)$ is called **out-singular** (**in-singular**) with respect to C_{3-i} if $v \Rightarrow C_{3-i}$ ($C_{3-i} \Rightarrow v$); v is **singular** with respect to C_{3-i} if it is either out-singular or in-singular with respect to C_{3-i}.

Lemma 5.7.10 *[68] Let $Q \cup C$ be a cycle factor in a semicomplete multipartite digraph D. Suppose that the cycle Q has no singular vertices (with respect to C) and D has no hamiltonian cycle, then for every arc xy of Q either the arc xy itself can be inserted into C, or both vertices x and y can be inserted into C.*

Proof: Assume without loss of generality that there is some arc xy on Q such that neither x nor xy can be inserted into C. Since D is a semicomplete multipartite digraph and x is non-singular and cannot be inserted into C, there exists a vertex v on C which is not adjacent to x and $v^- \rightarrow x \rightarrow v^+$. Furthermore, v is adjacent to y since x and y are adjacent. Since xy cannot be inserted into C, we have $v \rightarrow y$. Then D contains a Hamilton cycle $Q[y,x]C[v^+,v]y$, which contradicts the assumption. □

Lemma 5.7.11 *[68] Let D be a semicomplete multipartite digraph containing a cycle factor $C_1 \cup C_2$ such that C_i has no singular vertices with respect to C_{3-i}, for both $i = 1, 2$; then D is hamiltonian. Given C_1 and C_2, a hamiltonian cycle in D can be found in time $O(|V(C_1)||V(C_2)|)$.*

Proof: If at least one of the cycles C_1, C_2 is even, then by Corollary 5.6.23 and Lemma 5.7.10 we can find a Hamilton cycle in D in time $O(|V(C_1)||V(C_2)|)$. Thus, assume that both of C_1, C_2 are odd cycles. If some vertex in C_i can be inserted into C_{3-i} for some $i = 1$ or 2, then by Corollary 5.6.21 and Lemma 5.7.10, we can construct a Hamilton cycle in D in time $O(|V(C_1)||V(C_2)|)$. Thus, we may also assume that no vertex in C_i can be inserted into C_{3-i} for both $i = 1, 2$. So, by Lemma 5.7.10, every arc of C_i can be inserted into C_{3-i}.

Now we show that either D is hamiltonian or we may assume that every arc of C_i can be inserted between two different pairs of vertices in C_{3-i} ($i = 1, 2$). Consider an arc x_1x_2 of C_1. Since both x_1 and x_2 are non-singular and cannot be inserted into C_2, there exist vertices v_1 and v_2 on C_2 such that v_i is not adjacent to x_i and $v_i^- \rightarrow x_i \rightarrow v_i^+$, $i = 1, 2$. If $v_1 \rightarrow x_2$, then we obtain a Hamilton cycle. So we may assume that the only arc between x_2 and v_1 is x_2v_1. For the same reason, we may assume that v_2 dominates x_1 but is not dominated by x_1. Now the arc x_1x_2 can be inserted between v_1^- and v_1 and between v_2 and v_2^+.

Hence, x_1x_2 cannot be inserted between two pairs of vertices only in the case that $v_1^- = v_2$ and $v_1 = v_2^+$. We show that in this case D is hamiltonian. Construct, at first, a cycle $C^* = C_1[x_2, x_1]C_2[v_1^+, v_2^-]x_2$ which contains all the vertices of D but v_1^-, v_1. The arc $v_1^-v_1$ can be inserted into C_1, by the remark at the beginning of the proof. But $v_1^-v_1$ cannot be inserted between x_1 and x_2, since v_1 does not dominate x_2 and $v_1^- = v_2$ is not dominated by x_1. Hence, the arc $v_1^-v_1$ can be inserted into C^* to give a hamiltonian cycle of D. This completes the proof that either D is hamiltonian or every arc on C_i can be inserted between two different pairs of vertices in C_{3-i}.

Assume without loss of generality that the length of C_2 is not greater than that of C_1. Then C_1 has two arcs x_iy_i ($i = 1, 2$) that can be inserted between the same pair u, v of vertices in C_2. Since C_1 is odd, one of the paths $Q = C_1[y_1^+, x_2^-]$ and $C_1[y_2^+, x_1^-]$ has odd length. Without loss of generality, suppose that Q is odd. Obviously, $C^* = C_2[v, u]C_1[x_2, y_1]v$ is a cycle of D. By the fact shown above each arc of the path Q can be inserted into C_2 between a pair of vertices different from u, v. Therefore, each arc of Q can be inserted into C^*. Hence, by Corollary 5.6.22 we conclude that D has a hamiltonian cycle H. It is not difficult to verify that H can be found in time $O(|V(C_1)||V(C_2)|)$. □

Let D be a semicomplete multipartite digraph and let $C \cup C'$ be a cycle subdigraph of D. We write that $C \mathrel{\simeq\!>} C'$ if C contains singular vertices with respect to C' and they all are out-singular, and C' has singular vertices with respect to C and they all are in-singular. A cycle factor $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_t$ is **good** if for every pair i, j, $1 \le i < j \le t$, neither $C_i \mathrel{\simeq\!>} C_j$ nor $C_j \mathrel{\simeq\!>} C_i$.

Since this definition and the proof of Lemma 5.7.12 are quite important, we illustrate them in Figure 5.7. Observe that if C, C' are a pair of disjoint cycles in a semicomplete multipartite digraph D, then (up to switching the role of the two cycles) at least one of the following four cases apply (see Figure 5.7):

(a) Every vertex on C has an arc to and from C'.
(b) There exist vertices $x \in V(C), y \in V(C')$ such that $x \Rightarrow V(C')$ and $y \Rightarrow V(C)$, or $V(C') \Rightarrow x$ and $V(C) \Rightarrow y$.
(c) C contains distinct vertices x, y such that $x \Rightarrow V(C')$ and $V(C') \Rightarrow y$.
(d) $C \mathrel{\simeq\!>} C'$.

The alternatives (a)-(c) are covered by the definition of a good cycle factor (for cycle factors containing only two cycles); the alternative (d) is not.

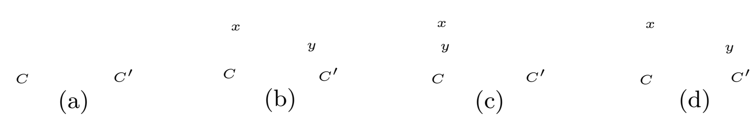


Figure 5.7 The four possible situations (up to switching the role of the two cycles or reversing all arcs) for arcs between two disjoint cycles in a semicomplete multipartite digraph. In (a) every vertex on C has arcs to and from C'. In (b)-(d) a fat arc indicates that all arcs go in the direction shown from or to the specified vertex (i.e. in (b) all arcs between x and C' leave x).

The following lemma gives the main result for a good cycle factor containing two cycles.

Lemma 5.7.12 *[68] If D is a semicomplete multipartite digraph containing a good factor $C_1 \cup C_2$, then D is hamiltonian. A Hamilton cycle in D can be constructed in time $O(|V(C_1)||V(C_2)|)$.*

Proof: The first case is that at least one of the cycles C_1 and C_2 has no singular vertices (Situation (a) in Figure 5.7). If both C_1, C_2 have no singular vertices then D is hamiltonian by Lemma 5.7.11 and we can find a Hamilton cycle in D in time $O(|V(C_1)||V(C_2)|)$. Assume now that only one of them has no singular vertices. Suppose without loss of generality that C_1 contains an out-singular vertex x and C_2 has no singular vertices. Since C_2 contains no singular vertices, C_1 has at least one vertex which is not out-singular. Suppose that $x \in V(C_1)$ was chosen such that x^+ is not out-singular. Hence there is a vertex y on C_2 dominating x^+. If $x \rightarrow y$, then y can be inserted into C_1 and hence, by Lemma 5.7.10 and Corollary 5.6.21, D is hamiltonian (consider $C_2[y^+, y]$ and C_1). Otherwise, x is not adjacent to y. In this case, $x \rightarrow y^+$ and D has the hamiltonian cycle $C_1[x^+, x]C_2[y^+, y]x$. The above arguments can be easily converted into an $O(|V(C_1)||V(C_2)|)$-algorithm.

Consider the second case: each of C_1, C_2 has singular vertices with respect to the other cycle. Assume without loss of generality that C_1 has an out-singular vertex x_1. If C_2 also contains an out-singular vertex x_2 (Situation (b) in Figure 5.7), then x_1 is not adjacent to x_2 and $x_i \rightarrow x^+_{3-i}$ for both $i = 1, 2$. Hence D is hamiltonian. If C_2 contains no out-singular vertices then it has in-singular vertices. Since $C_1 \cup C_2$ is a good factor, C_1 contains both out-singular and in-singular vertices (Situation (c) in Figure 5.7). Since both C_1 and C_2 have in-singular vertices, the digraph D' obtained from D by reversing the orientations of the arcs of D has two cycles C'_1 and C'_2 containing out-singular vertices. We conclude that D' (and hence D) is hamiltonian. Again, the above arguments can be converted into an $O(|V(C_1)||V(C_2)|)$-algorithm. $\square$

The main result on good cycle factors is the following theorem by Bang-Jensen, Gutin and Huang. This theorem can be proved by induction on t, the number of cycles in a good cycle factor. We leave the details to the reader (see Exercise 5.39).

Theorem 5.7.13 (Bang-Jensen, Gutin and Huang) *[68] If D is a strong semicomplete multipartite digraph containing a good cycle factor $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_t$ $(t \geq 1)$, then D is hamiltonian. Furthermore, given $\mathcal{F}$ one can find a hamiltonian cycle in D in time $O(n^2)$.* $\square$

5.7.3 Consequences of Lemma 5.7.12

In this subsection mostly based on [68], we will show that several important results on semicomplete multipartite digraphs are consequences of Lemma 5.7.12.

Proof of Theorem 5.7.1: It is sufficient to prove that if P is a path and C is a cycle of D such that $V(P) \cap V(C) = \emptyset$, then D has a path P' with $V(P') = V(P) \cup V(C)$. Let P and C be such a pair, and let u be the initial and v the terminal vertex of P. If u is non-singular or in-singular with respect to C, then obviously the path P' exists. Similarly if v is non-singular or out-singular with respect to C. Assume now that u is out-singular and v is in-singular with respect to C.

Add a new vertex w to D and the arcs zw, for all $z \neq u$ and the arc wu to obtain the semicomplete multipartite digraph D'. Then w forms a cycle C' with P in D' and $C \cup C'$ is a good cycle factor of D'. Therefore, by Lemma 5.7.12, D' has a hamiltonian cycle. Then D contains a hamiltonian path.

It is easy to see that the proof above supplies a recursive $O(n^2)$-algorithm for finding a hamiltonian path in D given a 1-path-cycle factor $\mathcal{F}$. Thus, the complexity result of this theorem is due to the fact that we can either construct a 1-path-cycle factor in a digraph or discover that it does not exist in time $O(n^{2.5})$: see Exercise 3.59. □

To obtain the rest of the proofs in this subsection, we need the following:

Lemma 5.7.14 *[68] Let D be a strong semicomplete multipartite digraph containing a cycle subdigraph $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_t$ such that for every pair i, j $(1 \leq i \leq j \leq t)$ $C_i \Rightarrow C_j$ or $C_j \Rightarrow C_i$ holds. Then D has a cycle C of length at least $|V(\mathcal{F})|$ and one can find C in time $O(n^2)$ for a given $\mathcal{F}$. If D is an extended semicomplete digraph, then we can choose C such that $V(\mathcal{F}) \subseteq V(C)$.*

Proof: Define a tournament $T(\mathcal{F})$ as follows: $\{C_1, \ldots, C_t\}$ forms the vertex set of $T(\mathcal{F})$ and $C_i \rightarrow C_j$ in $T(F)$ if and only if $C_i \Rightarrow C_j$ in D. Let H be the subdigraph of D induced by the vertices of $\mathcal{F}$ and let W be a partite set of D having a representative in C_1.

First consider the case that $T(\mathcal{F})$ is strong. Then it has a hamiltonian cycle. Without loss of generality assume that $C_1 C_2 \ldots C_t C_1$ is a hamiltonian cycle in $T(\mathcal{F})$. If each of C_i $(i = 1, 2, \ldots, t)$ has a vertex from W then for every $i = 1, 2, \ldots, t$ choose any vertex w_i of $V(C_i) \cap W$. Then $C_1[w_1, w_1^-]C_2[w_2, w_2^-] \ldots C_t[w_t, w_t^-]w_1$ is a hamiltonian cycle in H. If there exists a cycle C_i containing no vertices of W, then we may assume (shifting the cyclic order if needed) that C_t has no vertices from W. Obviously, H has a hamiltonian path starting at a vertex $w \in W \cap V(C_1)$ and finishing at some vertex v of C_t. Since $v \rightarrow w$, H is hamiltonian.

Now consider the case where $T(\mathcal{F})$ is not strong. Replacing in $\mathcal{F}$ every collection X of cycles which induce a strong component in $T(\mathcal{F})$ by a hamiltonian cycle in the subdigraph induced by X, we obtain a new cycle subdigraph $\mathcal{L}$ of D such that $T(\mathcal{L})$ has no cycles. The subdigraph $T(\mathcal{L})$ contains a unique hamiltonian path $Z_1 Z_2 \ldots Z_s$, where Z_i is a cycle of $\mathcal{L}$. Since D is strong there exists a path P in D with the first vertex in Z_s and the last vertex in Z_q $(1 \leq q < s)$ and the other vertices not in $\mathcal{L}$. Assume that q is as small as possible. Then we can replace the cycles $Z_q, \ldots, Z_s$ by a cycle consisting of all the vertices of $P \cup Z_q \cup \ldots \cup Z_s$ except maybe one and derive a new cycle subdigraph with less cycles. Continuing in this manner, we obtain finally a single cycle.

In the case of an extended semicomplete digraph D, if $D\langle V(\mathcal{F})\rangle$ is not strong, then $T(\mathcal{F})$ is not strong. Also, $C_i \Rightarrow C_j$ implies that $C_i \mapsto C_j$. This, combined with the above argument on semicomplete multipartite digraphs, allows one to construct a cycle C such that $V(\mathcal{F}) \subset V(C)$.

Using the above proof together with an $O(n^2)$-algorithm for constructing a hamiltonian cycle in a strong tournament (see Theorem 5.5.2 or Exercise 5.15) and obvious data structures one can obtain an $O(n^2)$-algorithm. $\square$

Lemma 5.7.15 *[68] Let $C \cup C'$ be a cycle factor in a strong semicomplete multipartite digraph D of order n. Then D has a cycle Z of length at least $n-1$ containing all vertices of C. The cycle Z can be found in time $O(|V(C)||V(C')|)$.*

Proof: Suppose that the (existence) claim is not true. By Lemma 5.7.12, this means that each of C and C' has singular vertices with respect to the other cycle, and all singular vertices on one cycle are out-singular and all singular vertices on the other cycle are in-singular. Assume without loss of generality that C has only out-singular vertices with respect to C'. Since D is strong C has a non-singular vertex x. Furthermore we can choose x such that its predecessor x^- on C is singular. Let y be some vertex of C' such that $y \rightarrow x$. If x^- is adjacent to y^+, the successor of y on C', then D has a hamiltonian cycle. Otherwise $x^- \rightarrow y^{++}$ and D has a cycle of length $n-1$ containing all vertices of C. The complexity result easily follows from the above arguments. $\square$

The next two results due to Gutin are easy corollaries of Lemma 5.7.15:

Corollary 5.7.16 *[353] Let $C \cup C'$ be a cycle factor in a strong semicomplete bipartite digraph D. Then D has a hamiltonian cycle Z. The cycle Z can be found in time $O(|V(C)||V(C')|)$.*

Proof: Since D is bipartite, it cannot have a cycle of length $n-1$. $\square$

Corollary 5.7.17 *[359] Let $C \cup C'$ be a cycle factor in a strong extended semicomplete digraph D. Then D has a hamiltonian cycle Z. The cycle Z can be found in time $O(|V(C)||V(C')|)$.*

Proof: If C and C' have a pair x, y of non-adjacent vertices ($x \in V(C),\ y \in V(C')$) then obviously $x \rightarrow y^+,\ y \rightarrow x^+$ and D has a Hamilton cycle that can be found in time $O(|V(C)|V(C')|)$. Assuming that any pair of vertices from C and C' is adjacent, we complete the proof as in Lemma 5.7.15. $\square$

Corollaries 5.7.16 and 5.7.17 imply immediately the following useful result.

Proposition 5.7.18 *If $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_k$ is a cycle factor in a digraph which is either semicomplete bipartite or extended semicomplete and there is no $\mathcal{F}' = C'_1 \cup C'_2 \cup \ldots \cup C'_r$ such that for every $i = 1, 2, \ldots, k$, $V(C_i) \subset V(C'_j)$ for some $j \in \{1, 2, , \ldots, r\}$, then without loss of generality $C_i \Rightarrow C_j$ for every $i < j$.*

Lemma 5.7.15 implies immediately the following result first proved by Ayel (see [449]).

Corollary 5.7.19 *If C is a longest cycle in a strong semicomplete multipartite digraph D, then $D - V(C)$ is acyclic.* □

Proof of Theorem 5.7.6: Let $\mathcal{F} = C_1 \cup \ldots \cup C_t$ be a cycle subdigraph of maximum order in a strong semicomplete bipartite digraph D. We construct a semicomplete digraph S, a generalization of the tournament T in Lemma 5.7.14, as follows. The vertices of S are the cycles in $\mathcal{F}$, $C_i \rightarrow C_j$ in S if and only if there is an arc from C_i to C_j in D. Cycles of length two in S indicate what cycles in $\mathcal{F}$ can be merged together by Corollary 5.7.16. Therefore, we can merge cycles in $\mathcal{F}$ till S becomes oriented, i.e. without 2-cycles. Now we can apply Lemma 5.7.14.

Complexity details are left to the reader. □

Proof of Theorem 5.7.7: The proof is similar to that of Theorem 5.7.6, applying Corollary 5.7.17 instead of Corollary 5.7.16. Details are left to the reader as Exercise 5.35. □

5.7.4 Yeo's Irreducible Cycle Subdigraph Theorem and its Applications

While Lemma 5.7.12 is strong enough to imply short proofs of results on longest cycles in some special families of semicomplete multipartite digraphs such as semicomplete bipartite graphs and extended semicomplete digraphs, this lemma does not appear strong enough to be used in proofs of longest cycle structure results for other families of semicomplete multipartite digraphs. In this subsection based on Yeo's paper [744], we formulate the very deep theorem of Yeo on irreducible cycle subdigraphs in semicomplete multipartite digraphs, the main theorem in [744], that is more powerful than Lemma 5.7.12. We give a proof of the main lemma (Lemma 5.7.20) in the original proof of Yeo's theorem, but do not provide the rest of the lemmas since these would require significant space. We provide short proofs of some important consequences of this theorem.

Recall that for two subdigraphs X, Y of D, a path P is an (X, Y)-path if P starts at a vertex $x \in V(X)$, terminates at a vertex $y \in V(Y)$ and $V(P) \cap (V(X) \cup V(Y)) = \{x, y\}$.

Lemma 5.7.20 *[744] Let D be a semicomplete multipartite digraph, and let C_1 and C_2 be a pair of disjoint cycles in D, such that $C_1 \simeq\!> C_2$ and $C_1 \not\Rightarrow C_2$. Assume that there is no cycle in D, with vertex set $V(C_1) \cup V(C_2)$. Then there exists a unique partite set V of D such that for every $(V(C_2), V(C_1))$-path P starting at vertex u and terminating at vertex v either $\{u^+_{C_2}, v^-_{C_1}\} \subseteq V$ or there exists a cycle C^* in D, with $V(C^*) = V(C_1) \cup V(C_2) \cup V(P)$.*

Proof: Since $C_1 \simeq\!> C_2$ and $C_1 \not\Rightarrow C_2$, there is a vertex $x \in V(C_1)$, with $x \Rightarrow C_2$ and $x^+ \not\Rightarrow C_2$. Let V be the partite set containing the vertex x. Let $y \in V(C_2)$ be chosen such that $y^- \rightarrow x^+$. Then $y \in V$, since otherwise $C = C_2[y, y^-]C_1[x^+, x]y$ is a cycle with $V(C) = V(C_1) \cup V(C_2)$. We will now show the following assertion:

$$V(C_1) \Rightarrow y. \tag{5.9}$$

Label the vertices in C_2 such that $C_2 = y_1 y_2 \ldots y_m y_1$, where $y_1 = y$, and assume that (5.9) is not true, i.e. $V(C_1) \not\Rightarrow y_1$. Define the statements α_K and β_K as follows.

α_K: The vertex $y_k \in V$ and $V(C_1) \not\Rightarrow y_k$, for every $k = 1, 3, 5, \ldots, K$.
β_K: The arc $y_k y_{k+1}$ can be inserted into $C_1[x^+, x]$, for every $k = 1, 3, 5, \ldots, K$.

We will now show that α_K and β_K are true for every odd K, with $1 \leq K < m$. Clearly α_1 holds, so if we prove the following two implications, we are done by induction.

α_K and β_{K-2} imply β_K (when $K = 1$, α_K implies β_K): If we can insert y_K into C_1, then it can be inserted into $C_1[x^+, x]$, since y_K cannot be inserted between x and x^+ (by α_K, $y_K \in V$). Also, by β_{K-2} and Corollary 5.6.22 we can insert the path $C_2[y_1, y_K]$ into the cycle $C_1[x^+, x]C_2[y_K^+, y_m]x^+$. So we may assume that y_K cannot be inserted into C_1. Since $C_1 \not\Rightarrow y_K$, there must be a $z_K \in V(C_1)$ such that $z_K \in V$ and $z_K^- \rightarrow y_K \rightarrow z_K^+$. Now $y_K^+ \rightarrow z_K$, since there otherwise would be a cycle, $C = C_2[y_K^+, y_K]C_1[z_K^+, z_K]y_K^+$, in D with $V(C) = V(C_1) \cup V(C_2)$. Thus $y_K y_K^+$ can be inserted between z_K^- and z_K, which implies that $y_K y_K^+$ can be inserted into $C_1[x^+, x]$, since $z_K^- \neq x$ ($z_K^- \notin V$).

α_{K-2} and β_{K-2} imply α_K: $y_K \in V$, since otherwise by β_{K-2} and Corollary 5.6.22 we can insert the path $P = y_1 y_2 \ldots y_{K-1}$ into the cycle

$$C_1[x^+, x]C_2[y_K, y_m]x^+$$

and obtain a cycle in D with vertex set $V(C_1) \cup V(C_2)$.
If $V(C_1) \Rightarrow y_K$, then $z_{K-2}^- \rightarrow y_K$, where z_{K-2} was defined when we proved β_{K-2}. When we defined z_{K-2}, we found that $y_{K-2}^+ \rightarrow z_{K-2}$. The cycle

$$C = C_1[z_{K-2}, z_{K-2}^-]C_2[y_K, y_{K-2}^+]z_{K-2}$$

has $V(C) = V(C_1) \cup V(C_2)$, a contradiction. This completes the proof that α_K holds.

Since y_m can be inserted into C_1 (namely between x and x^+), Corollary 5.6.22 implies that we can insert the path $C_2[y_1, y_m]$ into C_1 to obtain a new cycle in D with vertex set $V(C_1) \cup V(C_2)$. This is a contradiction, which implies that (5.9).

Let $u^+ = u^+_{C_2}$ and $v^- = v^-_{C_1}$. To complete the proof of this lemma it suffices to consider the following two cases.

Case 1: $\{u^+, v^-\} \cap V = \emptyset$. The cycle

$$C^* = C_1[x^+, v^-]C_2[y, u]P[u^+_P, v^-_P]C_1[v, x]C_2[u^+, y^-]x^+$$

has $V(C^*) = V(C_1) \cup V(C_2) \cup V(P)$.

Case 2: The vertices u^+ and v^- are in different partite sets. We claim that D contains a cycle C^*, with $V(C^*) = V(P) \cup V(C_1) \cup V(C_2)$. Assume that C^* does not exist. According to Case 1, we have that either $u^+ \in V$ or $v^- \in V$, but not both. Now we may assume that $u^+ \rightarrow v^-$, since otherwise $C^* = C_1[v, v^-]C_2[u^+, u^-_{C_2}]P$ would have $V(C^*) = V(P) \cup V(C_1) \cup V(C_2)$. Now according to Case 1, used for the path $P' = u^+v^-$, we have that either $u^{++} \in V$ or $v^{--} \in V$, but not both, since either $u^+ \in V$ or $v^- \in V$. Continuing this process and using the fact that D has no cycle with vertex set $V(P) \cup V(C_1) \cup V(C_2)$ we obtain that $u^+ \rightarrow v^-, u^{++} \rightarrow v^{--}, \ldots$ which clearly is impossible since C_1 has an out-singular vertex with respect to C_2. This is a contradiction, and thus C^* exists. □

Lemma 5.7.20 and several other results in [744] imply the following powerful theorem. Notice that, in fact, Yeo [744] proved three sets of properties of irreducible subdigraph. We include only the two most important ones.

Theorem 5.7.21 (Yeo's irreducible cycle subdigraph theorem) *[744] Let D be a semicomplete multipartite digraph with partite sets $V_1, V_2, \ldots, V_c$. Let $X \subseteq V(D)$ and let $\mathcal{F}$ be a cycle subdigraph of D consisting of t cycles that covers X, such that t is minimum. Then the following holds.*

(a) We can label the cycles $C_1, C_2, \ldots, C_t$ of $\mathcal{F}$, such that $C_i \simeq> C_j$, whenever $1 \le i < j \le t$.
(b) Assume that $C_1, C_2, \ldots, C_t$ are ordered as stated in (a), then there are cycles $C_{n_0}, C_{n_1}, \ldots, C_{n_m}$ ($n_0 = 1$, $n_m = t$), and integers $q_1, q_2, \ldots, q_m \in \{1, 2, \ldots, c\}$, such that the following is true. For every (C_j, C_i)-path P starting at u and terminating at v with $V(P) \cap V(\mathcal{F}) = \{u, v\}$ and $1 \le i < j \le t$, there exists an integer $k \in \{1, 2, \ldots, m\}$, such that $n_{k-1} \le i < j \le n_k$ and $\{u^+_{C_i}, v^-_{C_j}\} \subseteq V_{q_k} \cap X$. □

By a careful analysis of the complete proof of Theorem 5.7.21 in [745] one can obtain the following:

Theorem 5.7.22 *[745] Let D be a semicomplete multipartite digraph, and let $X \subseteq V(D)$ be arbitrary. Let $\mathcal{F}$ be a cycle subdigraph of D that covers X. Then in $O(|V(D)|^3)$ time we can find a new cycle subdigraph, $\mathcal{F}'$, of D, that covers X, such that $\mathcal{F}'$ has the properties (a) and (b) given in Theorem 5.7.21. Furthermore we can find $\mathcal{F}'$, such that for every cycle C in $\mathcal{F}$, the vertices $X \cap V(C)$ are included in some cycle of $\mathcal{F}'$.* □

Theorems 5.7.21 and 5.7.22 are very important starting points of [72], where polynomial solvability of the Hamilton cycle problem for semicomplete multipartite digraphs is established. We will prove some important consequences of Theorem 5.7.21 and state several more of them.

Theorem 5.7.23 *[744] Every regular semicomplete multipartite digraph is hamiltonian.*

Proof: Let D be a regular semicomplete multipartite digraph. By Exercise 3.70, D contains a cycle factor $\mathcal{F} = C_1 \cup C_2 \cup \ldots \cup C_t$. We may assume that $\mathcal{F}$ is chosen, such that t is minimum. If $t = 1$ then D is hamiltonian, so assume that $t > 1$.

Let $X = V(D)$. Let $C_{n_0}, C_{n_1}, \ldots, C_{n_m}$ and $q_1, q_2, \ldots, q_m$ be defined as in Theorem 5.7.21. Let $yx \in A(D)$ be an arc from $y \in V(C_i)$, with $i \in \{2, 3, \ldots, t\}$ to $x \in V(C_1)$. Part (b) of Theorem 5.7.21 implies that $x^-, y^+ \in V_{q_1}$. Now we define the two distinct arcs $a_1(yx) = xy^+$ and $a_2(yx) = x^-y$. By Theorem 5.7.21, $a_1(yx)$ and $a_2(yx)$ are arcs in D. Indeed, x and y^+ (x^- and y) are adjacent. If $y^+ \rightarrow x$ then $y^{++} \in V_{q_1}$, which is impossible.

If $y'x'$ and yx are distinct arcs from $V(D) - V(C_1)$ to $V(C_1)$, then we see that $a_1(yx)$, $a_2(yx)$, $a_1(y'x')$ and $a_2(y'x')$ are four distinct arcs from $V(C_1)$ to $V(D) - V(C_1)$. We have now shown that the number of arcs leaving $V(C_1)$ is at least twice as large as the number of arcs entering $V(C_1)$. However this contradicts the fact that D is an eulerian digraph (see Corollary 1.6.4). $\square$

Theorem 5.7.24 (Yeo) *[744] Let D be a $(\lfloor k/2 \rfloor + 1)$-strong semicomplete multipartite digraph, and let X be an arbitrary set of vertices in D such that X includes at most k vertices from each partite set of D. If there is a cycle subdigraph $\mathcal{F} = C_1 \cup \ldots \cup C_t$, which covers X, then there is a cycle C in D, such that $X \subseteq V(C)$.*

Proof: We may clearly assume that $\mathcal{F}$ has the properties described in Theorem 5.7.21, and $t \geq 2$, since otherwise we are done. Let $C_{n_0}, C_{n_1}, \ldots, C_{n_m}$ and $q_1, q_2, \ldots, q_m$ be defined as in Theorem 5.7.21. Since X contains at most k vertices from each partite set, we have that $\min\{|V_{q_1} \cap V(C_1) \cap X|, |V_{q_1} \cap V(C_{n_1}) \cap X|\} = r \leq \lfloor k/2 \rfloor$. Assume without loss of generality that $|V_{q_1} \cap V(C_{n_1}) \cap X| = r$. Since D is $(\lfloor k/2 \rfloor + 1)$-strong we get that there exists a $(V(C_{n_1}) - (V_{q_1} \cap V(C_{n_1}) \cap X)^-, V(C_1) \cup \ldots \cup V(C_{n_1 - 1}))$-path in $D - (V_{q_1} \cap V(C_{n_1}) \cap X)^-$, $P = p_1 \ldots p_l$. Assume that $p_l \in V(C_i)$ $(1 \leq i < n_1)$. By Theorem 5.7.21, the (C_{n_1}, C_i)-path P contradicts the minimality of $\mathcal{F}$, since $n_0 \leq i < n_1$ and $p_1^+ \notin X \cap V_{q_1}$. $\square$

A family of semicomplete multipartite digraphs described in [744] shows that one cannot weaken the value $\lfloor k/2 \rfloor + 1$ of strong connectivity in this theorem. Using the fact that every k-strong digraph of independence number at most k has a cycle factor (see Proposition 3.11.12) and applying Theorem 5.7.24, we obtain the following two corollaries:

Corollary 5.7.25 *[744] If a k-strong semicomplete multipartite digraph D has at most k vertices in each partite set, then D contains a Hamilton cycle.* □

Corollary 5.7.26 *[744] A k-strong semicomplete multipartite digraph has a cycle through any set of k vertices.* □

Theorem 5.7.23 was generalized by Yeo [748] as follows (its proof also uses Theorem 5.7.21). Let $i_l(D) = \max\{|d^+(x) - d^-(x)| : x \in V(D)\}$ and $i_g(D) = \Delta^0(D) - \delta^0(D)$ for a digraph D (the two parameters are called the **local irregularity** and the **global irregularity**, respectively, of D [748]). Clearly, $i_l(D) \le i_g(D)$ for every digraph D.

Theorem 5.7.27 *[748] Let D be a semicomplete c-partite digraph of order n with partite sets of cardinalities $n_1, n_2, \ldots, n_c$ such that $n_1 \le n_2 \le \ldots \le n_c$. If $i_g(D) \le (n - n_{c-1} - 2n_c)/2 + 1$ or $i_l(D) \le \min\{n - 3n_c + 1, (n - n_{c-1} - 2n_c)/2 + 1\}$, then D is hamiltonian.* □

The result of this theorem is best possible in a sense: Yeo [748] constructed an infinite family $\mathcal{D}$ of non-hamiltonian semicomplete multipartite digraphs such that every $D \in \mathcal{D}$ has $i_l(D) = i_g(D) = (n - n_{c-1} - 2n_c + 1)/2 + 1 \le n - 3n_c + 2$.

Another generalization of Theorem 5.7.23, whose proof is based on Theorem 5.7.21, was obtained by Guo, Tewes, Volkmann and Yeo [348]. For a digraph D and a positive integer k, define

$$f(D,k) = \sum_{x \in V(D), d^+(x) > k} (d^+(x) - k) + \sum_{x \in V(D), d^-(x) < k} (k - d^-(x)).$$

Theorem 7.5.3 in Ore's book [595] on the existence of a perfect matching in a bipartite graph can easily be transformed into a sufficient condition for a digraph to contain a cycle factor. This condition is as follows. If, for a digraph D and positive integer k, we have $f(D,k) \le k-1$, then D has a cycle factor. For a positive integer $k \ge 2$, let G'_k be a semicomplete 3-partite digraph with the partite sets $V_1 = \{x\}$, $V_2 = \{y_1, y_2, \ldots, y_{k-1}\}$, and $V_3 = \{z_1, z_2, \ldots, z_k\}$ and arc set

$$\{yx, xz, zy, yv : y \in V_2, z \in V_3, v \in V_3 - z_1\} \cup \{z_1 x\}.$$

The digraph G''_k is the converse of G'_k. We observe that $f(G'_k, k) = k - 1$ (Exercise 5.43), but G'_k is not hamiltonian, as a hamiltonian cycle would contain the arc xz_1 and every second vertex on the cycle would belong to the partite set V_3. Since x has no in-neighbour in $V_3 - z_1$, this is not possible. Clearly, G''_k is not hamiltonian either.

Theorem 5.7.28 *[348] Let D be a semicomplete multipartite digraph such that $f(D,k) \le k-1$ for some positive integer k. If D is not isomorphic to G'_k or G''_k, then D is Hamiltonian.* □

The authors of [348] introduced the following family of semicomplete multipartite digraphs. Let D be a semicomplete multipartite digraph with partite sets $V_1, V_2, \dots, V_k$. If $\min\{|(x_i, V_j)|, |(V_j, x_i)|\} \geq \frac{1}{2}|V_j|$ for every vertex $x_i \in V_i$ and for every $1 \leq i, j \leq k, \quad j \neq i$, then D is called a **semi-partitioncomplete digraph**. Several sufficient conditions to guarantee hamiltonicity of semi-partitioncomplete digraphs were derived in [348]. In particular, the following result was proved.

Theorem 5.7.29 *If a strong semi-partitioncomplete digraph D of order n has less than $n/2$ vertices in every partite set, then D is hamiltonian.* □

5.8 Longest Paths and Cycles in Extended Locally Semicomplete Digraphs

¿From Section 5.5, we know that characterizations of hamiltonian and traceable locally semicomplete digraphs are practically the same as those of semicomplete digraphs: every strong locally semicomplete digraph is hamiltonian and every connected locally semicomplete digraph is traceable. In the previous section, we derived characterizations of hamiltonian and traceable extended semicomplete digraphs. The reader may suspect that similar characterizations hold for extended locally semicomplete digraphs. This is indeed true. Moreover, the hamiltonicity characterization can be generalized even to extended locally in-semicomplete digraphs. However, the traceability one does not hold for extended locally in-semicomplete digraphs. In this section we briefly consider these characterizations and their generalizations to the longest path and cycle problems. We start from the following characterization by Bang-Jensen and Gutin [62].

Theorem 5.8.1 *An extended locally semicomplete digraph is hamiltonian if and only if it is strongly connected and has a cycle factor. Given a cycle factor of a strong extended locally semicomplete digraph D, a hamiltonian cycle of D can be found in time $O(n^2)$, where n is the number of vertices in D.* □

This theorem can be generalized to extended locally in-semicomplete digraphs [59]. Theorem 5.8.2, whose proof is left as Exercise 5.44, shows that extended locally semicomplete digraphs are still 'nicer' with respect to the longest cycle than semicomplete bipartite digraphs (see the remark after Theorem 5.7.7).

Theorem 5.8.2 *[62] Let D be a strongly connected extended locally semicomplete digraph. Given a cycle subdigraph $\mathcal{F} = C_1 \cup \dots \cup C_t$ of D of maximum order, one can find a (longest) cycle C of D such that $V(C) = V(C_1) \cup \dots \cup V(C_t)$ in time $O(n^2)$.* □

Theorem 5.8.3 *[62] A connected extended locally semicomplete digraph D has a hamiltonian path if and only if it contains a 1-path-cycle factor. Given a 1-path-cycle factor of D, one can construct a hamiltonian path of D in time $O(n^2)$.*

Proof: Exercise 5.45. □

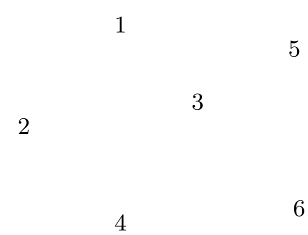


Figure 5.8 The digraph L.

Unlike Theorem 5.8.1, Theorem 5.8.3 cannot be generalized to extended locally in-semicomplete digraphs as one can see from the following example [59]. The extended locally in-semicomplete digraph L in Figure 5.8 contains a 1-path-cycle factor consisting of path 1234 and cycle 565 (and even an in-branching rooted in the vertex 6), but has no hamiltonian path. It is natural to pose the following problem:

Problem 5.8.4 *[65]*

(a) Find a characterization of traceable extended locally in-semicomplete digraphs.
(b) Establish the complexity of the problem of deciding whether an extended locally in-semicomplete digraph has a hamiltonian path.

Theorem 5.8.3 can easily be generalized to longest paths.

Theorem 5.8.5 *[62] The order of a longest path in an extended locally semicomplete digraph D equals to the maximum order of a 1-path-cycle subdigraph of D. Moreover, given a 1-path-cycle subdigraph $\mathcal{F}$ of an extended locally semicomplete digraph D, a path P such that $V(P) = V(\mathcal{F})$ can be found in time $O(n^2)$.* □

5.9 Hamilton Paths and Cycles in Quasi-Transitive Digraphs

The methods developed in [79] by Bang-Jensen and Huang and [365] by Gutin to characterize hamiltonian and traceable quasi-transitive digraphs as well as to construct polynomial algorithms for verifying the existence of Hamilton paths and cycles in quasi-transitive digraphs can be easily generalized to much wider classes of digraphs [65]. Thus, in this section, along with quasi-transitive digraphs, we consider totally Φ-decomposable digraphs for various sets Φ of digraphs.

By Theorem 4.8.5, every strong quasi-transitive digraph D has a decomposition $D = S[Q_1, Q_2, \ldots, Q_s]$, where S is a strong semicomplete digraph, $s = |V(S)|$, and each Q_i, $i = 1, 2, \ldots, s$, is either just a single vertex or a non-strong quasi-transitive digraph. Also, a non-strong quasi-transitive digraph D with at least two vertices has a decomposition $D = T[H_1, H_2, \ldots, H_t]$, where T is a transitive oriented graph, $t = |V(T)|$, and every H_i is a strong semicomplete digraph. These decompositions are called canonical decompositions. The following characterization of hamiltonian quasi-transitive digraphs is due to Bang-Jensen and Huang [79].

Theorem 5.9.1 *[79] A strong quasi-transitive digraph D with canonical decomposition $D = S[Q_1, Q_2, \ldots, Q_s]$ is hamiltonian if and only if it has a cycle factor $\mathcal{F}$ such that no cycle of $\mathcal{F}$ is a cycle of some Q_i.*

Proof: Clearly, a Hamilton cycle in D crosses every Q_i. Thus, it suffices to show that, if D has a cycle factor $\mathcal{F}$ such that no cycle of $\mathcal{F}$ is a cycle of some Q_i, then D is hamiltonian. Observe that $V(Q_i) \cap \mathcal{F}$ is a path factor $\mathcal{F}_i$ of Q_i for every $i = 1, 2, \ldots, s$. For every $i = 1, 2, \ldots, s$, delete the arcs between end-vertices of all paths in $\mathcal{F}_i$ except for the paths themselves, and then perform the operation of path-contraction for all paths in $\mathcal{F}_i$. As a result, one obtains an extended semicomplete digraph S' (since S is semicomplete). Clearly, S' is strong and has a cycle factor. Hence, by Theorem 5.7.5, S' has a Hamilton cycle C. After replacing every vertex of S' with the corresponding path from $\mathcal{F}$, we obtain a Hamilton cycle in D. $\square$

Similarly to Theorem 5.9.1, one can prove the following characterization of traceable quasi-transitive digraphs (see Exercise 5.47).

Theorem 5.9.2 *[79] A quasi-transitive digraph D with at least two vertices and with canonical decomposition $D = R[G_1, G_2, \ldots, G_r]$ is traceable if and only if it has a 1-path-cycle factor $\mathcal{F}$ such that no cycle or path of $\mathcal{F}$ is completely in some $D\langle V(G_i)\rangle$.* $\square$

It appears that Theorems 5.9.1 and 5.9.2 do not imply polynomial algorithms to verify hamiltonicity and traceability, respectively (see Exercise 5.46). The following characterization of hamiltonian quasi-transitive digraphs is given implicitly in the paper [365] by Gutin:

Theorem 5.9.3 (Gutin) *[365] Let D be a strong quasi-transitive digraph with canonical decomposition $D = S[Q_1, Q_2, \ldots, Q_s]$. Let $n_1, \ldots, n_s$ be the orders of the digraphs $Q_1, Q_2, \ldots, Q_s$, respectively. Then D is hamiltonian if and only if the extended semicomplete digraph $S' = S[\overline{K}_{n_1}, \overline{K}_{n_2}, \ldots, \overline{K}_{n_s}]$*

has a cycle subdigraph which covers at least $\mathrm{pc}(Q_j)$ *vertices of* $\overline{K}_{n_j}$ *for every* $j = 1, 2, \ldots, s$.

Proof: Suppose that D has a Hamilton cycle H. For every $j = 1, 2, \ldots, s$, $V(Q_j) \cap H$ is a k_j-path factor $\mathcal{F}_j$ of Q_j. By the definition of the path covering number, we have $k_j \geq \mathrm{pc}(Q_j)$. For every $j = 1, 2, \ldots, s$, the deletion of the arcs between end-vertices of all paths in $\mathcal{F}_j$ except for the paths themselves, and then path-contraction of all paths in $\mathcal{F}_j$ transforms H into a cycle of S' having at least $\mathrm{pc}(Q_j)$ vertices of $\overline{K}_{n_j}$ for every $j = 1, 2, \ldots s$.

Suppose now that S' has a cycle subdigraph $\mathcal{L}$ containing $p_j \geq \mathrm{pc}(Q_j)$ vertices of $\overline{K}_{n_j}$ for every $j = 1, 2, \ldots s$. Since S' is a strong extended semicomplete digraph, by Theorem 5.7.7, S' has a cycle C such that $V(C) = V(\mathcal{L})$. Clearly, every Q_j has a p_j-path factor $\mathcal{F}_j$. Replacing, for every $j = 1, 2, \ldots s$, the p_j vertices of $\overline{K}_{n_j}$ in C with the paths of $\mathcal{F}_j$, we obtain a hamiltonian cycle in D. □

Theorem 5.9.3 can be used to show that the Hamilton cycle problem for quasi-transitive digraphs is polynomial time solvable.

Theorem 5.9.4 (Gutin) *[365] There is an* $O(n^4)$ *algorithm which, given a quasi-transitive digraph* D*, either returns a hamiltonian cycle in* D *or verifies that no such cycle exists.* □

The approach used in the proofs of Theorems 5.9.3 and 5.9.4 in [365] can be generalized to a much wider class of digraphs as was observed by Bang-Jensen and Gutin [65]. We follow the main ideas of [65].

Theorem 5.9.5 *Let* Φ *be an extension-closed set of digraphs, i.e.* $\Phi^{ext} = \Phi$*, including the trivial digraph* $\overline{K}_1$ *on one vertex. Suppose that for every digraph* $H \in \Phi$ *we have* $\mathrm{pcc}(H) = \mathrm{pc}(H)$*. Let* D *be a totally* Φ*-decomposable digraph. Then, given a total* Φ*-decomposition of* D*, the path covering number of* D *can be calculated and a minimum path factor found in time* $O(n^4)$*.*

Proof: We prove this theorem by induction on n. For $n = 1$ the claim is trivial.

Let D be a totally Φ-decomposable digraph and let $D = R[H_1, \ldots, H_r]$ be a Φ-decomposition of D such that $R \in \Phi$, $r = |V(R)|$ and every H_i (of order n_i) is totally Φ-decomposable. A $\mathrm{pc}(D)$-path factor of D restricted to every H_i corresponds to a disjoint collection of some p_i paths covering $V(H_i)$. Hence, we have $\mathrm{pc}(H_i) \leq p_i \leq n_i$. Therefore, arguing similarly to that in the proof of Theorem 5.9.3, we obtain

$$\mathrm{pc}(D) = \min\{\mathrm{pc}(R[\overline{K}_{p_1}, \ldots, \overline{K}_{p_r}]) : \ \mathrm{pc}(H_i) \leq p_i \leq n_i, \ i = 1, \ldots, r\}.$$

Since Φ is extension-closed, and since, for every digraph $Q \in \Phi$, $\mathrm{pc}(Q) = \mathrm{pcc}(Q)$, we obtain

$$\mathrm{pc}(D) = \min\{\mathrm{pcc}(R[\overline{K}_{p_1}, \dots, \overline{K}_{p_r}]) : \ \mathrm{pc}(H_i) \le p_i \le n_i, \ i = 1, \dots, r\}. \tag{5.10}$$

By the result of Exercise 3.60, given the lower and upper bounds $\mathrm{pc}(H_i)$ and n_i ($i = 1, \dots, r$), we can find the minimum in (5.10) and thus $\mathrm{pc}(D)$ in time $O(n^3)$. Let $T(n)$ be the time needed to find the path covering number of a totally Φ-decomposable digraph of order n. Then, by (5.10),

$$T(n) = O(n^3) + \sum_{i=1}^{r} T(n_i).$$

Furthermore, $T(1) = O(1)$. Hence $T(n) = O(n^4)$. $\square$

Recall (see Section 4.12) that Φ_0 (Φ_2) is the family of all semicomplete multipartite, extended locally semicomplete and acyclic digraphs (semicomplete bipartite, extended locally semicomplete and acyclic digraphs). Clearly, both families of digraphs are extension-closed. As we know, $\mathrm{pc}(D) = \mathrm{pcc}(D)$ for every semicomplete multipartite digraph D (see Theorem 5.7.2), for every extended locally semicomplete digraph D (by Theorem 5.8.3) and every acyclic digraph D (which is trivial). Notice that one can check whether a digraph D is totally Φ_0-decomposable (totally Φ_2-decomposable) and, if this is the case, find a total Φ_0-decomposition (Φ_2-decomposition) in time $O(n^4)$ (see Section 4.12). Therefore, Theorem 5.9.5 implies the following theorem by Bang-Jensen and Gutin:

Theorem 5.9.6 *[66] The path covering number of a totally Φ_0-decomposable digraph can be calculated in time $O(n^4)$.* $\square$

Corollary 5.9.7 *[66] One can verify whether a totally Φ_2-decomposable digraph is hamiltonian in time $O(n^4)$.*

Proof: Let $D = R[H_1, \dots, H_r]$, $r = |R|$, be a decomposition of a strong digraph D ($r \ge 2$). Then, D is hamiltonian if and only if the following family $\mathcal{S}$ of digraphs contains a hamiltonian digraph:

$$\mathcal{S} = \{R[\overline{K}_{p_1}, \dots, \overline{K}_{p_r}] : \ \mathrm{pc}(H_i) \le p_i \le |V(H_i)|, \ i = 1, \dots, r\}.$$

Now suppose that D is a totally Φ_2-decomposable digraph. Then, every digraph of the form $R[\overline{K}_{p_1}, \dots, \overline{K}_{p_r}]$ is in Φ_2. We know (see Theorems 5.7.4 and 5.8.1) that every digraph in Φ_2 is hamiltonian if and only if it is strong and contains a cycle factor. Thus, all we need is to verify whether there is a digraph in $\mathcal{S}$ containing a cycle factor. It is easily seen that there is a digraph in $\mathcal{S}$ containing a cycle factor if and only if there is a circulation in the network formed from R by adding lower bounds $\mathrm{pc}(H_i)$ and upper bounds $|V(H_i)|$ to the vertex v_i of R for every $i = 1, \dots, r$. Since the lower bounds can be found in time $O(n^4)$ (see Theorem 5.9.5) and the existence of a circulation checked in time $O(n^3)$ (see Exercise 3.31), we obtain the required complexity $O(n^4)$. □

Since every quasi-transitive digraph is totally Φ_2-decomposable this theorem immediately implies Theorem 5.9.4. Note that the minimum path factors in Theorem 5.9.5 can be found in time $O(n^4)$. Also, a hamiltonian cycle in a hamiltonian totally Φ_2-decomposable digraph can be constructed in time $O(n^4)$.

5.10 Vertex-Heaviest Paths and Cycles in Quasi-Transitive Digraphs

The approach described in the previous section seems to be of not sufficient power to allow us to construct polynomial time algorithms for longest paths and cycles in quasi-transitive digraphs and their generalizations. A more powerful method that leads to such algorithms was first suggested by Bang-Jensen and Gutin [63]. In this section, we describe the method in [63].

¿From now on, assume that every digraph D we consider has non-negative weights $w(.)$ on the vertices. Recall that the (vertex-)weight $w(H)$ of a subdigraph of D is the sum of the weights of its vertices. For a positive integer k, the symbol $w_k(D)$ denotes the weight of a heaviest k-path subdigraph of D, i.e. one with the maximum weight among all k-path subdigraphs. For convenience we define $w_0(D) = 0$. We consider the following problem which we call **the HPS problem**. Given a digraph D on n vertices, find a heaviest k-path subdigraph of D for every $k = 1, 2, \dots, n$.

Theorem 5.10.1 *[63] Let Φ be a set of digraphs including the digraph on one vertex. Suppose that $\Phi = \Phi^{ext}$ and, for every $D \in \Phi$ on n vertices,*

$$w_{k+1}(D) - w_k(D) \leq w_k(D) - w_{k-1}(D), \tag{5.11}$$

where $k = 1, 2, \dots, n-1$. If there is a constant $s \geq 2$ so that, for every $L \in \Phi$, the HPS problem can be solved in time $O(|V(L)|^s)$, then, for every totally Φ-decomposable digraph D, the HPS problem can be solved in time $O(|V(D)|^{s+1})$, provided we are given a total Φ-decomposition of D.

Proof: Let $D = R[H_1, \dots, H_r]$ be a decomposition of D, where $R \in \Phi$ and H_i is totally Φ-decomposable and has n_i vertices ($i = 1, \dots, r$). Set $D_0 = R[E_1, \dots, E_r]$, where E_i is the digraph with n_i vertices and no arcs. Assign new weights to the vertices of D_0 as follows. The ith vertex of E_j is assigned the weight

$$\widetilde{w}_{ij} = w_i(H_j) - w_{i-1}(H_j), \quad j = 1, \dots, r;\ i = 1, \dots, n_j.$$

We show that, given solutions of the HPS problem for $H_1, \ldots, H_r$ and D_0, one can easily construct a solution of the HPS problem for D. This will lead to a recursive algorithm as desired.

Let $\mathcal{F}_k$ be a heaviest k-path subdigraph of D_0 and let m_j be the number of vertices in $\mathcal{F}_k$, which belong to E_j ($j = 1, \ldots, r$). By (5.11), $\widetilde{w}_{ij} \geq \widetilde{w}_{qj}$ whenever $q > i$. Therefore, using that all vertices in E_j are similar, we can always change the vertices of $\mathcal{F}_k$ so that $\mathcal{F}_k$ contains precisely the first m_j vertices of E_j for each $j = 1, \ldots, r$. Assume now that this is the case. Now, for each $j = 1, \ldots, r$, replace the vertices of E_j in $\mathcal{F}_k$ by a heaviest m_j-path subdigraph of H_j. This replacement provides a k-path subdigraph $\mathcal{T}_k$ of D. It is easy to check that

$$\widetilde{w}(\mathcal{F}_k) = \sum_{j=1}^{r} \sum_{i=1}^{m_j} \widetilde{w}_{ij} = \sum_{j=1}^{r} w_{m_j}(H_j) = w(\mathcal{T}_k) \leq w_k(D).$$

So, the weight of a heaviest k-path subdigraph of D_0 is at most $w_k(D)$. Analogously, starting with a heaviest k-path subdigraph of D, one can prove that the weight of a heaviest k-path subdigraph of D_0 is at least $w_k(D)$. Therefore, $\mathcal{T}_k$ is a heaviest k-path subdigraph of D.

The arguments above lead to the following recursive algorithm called $\mathcal{A}_{\mathcal{HPS}}$.

1. Use the total Φ-decomposition of D to find the decomposition $D = R[H_1, \ldots, H_r]$.
2. Solve the HPS problem for $H_1, \ldots, H_r$ using $\mathcal{A}_{\mathcal{HPS}}$.
3. Form D_0 (with the weights $\widetilde{w}_{ij}$) and solve the HPS problem for D_0 using an $O(|V(D)|^s)$-time algorithm. Change the solutions $\mathcal{F}_k$ (if it is necessary) so that each of $\mathcal{F}_k$ contains the first vertices of E_j without 'blanks', for each $j = 1, \ldots, r$.
4. Using the solutions obtained in Step 2, transform every $\mathcal{F}_k$ into a k-path subdigraph $\mathcal{T}_k$ of D as in the discussion above.

It is easy to check that the complexity of Algorithm $\mathcal{A}_{\mathcal{HPS}}$ is $O(|D|^{s+1})$. $\square$

Using Theorem 5.10.1, we will prove the following:

Theorem 5.10.2 (Bang-Jensen and Gutin) *[63] For a quasi-transitive digraph D on n vertices, the following two problems can be solved in time $O(n^5)$:*

(1) For every $k = 1, 2, \ldots, n$, find a heaviest k-path subdigraph of D.
(2) Find a heaviest cycle of D.

Let Ψ be the class of all transitive oriented graphs and all extended semicomplete digraphs. It follows from Theorem 4.8.5 that every quasi-transitive digraph is totally Ψ-decomposable. Thus, to prove the first part of Theorem 5.10.2, it suffices to show that every digraph $D \in \Psi$ satisfies the conditions of Theorem 5.10.1 with $s = 4$.

Proof of Part (a) of Theorem 5.10.2: Consider a digraph $D \in \Psi^{ext}$ on n vertices. We show that D satisfies the conditions of Theorem 5.10.1 with $s = 4$. A total Ψ-decomposition of D can be found in $O(n^4)$, see Section 4.12. For a non-negative integer k, let $w'_k(D)$ denote the weight of a heaviest k-path-cycle subdigraph of D.

Let D' be the digraph obtained from D by the vertex splitting procedure. In other words, we replace every vertex v of D by the arc $v'v''$ such that v'' dominates a vertex u' if and only if $v \rightarrow u$. Also, we define $w(v'v'') = w(v)$ for every $v \in V(D)$ and $w(v''u') = 0$ for every pair u, v of distinct vertices of D. Construct a network N_D as follows. Add a pair s, t of new vertices to D'. For each vertex v of D, we add the arcs (s, v') and (v'', t) to D'. Assign capacity one to each arc of N_D. Finally, assign cost zero to every arc adjacent to either s or t and cost $c(a) = -w(a)$ for each arc $a \in A(D')$.

By Exercise 3.64, we can find a maximum weight cycle subdigraph $\mathcal{L}$ in D' in time $O(n^3)$. Since s and t cannot be on any cycle in N_D, the digraph $\mathcal{L}$ corresponds to the minimum cost circulation f_0 in N_D (see Theorem 3.3.1). Starting from f_0 and using the buildup algorithm introduced in Section 3.10 we can construct, in time $O(n^4)$, minimum cost flows $f_1, \dots, f_n$ of values $1, \dots, n$ in N_D. By Theorem 3.3.1, every f_k is the sum of k flows of value 1 along paths from the source s to the sink t and a number of cycle flows. Hence, f_k provides a collection $\mathcal{F}_k$ of k paths and a number of cycles such that the paths and the cycles have no common vertices, except the source and the sink of the network. Moreover, by the definition of N_D, none of the cycles contain the source or the sink. It follows from the definition of N_D and the fact that f_k is a minimum cost flow in N_D that the paths and the cycles in $\{Q - \{s,t\} :\ Q \in \mathcal{F}_k\}$ form a heaviest k-path-cycle subdigraph $\mathcal{L}_k$ in D. In particular, $c(f_k) = -w'_k(D)$ for every $k = 1, \dots, n$.

If D is an extended semicomplete digraph then, by Theorem 5.7.1, for every $k = 1, \dots, n$, we can construct a k-path subdigraph $\mathcal{Q}_k$ so that $V(\mathcal{Q}_k) = V(\mathcal{L}_k)$. If D is acyclic then just let $\mathcal{Q}_k = \mathcal{L}_k$. Obviously, $\mathcal{Q}_k$ is a heaviest k-path subdigraph of D. Note that $\mathcal{Q}_1, \dots, \mathcal{Q}_n$ can be found in time $O(n^4)$. Since $w_k(D) = w'_k(D) = -c(f_k)$, it follows from Proposition 3.10.7 that (5.11) holds.

The proof that the complexity bound of $O(n^5)$ is left as Exercise 5.50. □

Proof of Part (b) of Theorem 5.10.2: Let D be a strong quasi-transitive digraph on $n \geq 2$ vertices and let $D = R[H_1, \dots, H_r]$, where R is semicomplete, $H_1, \dots, H_r$ are quasi-transitive digraphs and $r \geq 2$. (If D is not strong, then we consider the strong components of D one by one.) We claim that D has a heaviest cycle C containing vertices from more than one of the digraphs $H_1, \dots, H_r$. Indeed, let C' be a heaviest cycle of D completely contained in a H_i. Since D is strong, there is a path in D, of length at least 2, starting at a vertex x of C', terminating at a vertex y of C' and containing no other vertices from H_i. Hence, by the definition of $R[H_1, \dots, H_r]$, there is a path of length at least 2, starting at x, terminating at the successor x' of x (in C') and containing no other vertices from H_i. Clearly, the last path and C' minus the arc (x, x') form a cycle as desired.

Now it is easy to see the correctness of the following algorithm for finding a heaviest cycle of D. Note that our approach finds a heaviest cycle C which contains vertices from at least two H_i's. By the remark above this is also a heaviest cycle of D.

1. Solve the HPS problem for $H_1, \dots, H_r$ using Algorithm $\mathcal{A}_{\mathcal{HPS}}$.
2. Form D_0 with the weights $\widetilde{w}_{ij}$, as in the proof of Theorem 5.10.1, and the network N_{D_0}.
3. Construct a minimum cost circulation f_0 in N_{D_0}. Deleting the source and sink of N_{D_0}, form a heaviest cycle subdigraph $\mathcal{Z}$ of D_0.
4. Using Theorem 5.7.7, construct a heaviest cycle C of D_0 by merging the cycles in $\mathcal{Z}$.
5. Using the solutions of Step 1 and the cycle C, form a heaviest cycle of D (analogously to what we did in the proof of Theorem 5.10.1).

The proof that the complexity bound is $O(n^5)$ is left as Exercise 5.50. □

Theorem 5.10.2 implies the following:

Corollary 5.10.3 *[63] For a quasi-transitive digraph D on n vertices, the following problems can be solved in time $O(n^5)$.*

(a) Find a longest path of D.
(b) Find a longest cycle of D.
(c) For a set $X \subseteq V(D)$, check if D contains a cycle through X and construct one (if it exists).

Proof: Exercise 5.51. □

Theorem 5.10.2 can be generalized to the following result by Bang-Jensen and Gutin (see the definitions of Φ_i-decomposable digraphs in Section 4.12):

Theorem 5.10.4 *[62] Let D be a digraph of order n with non-negative weights on the vertices. Then*

(a) If D is totally Φ_0-decomposable, then for all $k = 1, \dots, n$, some maximum weight k-path subdigraphs of D can be found in time $O(n^5)$.
(b) If D is totally Φ_0-decomposable and $X \subseteq V(D)$, then we can check if D has a path covering all the vertices of X and find one (if it exists) in time $O(n^5)$.
(c) If D is totally Φ_2-decomposable, then a maximum weight cycle of D can be found in time $O(n^5)$.
(d) If D is totally Φ_2-decomposable and $X \subseteq V(D)$, then a cycle of D containing all vertices of X can be found in time $O(n^5)$ (if it exists).

(e) If D is totally Φ_1-decomposable, then a longest cycle of D can be found in time $O(n^5)$.

□

5.11 Hamilton Paths and Cycles in Various Classes of Digraphs

Grötschel and Harary [336] showed that only very few bridgeless graphs have the property that every strong orientation is hamiltonian.

Theorem 5.11.1 *[336] Let G be a bridgeless graph. If G is neither a cycle nor a complete graph, then G contains a strong non-hamiltonian orientation.*

□

However, there are quite a number of graphs with the property that every strong orientation is traceable.

Theorem 5.11.2 (Thomassen) *[699] Let G be a 2-edge-connected undirected graph such that every connected component of $\overline{G}$ is either bipartite or an odd cycle of length at least 5. Also assume that $\overline{G}$ has at most one non-bipartite component. Then every strong orientation of G is traceable.*

To prove Theorem 5.11.2, we need the following lemma whose proof is left as Exercise 5.49.

Lemma 5.11.3 *Let L be the complement of an odd cycle $u_1u_2\dots u_{2k+1}u_1$, $k \geq 2$, and let F be an orientation of L. Then, there are $i \neq j \in \{1, 2, \dots, 2k+1\}$ such that $u_iu_ju_{i+1}$ or $u_{i+1}u_ju_i$ is a path in F.* □

Proof of Theorem 5.11.2: Let $G_1, \dots, G_r$ be bipartite connected components of $\overline{G}$ such that A_i, B_i are partite sets of G_i, $i = 1, \dots, r$. Let $Z = u_1u_2\dots u_{2k+1}u_1$ be the odd cycle in $\overline{G}$, if one exists.

Let H be a strong orientation of G. Define a partition A, B of $V(G)$ as follows. Let $A^* = A_1 \cup \dots \cup A_r$ and $B^* = B_1 \cup \dots \cup B_r$. If Z does not exist (in $\overline{G}$), then $A = A^*$, $B = B^*$. Otherwise, by Lemma 5.11.3, without loss of generality, we have that there exists a j such that $u_1u_ju_2$ is a directed path in H. Let $A = A^* \cup \{u_3, u_5, \dots, u_{2k+1}\}$, $B = B^* \cup \{u_2, u_4, \dots, u_{2k}\} \cup \{u_1\}$. By this construction, $H\langle A\rangle$ is a tournament and $H\langle B\rangle$ is either a tournament (if Z does not exist) or H has a path xzy such that $x, y \in B$ and $xy \notin G\langle B\rangle$.

We now show that H has a cycle C including all vertices of A. If $H\langle A\rangle$ is strong, then C exists by Camion's theorem (see Theorem 1.5.2). If $H\langle A\rangle$ is not strong, then there is a shortest path P in H from the terminal strong component of $H\langle A\rangle$ to its initial strong component. Let P start at u and terminate at w. (Clearly, P does not have vertices other than u and w in these two components.) It is easy to check that $H\langle(A-V(P))\cup\{u,w\}\rangle$ has a hamiltonian (w,u)-path Q. The paths P and Q form a cycle containing A. Let C be a longest cycle containing A.

If $H-V(C)$ is a tournament, then some vertex of C dominates a vertex v of the initial strong component of $H-V(C)$. The tournament $H-V(C)$ has a hamiltonian path starting at v; this path can be extended to a hamiltonian path in H. Thus, we may assume that $H-V(C)$ is not a tournament. In particular, $x,y\in V(H)-V(C)$. Let $C=v_1v_2\dots v_mv_1$. We consider two cases.

Case 1: $z\in V(C)$. We first prove that C contains vertices v_i,v_{i+j} such that v_i dominates one of x,y and v_{i+j} is dominated by the other one and $1\le j\le m-1$. Since $\overline{G}$ has no triangles, each of z^+ and z^- is adjacent to at least one of x,y. By the maximality of C, if z^+ and y are adjacent, we must have $z^+\!\rightarrow\! y$ and then z,z^+ is the desired pair. Hence, we may assume that z^+ is adjacent to x and, hence, either z,z^+ is the desired pair or $z^+\!\rightarrow\! x$. Now considering z^- one can prove that either z^-,z is the desired pair or z^-,z^+ is the desired pair.

Among all pairs v_i,v_{i+j} satisfying the above property choose one such that j is the smallest possible. We may assume (by interchanging x and y if needed) that $v_i\!\rightarrow\! x$ and $y\!\rightarrow\! v_{i+j}$. We show that $j=1$. Assume that $j>1$. Because of the minimality of j, x is not dominated by v_{i+s} when $1\le s<j$ and because of the maximality of C, x does not dominate v_{i+1}. Hence, x is not adjacent to v_{i+1}. Similarly, we can see that y is not adjacent to v_{i+j-1} and none of the vertices v_{i+s}, $1\le s<j$, is dominated by y. Since $\overline{G}$ has no triangle, $j\ge 3$ and $v_{i+1}\!\rightarrow\! y$ and $x\!\rightarrow\! v_{i+j-1}$; a contradiction to the minimality of j. Thus, we may assume that $v_i\!\rightarrow\! x,\ y\!\rightarrow\! v_{i+1}$.

We add to the oriented graph $H-V(C)$ the arc yx obtaining a tournament T. Let v be a vertex in the initial strong component of T dominated by a vertex u in C. By Camion's theorem, T has a hamiltonian path P starting at v and terminating at some vertex w. If yx is not on P, then $C[u^+,u]P$ is a hamiltonian path of H. If yx is on P, then $P[v,y]C[v_{i+1},v_i]P[x,w]$ is a hamiltonian path of H.

Case 2: $z\not\in V(C)$. If $H-V(C)$ is strong, then we consider any arc of H between x and C (such an arc exists as the degree of x in $\overline{G}$ equals 2). If this arc starts (terminates) at x, we add to $H-V(C)$ the arc xy (yx) and consider a hamiltonian cycle in the resulting tournament. Using this together with C and the arc between x and C, it is easy to find a hamiltonian path in H.

So we assume that $H-V(C)$ is not strong. Let $H_1,H_2,\dots,H_p$ be an acyclic ordering of strong components of $H-V(C)$. We may assume without loss of generality (consider the converse of H if needed) that at most one of x,y belongs to $V(H_1)$. Clearly, some vertex v in H_1 is dominated by a vertex in C. We can find a hamiltonian path in H as in the case when $H-V(C)$

is a tournament unless for some i, $V(H_i) = \{x\}$ and $V(H_{i+1}) = \{y\}$ or $V(H_{i-1}) = \{y\}$. But this is impossible due to the existence of xzy. $\square$

In this theorem it is important that $\overline{G}$ does not contain a 3-cycle. Indeed, let M be a multipartite tournament consisting of a strong tournament T with fixed vertex y and triple x_1, x_2, x_3 of independent vertices such that $N^+(x_i) = \{y\}$ for every $i = 1, 2, 3$. Since $|N^+(\{x_1, x_2, x_3\})| < 2$ (see Exercise 3.61), M has no 1-path-cycle factor. (Recall that a multipartite tournament is traceable if and only if it has a 1-path-cycle factor, see Theorem 5.7.1.) However, Thomassen [699] remarks that Theorem 5.11.2 is perhaps far from being the best possible. He claims that by using the method of the proof of this theorem, it is not difficult to show that any strong orientation of a graph, whose complement is a disjoint union of two 5-cycles and independent vertices, has a hamiltonian path.

Problem 5.11.4 *Find a non-trivial extension of Theorem 5.11.2.*

We recall that a digraph D is unilateral if for every pair x, y of distinct vertices of D there is a path between x and y (not necessarily both (x, y)-path and (y, x)-path). For some of the graphs in Theorem 5.11.2 not only all strong orientations are traceable, but also all unilateral ones satisfy this property. This was shown by Fink and Lesniak-Foster in the following theorem.

Theorem 5.11.5 *[235] Let G be a graph and let $\mathcal{F} = Q_1 \cup \ldots \cup Q_k$ be a path subgraph of G in which every path Q_i is of length 1 or 2. Then an orientation of $G - \cup_{i=1}^{k} E(Q_i)$ is traceable if and only if it is unilateral.* $\square$

Erdős and Trotter [223] investigated when the Cartesian product of two directed cycles is hamiltonian. They proved the following (below gcd means the greatest common divisor):

Theorem 5.11.6 *Let $d = \gcd(k, m)$. The Cartesian product $\vec{C}_k \times \vec{C}_m$ is hamiltonian if and only if $d \geq 2$ and there exist positive integers d_1, d_2 such that $d_1 + d_2 = d$ and $\gcd(k, d_1) = \gcd(m, d_2) = 1$.* $\square$

For a generalization of Theorem 5.11.6, see Theorem 10.10.5.

In Section 4.6, we introduced de Bruijn digraphs $D_B(d, t)$, Kautz digraphs $D_K(d, t)$ as well as their generalizations: $D_G(d, n)$, $D_I(d, n)$, $D(d, n, q, r)$. (The digraphs $D(d, n, 1, r)$ are special circulant digraphs.) The consecutive-d digraphs $D(d, n, q, r)$ are the most general among the digraphs listed above. Thus, we restrict our attention to these digraphs. Du, Hsu and Hwang [206] proved the following result for digraphs $D(d, n, q, r)$.

Theorem 5.11.7 *If $\gcd(n, q) \geq 2$, or $\gcd(n, q) = 1$ and $q \geq 5$, then $D(d, n, q, r)$ is hamiltonian.* $\square$

Hwang [439] as well as Du and Hsu [205] characterized hamiltonian digraphs $D(d, n, q, r)$ for $\gcd(n, q) = 1$ and $d = 1$ ($d = 2$, respectively). Chang, Hwang and Tong [143] showed that every digraph $D(4, n, q, r)$ is hamiltonian. They also gave examples of digraphs $D(3, n, q, r)$, which are not hamiltonian [142].

We finish this chapter by the following result by Cooper, Frieze and Molloy. For a fixed integer r and a property P, we say that **almost all** r-regular digraphs satisfy P if the fraction of r-regular digraphs of order n with P (among all r-regular digraphs of order n) tends to 1 when $n \rightarrow \infty$.

Theorem 5.11.8 *[167] For a fixed integer $r \geq 3$, almost all r-regular digraphs are hamiltonian.* □

It is easy to show that almost all 1-regular digraphs are non-hamiltonian (Exercise 5.54). The fact that almost all 2-regular digraphs have no hamiltonian cycle follows directly from the fact that the expected number of hamiltonian cycles in a randomly and uniformly chosen 2-regular digraph tends to zero (for details see Section 3 of Chapter 4 in the book [14] by Alon and Spencer).

5.12 Exercises

5.1. **(+)** Let G_k be an undirected graph with vertex set $X \cup Z \cup Y$, where $X = \{x_1, x_2, \ldots, x_k\}$, $Y = \{y_1, y_2, \ldots, y_{k+1}\}$ and $Z = \{z_1, z_2, \ldots, z_{k+1}\}$, and edge set
$$\{xv : \ x \in X, v \in Y \cup Z\} \cup \{y_i z_i : \ i = 1, 2, \ldots, k+1\}.$$
Let $D_k = \overleftrightarrow{G}_k$. Prove that $qhn(D_k) = k$ (Gutin and Yeo [379]).

5.2. Prove Theorem 5.1.6.

5.3. Prove Theorem 5.1.7.

5.4. Let a digraph Z have $V(Z) = \{1, 2, \ldots, 6\}$ and $A(Z) = \{ij : \ j - i = 2 \text{ or } 3 \pmod 6\}$. Find qhn$(Z)$. Is Z hamiltonian?

5.5. **(+)** Prove without using Theorem 5.2.1 that every acyclic digraph D has an $\alpha(D)$-path factor. Hint: use Theorem 3.8.2.

5.6. **A refinement of the Gallai-Milgram theorem.** We say that a path P from x to y is **end-extendable** if there exists another path P' such that $P = P'[x, y]$. If no such path P' exists then P is **non-end-extendable**. Prove the following slight strengthening of the Gallai-Milgram theorem.

Proposition 5.12.1 *Every digraph D with independence number $\alpha(D) = \alpha$ has a path factor $P_1, P_2, \ldots, P_t$, $t \leq \alpha$, such that P_1 is a non-end-extendable path in D and P_i is a non-end-extendable path in $D - V(P_1) \cup \ldots \cup V(P_{i-1})$ for $2 \leq i \leq t$.*

Hint: show how to modify a given path factor into one with the property above.

5.7. Show that Theorem 5.2.5 implies Theorem 5.2.1.

5.8. **Scheduling airplanes.** An airport has a certain number of runways that can be used for landing of airplanes. How would you schedule airplanes to use the minimum number of the runways (in order to possibly have some spare ones permanently ready for emergency landings) if every use of a runway can be determined as a fixed time interval ?

5.9. **(−)** Show by examples that property (1) and (2) of Lemma 5.3.3 need not hold for arbitrary acyclic digraphs.

5.10. Using the proof of Theorem 5.4.2, Lemma 5.4.1 and Proposition 4.9.3, prove Corollary 5.4.3.

5.11. Prove Theorem 5.2.4 for path-mergeable digraphs.

5.12. Prove that every strong locally in-semicomplete digraph has a 2-connected underlying graph.

5.13. Give a direct proof of the following result. A locally semicomplete digraph has a hamiltonian cycle if and only if it is strong (Bang-Jensen [44]).

5.14. Give a direct proof of the following result. A locally semicomplete digraph has a hamiltonian path if and only if it is connected (Bang-Jensen [44]). Hint: use Lemma 4.10.2.

5.15. Give a direct proof of the following result. One can find a longest cycle is a semicomplete digraph in time $O(n^2)$ (Manoussakis [546]).

5.16. **(−)** Using Proposition 5.0.3 and Theorem 5.6.1 prove the following:

Proposition 5.12.2 *Let D be a digraph of order n. Suppose that, for every dominated pair of non-adjacent vertices $\{x, y\}$, either $d(x) \geq n-1$ and $d(y) \geq n-2$ or $d(x) \geq n-2$ and $d(y) \geq n-1$. Then D is traceable.*

5.17. Prove that the digraph Q_n introduced before Theorem 5.6.1 is strong and non-hamiltonian.

5.18. Prove Lemma 5.6.24.

5.19. Find an infinite family of hamiltonian digraphs that satisfy the conditions of both Theorem 5.6.1 and Theorem 5.6.5, but do not satisfy the conditions of Theorem 5.6.7 and are neither locally out-semicomplete nor locally in-semicomplete (Bang-Jensen, Gutin and Li [69]).

5.20. Find an infinite family of hamiltonian digraphs that satisfy the conditions of Theorem 5.6.12, but do not satisfy the conditions of Theorem 5.6.7 (Zhao and Meng [758]).

5.21. Prove Corollaries 5.6.21 and 5.6.22.

5.22. Using Meyniel's theorem, prove that if a strong digraph D has at least $n^2 - 3n + 5$ arcs, then D is hamiltonian (Lewin [514]).

5.23. Prove that every digraph with more than $(n-1)^2$ arcs is hamiltonian (Lewin [514]).

5.24. Prove that, if the minimum semi-degree of a digraph D of order n is at least $(n+1)/2$, then every arc of D is contained in a Hamilton cycle of D.

5.25. Construct an infinite family of non-hamiltonian strong digraphs that satisfy both (5.1) and (5.2) (Bermond and Thomassen [115]).

5.26. Prove that every vertex of a semicomplete multipartite digraph D belongs to a longest path in D (Volkmann [729]).

5.27. **(+)** Give a direct proof of the first (non-algorithmic) part of Theorem 5.7.1 (Gutin [358, 363]).

5.28. Show that the multipartite tournament in Figure 5.6 is non-hamiltonian.

5.29. Show that the analogue of Theorem 5.7.7 for semicomplete bipartite digraphs does not hold, i.e., there are a strong semicomplete bipartite digraph D and a maximum cycle subdigraph $\mathcal{F}$ in D such that $D\langle V(\mathcal{F})\rangle$ is not hamiltonian (Gutin [362]).

5.30. An oriented graph $D = (V, A)$ is an **arc-locally tournament digraph** if it has the following two properties:
(i) Whenever x, y are distinct vertices and there exists an arc $uv \in A$ such that $xu, yv \in A$, there is at least one arc between x and y in D.
(ii) Whenever x, y are distinct vertices and there exists an arc $zw \in A$ such that $zx, wy \in A$, there is at least one arc between x and y in D.
Prove that, if $D = (V, A)$ is a connected arc-local tournament digraph and C is a cycle, then every vertex of $V - C$ is adjacent to a vertex of C.

5.31. **(+) Hamiltonian paths and cycles in arc-locally tournament digraphs.** Prove the following two theorems by Bang-Jensen [48]:

Theorem 5.12.3 *An arc-locally tournament digraph is hamiltonian if and only if it is strong and has a cycle factor.*

Theorem 5.12.4 *A connected arc-locally tournament digraph is traceable if and only if it has a 1-path-cycle factor.*

Hint: use Exercise 5.30 and study the structure of the arcs between disjoint cycles.

5.32. **(−)** Arc-local tournament digraphs were defined above. Prove that every bipartite tournament is an arc-local tournament digraph.

5.33. Prove Theorem 5.7.13 by induction on t.

5.34. By inspecting all intermediate steps in the proof of Corollary 5.7.16, show that the following statement holds. Let D be a bipartite digraph obtained by taking two disjoint even cycles $C = u_1u_2 \dots u_{2k-1}u_{2k}u_1$ and $Z = v_1v_2 \dots v_{2r-1}v_{2r}v_1$ and adding an arc between v_{2i-1} and u_{2j} and between v_{2i} and u_{2j-1} (in any direction, possibly one in each direction) for all $i = 1, 2, \dots, k$ and $j = 1, 2, \dots, r$. D is hamiltonian if and only if it is strong. Moreover, if D is strong, then, given cycles C and Z as above, a hamiltonian cycle of D can be found in time $O(|V(C)||V(Z)|)$ (Gutin [362]).

5.35. Prove Theorem 5.7.7.

5.36. Prove the following generalization of Lemma 5.7.15. If a strong semicomplete multipartite digraph D has a cycle subdigraph $\mathcal{F} = C_1 \cup \dots \cup C_t$ with $p(\leq n)$ vertices, then, for every i, D has a cycle of length at least $p - t + 1$ covering all vertices of C_i (Bang-Jensen, Gutin and Huang [68]).

5.37. Construct an infinite family of semicomplete multipartite digraphs showing that the result of Exercise 5.36 is best possible (Bang-Jensen, Gutin and Huang [68]).

5.38. Using the result of Exercise 5.36, prove that every strong semicomplete multipartite digraph D with 1-path-cycle subdigraph $\mathcal{F} = P \cup C_1 \cup \ldots \cup C_t$ of order p has a path of length at least $p-t-1$ starting at the initial vertex of P (Bang-Jensen, Gutin and Huang [68]).

5.39. Prove Theorem 5.7.13.

5.40. Prove the following proposition. Let D be a strong semicomplete multipartite digraph of order n and let r be the cardinality of minimum partite set of D. If for each pair of dominated non-adjacent vertices x, y, $d(x) + d(y) \geq \min\{2(n-r)+3, 2n-1\}$, then D is hamiltonian (Zhou and Zhang [760]).

5.41. $(-)$Prove that every oriented graph of minimum in-degree and out-degree $k \geq 2$, on at most $2k+2$ vertices, is a multipartite tournament with at most two vertices in each partite set.

5.42. Prove the following theorem due to Jackson:

Theorem 5.12.5 *[449] Every oriented graph of minimum in-degree and out-degree $k \geq 2$, on at most $2k+2$ vertices, is hamiltonian.*

5.43. $(-)$ Check that $f(G'_k, k) = k-1$, where the digraph G'_k and the function f are introduced after Theorem 5.7.27.

5.44. Prove Theorem 5.8.2.

5.45. **Characterization of traceable extended locally semicomplete digraphs.** Prove Theorem 5.8.3.

5.46. $(-)$ Prove that the following problem is $\mathcal{NP}$-complete: Given a digraph $D = (V, A)$ and a partition $V = V_1 \cup \ldots \cup V_p$, check whether D has a cycle factor $C_1 \cup \ldots \cup C_k$ such that no cycle C_i is contained in a set V_j, $j = 1, 2, \ldots, p$.
Hint: consider an arbitrary vertex x in D and let $V_1 = V(D) - \{x\}, V_2 = \{x\}$.

5.47. $(-)$ **Characterization of traceable quasi-transitive digraphs.** Prove Theorem 5.9.2 using Theorem 5.7.1.
Hint: see the proof of Theorem 5.9.1.

5.48. $(-)$ **Another characterization of traceable quasi-transitive digraphs.** Formulate and prove a characterization of traceable quasi-transitive digraphs similar to Theorem 5.9.3.

5.49. Prove Lemma 5.11.3.

5.50. Prove the complexity bound for both parts of Theorem 5.10.2.

5.51. $(-)$ Deduce the results of Corollary 5.10.3 from Theorem 5.10.2.

5.52. Prove that if D is a strong oriented graph of order at least three and D does not contain, as induced subdigraph, any digraph in Figure 5.9, then D is hamiltonian (Kemnitz and Greger [477]).
Hint : show that D is locally out-semicomplete and use the characterization of hamiltonian locally out-semicomplete digraphs (Gutin and Yeo [380]).

Figure 5.9 Forbidden digraphs. Unoriented arcs can be oriented arbitrarily.

5.53. **A counterexample to a conjecture from** [477]. Consider the tournament D with $V(D) = \{x_1, x_2, x_3, x_4, x_5\}$ and

$$A(D) = \{x_1x_2, x_2x_3, x_3x_4, x_4x_5, x_5x_1, x_1x_3, x_2x_4, x_3x_5, x_4x_1, x_5x_2\}$$

and any 2-strong tournament T, containing three vertices y_1, y_2, y_3 such that

$$\{y_1y_2, y_2y_3, y_3y_1\} \subseteq A(T).$$

Let us construct an oriented graph T^* with vertex set $V(D) \cup V(T)$ and arc set

$$A(D) \cup A(T) \cup \{y_1x_2, x_4y_1, y_2x_2, x_4y_2, y_3x_4, x_2y_3\}.$$

Prove that

(a) T^* is strong.
(b) T^* does not contain, as induced subdigraph, any orientation of $K_{1,3}$.
(c) For every vertex v in T^*, $T^*\langle N(v)\rangle$ is strong.
(d) T^* is not hamiltonian.

(Gutin and Yeo [380])

5.54. **(−)** Prove that almost all 1-regular digraphs are non-hamiltonian.

5.55. **Connected (g, f)-factors in some semicomplete multipartite digraphs.** Given a digraph D and two positive integers $f(x), g(x)$ for every $x \in V(D)$, a subgraph H of D is called a **(g, f)-factor** if $g(x) \leq d^+_H(x) = d^-_H(x) \leq f(x)$ for every $x \in V(D)$. If $f(x) = g(x) = 1$ for every x, then a connected (g, f)-factor is a hamiltonian cycle. Prove the following result by Gutin [370]:

Theorem 5.12.6 *Let D be a semicomplete bipartite digraph or an extended locally in-semicomplete digraph. Then D has a connected (g, f)-factor if and only if D is strongly connected and contains a (g, f)-factor. One can check whether D has a connected (g, f)-factor in $O(n^3)$ time.* □

5.56. **Connected (g, f)-factors in quasi-transitive digraphs.** The additional terminology used in this exercise are introduced in the previous exercise. Prove the following assertion. The connected (g, f)-factor problem is polynomial time solvable for quasi-transitive digraphs (Gutin [370]).

5.57. Let G be the complete graph on 5 vertices with one edge deleted. Find a strong orientation of G which is not hamiltonian.

6. Hamiltonian Refinements

In this chapter we discuss results which in one way or another generalize the notion of hamiltonicity. As can be seen from the content of the chapter, there are quite a number of such topics. In fact many more could be added, but we feel that the ones included here are representative.

We start by studying hamiltonian paths with one or more end vertices prescribed, that is, we study paths which start in a prescribed vertex, paths which connect two prescribed vertices and finally paths which start and end in prescribed vertices. Not surprisingly, the level of difficulty of these problems increase when we fix more and more end vertices. Even for tournaments the last problem is still not completely solved.

The next topic is pancyclicity, which may be seen as a generalization of hamiltonicity. We first study digraphs of order n which have cycles of all lengths from 3 to n and subsequently digraphs in which every vertex is in a k-cycle for every $k \in \{3, 4, \ldots, n\}$. After that we discuss briefly arc-pancyclicity where we want cycles of all possible lengths from 3 to n through each arc. These problems are very hard and almost all known results deal with tournaments, generalizations of tournaments or digraphs which are almost complete.

Another topic covered is hamiltonian cycles which either avoid or contain certain prescribed arcs. These problems are very difficult even for tournaments. As we will show in Section 6.7, some of these results imply that the problem of deciding the existence of a hamiltonian cycle in a digraph obtained from a semicomplete digraph by adding just a few new vertices and some arcs is already very difficult. In fact the problem is highly non-trivial even if we add just one extra vertex. We also discuss various results concerning arc-disjoint hamiltonian paths and cycles, in particular the conjecture by Kelly that the arcs of every regular tournament can be decomposed into arc-disjoint hamiltonian cycles.

We then move on to orientations of hamiltonian cycles. We discuss in some detail one of the main tools in a recent proof by Havet and Thomassé of the deep result that every tournament on at least 8 vertices contains every orientation of a hamiltonian undirected path.

After this we briefly discuss another relative of the hamiltonian cycle problem; the problem of finding a set of few cycles that cover all vertices of a digraph. We study both the case when these cycles are allowed to intersect, pairwise but only in a path, and the case when we want the cycles to be disjoint.

The last two sections deal with applications. First we show that for every strong digraph D belonging to one of several classes of generalizations of tournaments, one can find a spanning subgraph which is strongly connected and has the minimum number of arcs among all such subdigraphs of D in polynomial time. For general digraphs this problem is $\mathcal{NP}$-complete since it generalizes the hamiltonian cycle problem. Finally we address the TSP problem and show that some widely used heuristics for the problem find tours which are better than a fraction (depending on n) of all possible tours, thus indicating that the solutions they find may be expected to be of reasonable quality.

6.1 Hamiltonian Paths with a Prescribed End-Vertex

We begin with hamiltonian paths starting or ending at a prescribed vertex. Besides being of independent interest, results of this type are also useful in connection with results on hamiltonian paths with both end vertices prescribed (but not necessarily the direction of the path).

To get a feeling for arguing with extended tournament structure, we start with the following easy result.

Proposition 6.1.1 *Suppose that a strong extended tournament D has an (x, y)-path P such that $D - P$ has a cycle factor. Then D has a hamiltonian path starting at x and a hamiltonian path ending at y.*

Proof: Choose a path P' starting at x as long as possible so that $D - P'$ has a factor which consists of minimal number of cycles $C_1, C_2, \ldots, C_q$. Then, by Proposition 5.7.18, we may assume that $C_i \Rightarrow C_j$ when $i < j$. Let $P' = u_1u_2 \ldots u_r$ where $u_1 = x$. If $q \neq 0$, then, by the assumption on P', u_r is completely dominated by C_1. Since D is strong, there is an arc from P' to C_1. Let u_i be the vertex of P' with largest index $i < r$ such that there is an arc u_iz from u_i to C_1. Let z^- be the predecessor of z on C_1. Since u_{i+1} has no arc to C_1, we obtain $z^- \rightarrow u_{i+1}$. Here we used the property that nonadjacent vertices of an extended semicomplete digraph are similar (defined in Chapter 1). Hence $C_1[z, z^-]$ can be inserted between u_i and u_{i+1}, contradicting the choice of P'. So $q = 0$ and P' is a hamiltonian path starting at x. A similar argument can be applied to show that D has a hamiltonian path ending at y. □

The following result, due to Bang-Jensen and Gutin, shows that, for digraphs that are either semicomplete bipartite or extended locally semicomplete, there is a nice necessary and sufficient condition for the existence of a hamiltonian path starting at a prescribed vertex.

Theorem 6.1.2 *[66] Let $D = (V, A)$ be a digraph which is either semicomplete bipartite or extended locally out-semicomplete and let $x \in V$. Then D has a hamiltonian path starting at x if and only if D contains a 1-path-cycle factor $\mathcal{F}$ of D such that the path of $\mathcal{F}$ starts at x, and, for every vertex y of $V - \{x\}$, there is an (x, y)-path[1] in D. Moreover, if D has a hamiltonian path starting at x, then, given a 1-path-cycle factor $\mathcal{F}$ of D such that the path of $\mathcal{F}$ starts at x, the desired hamiltonian path can be found in time $O(n^2)$.*

Proof: As the necessity is clear, we will only prove the sufficiency. Suppose that $\mathcal{F} = P \cup C_1 \cup ... \cup C_t$ is a 1-path-cycle factor of D that consists of a path P starting at x and cycles C_i, $i = 1, ..., t$. Suppose also that every vertex of D is reachable from x. Then, without loss of generality, there is a vertex of P that dominates a vertex of C_1. Let $P = x_1x_2 \dots x_p$, $C_1 = y_1y_2 \dots y_qy_1$, where $x = x_1$ and $x_k \rightarrow y_s$ for some $k \in \{1, 2, \dots, p\}$, $s \in \{1, 2, \dots, q\}$. We show how to find a new path starting at x which contains all the vertices of $V(P) \cup V(C_1)$. Repeating this process we obtain the desired path. Clearly, we may assume that $k < p$ and that x_p has no arc to $V(C_1)$.

Assume first that D is an extended locally out-semicomplete digraph. If P has a vertex x_i which is similar to a vertex y_j in C_1, then $x_iy_{j+1}, y_jx_{i+1} \in A$ and using these arcs we see that $P[x_1, x_i]C[y_{j+1}, y_j]P[x_{i+1}, x_p]$ is a path starting from x and containing all the vertices of $P \cup C_1$. If P has no vertex that is similar to a vertex in C_1, then we can apply the result of Exercise 4.37 to $P[x_k, x_p]$ and $x_kC_1[y_s, y_{s-1}]$ and merge these two paths into a path R starting from x_k and containing all the vertices of $P[x_k, x_p] \cup C_1$. Now, $P[x_1, x_{k-1}]R$ is a path starting at x and containing all the vertices of $P \cup C_1$.

Suppose now that D is semicomplete bipartite. Then either $y_{s-1} \rightarrow x_{k+1}$, which implies that $P[x_1, x_k]C_1[y_s, y_{s-1}]P[x_{k+1}, x_p]$ is a path starting at x and covering all the vertices of $P \cup C_1$, or $x_{k+1} \rightarrow y_{s-1}$. In the latter case, we consider the arc between x_{k+2} and y_{s-2}. If $y_{s-2} \rightarrow x_{k+2}$ we can construct the desired path, otherwise we continue to consider arcs between x_{k+3} and y_{s-3} and so on. If we do not construct the desired path in this way, then we find that the last vertex of P dominates a vertex in C_1, contradicting our assumption above.

Using the process above and breadth-first search, one can construct an $O(n^2)$-algorithm for finding the desired hamiltonian path starting at x. □

Just as the problem of finding a minimum path factor generalizes the hamiltonian path problem, we may generalize the problem of finding a hamiltonian path starting at a certain vertex to the problem of finding a path factor with as few paths as possible such that one of these paths starts at a specified vertex x. We say that a path factor **starts at *x*** if one of its paths starts at x and denote by $\mathrm{pc}_x(D)$ the minimum number of paths in a path factor that

[1] This is equivalent to saying that D has an out-branching with root x.

starts at x. The problem of finding a path factor with $\mathrm{pc}_x(D)$ paths which starts at x in a digraph D is called the **PFx** problem[2].

Let Φ_1 be the union of all semicomplete bipartite, extended locally semicomplete and acyclic digraphs. Using an approach similar to that taken in Section 5.10, Bang-Jensen and Gutin proved the following.

Theorem 6.1.3 *[66] Let D be a totally Φ_1-decomposable digraph. Then the PFx problem for D can be solved in time $O(|V(D)|^4)$.* □

6.2 Weakly Hamiltonian-Connected Digraphs

Recall that an $[x, y]$-path in a digraph $D = (V, A)$ is a path which either starts at x and ends at y or oppositely. We say that D is **weakly hamiltonian-connected** if it has a hamiltonian $[x, y]$-path (also called an **$[x, y]$-hamiltonian path**) for every choice of distinct vertices $x, y \in V$. Obviously deciding whether a digraph contains an $[x, y]$-hamiltonian path for some x, y is not easier than determining whether D has any hamiltonian path and hence for general digraphs this is an $\mathcal{NP}$-complete problem by Theorem 5.0.2 (see also Exercise 6.3). In this section we discuss various results that have been obtained for generalizations of tournaments. All of these results imply polynomial algorithms for finding the desired paths.

6.2.1 Results for Extended Tournaments

We start with a theorem due to Thomassen [698] which has been generalized to several classes of generalizations of tournaments as will be seen in the following subsections.

Theorem 6.2.1 *[698] Let $D = (V, A)$ be a tournament and let x_1, x_2 be distinct vertices of D. Then D has an $[x_1, x_2]$-hamiltonian path if and only if none of the following holds.*

(a) D is not strong and either none of x_1, x_2 belongs to the initial strong component of D or none of x_1, x_2 belongs to the terminal strong component (or both).
(b) D is strong and for $i = 1$ or 2, $D - x_i$ is not strong and x_{3-i} belongs to neither the initial nor the terminal strong component of $D - x_i$.
(c) D is isomorphic to one of the two tournaments in Figure 6.1 (possibly after interchanging the names of x_1 and x_2).

The following easy corollary is left as Exercise 6.4:

[2] Observe that $\mathrm{pc}_x(D) \leq \mathrm{pc}(D) + 1$ holds for every digraph D.

x_1

x_2

Figure 6.1 The exceptional tournaments in Theorem 6.2.1. The edge between x_1 and x_2 can be oriented arbitrarily.

Corollary 6.2.2 *[698] Let D be a strong tournament and let x, y, z be distinct vertices of D. Then D has a hamiltonian path connecting two of the vertices in the set $\{x, y, z\}$.* $\square$

Thomassen [698] used a nice trick in his proof of Theorem 6.2.1 by using Corollary 6.2.2 in the induction proof. We will give his proof below.

Proof of Theorem 6.2.1: Let x_1, x_2 be distinct vertices in a tournament D. It is easy to check that if any of (a)-(c) holds, then there is no $[x_1, x_2]$-hamiltonian path in D.

Suppose now that none of (a)-(c) hold. We prove by induction on n that D has an $[x_1, x_2]$-hamiltonian path. This is easy to show when $n \leq 4$ so assume now that $n \geq 5$ and consider the induction step with the obvious induction hypothesis. If D is not strong then let $D_1, D_2, \ldots, D_s$, $s \geq 2$ be the acyclic ordering of the strong components of D. Since (a) does not hold, we may assume without loss of generality that $x_1 \in V(D_1)$ and $x_2 \in V(D_s)$. Observe that D_1 has a hamiltonian path P_1 starting at x_1 (Exercise 6.1) and D_s has a hamiltonian path P_s ending at x_2. Let P_i be a hamiltonian path in D_i for each $i = 2, 3, \ldots, s-1$. Then $P_1 P_2 \ldots P_{s-1} P_s$ is an (x_1, x_2)-hamiltonian path.

If $D - x_i$ is not strong for $i = 1$ or 2, then we may assume without loss of generality that $i = 1$. Let $D'_1, \ldots, D'_p$, $p \geq 2$ be the acyclic ordering of the strong components of $D - x_1$. Since (b) does not hold we may assume, by considering the converse of D if necessary, that x_2 belongs to D'_p. Let y be any out-neighbour of x_1 in D'_1. Our argument for the previous case implies that there is a (y, x_2)-hamiltonian path P in $D - x_1$, implying that $x_1 P$ is an (x_1, x_2)-hamiltonian path in D. Hence we may assume that $D - x_i$ is strong for $i = 1, 2$.

If $D - \{x_1, x_2\}$ is not strong, then it is easy to prove that D has an (x_i, x_{3-i})-hamiltonian path for $i = 1, 2$ (Exercise 6.2). Hence we only need to consider the case when $D' = D - \{x_1, x_2\}$ is strong. Let $u_1 u_2 \dots u_{n-2} u_1$ be a hamiltonian cycle of D'. By considering the converse if necessary, we may assume that x_1 dominates u_1. Then D has an (x_1, x_2)-hamiltonian path unless x_2 dominates u_{n-2} so we may assume that is the case. By the same argument we see that either the desired path exists or x_1 dominates u_{n-3} and x_2 dominates u_{n-4}. Now it is easy to see that either the desired path exists, or $n-2$ is even and we have $x_1 \mapsto \{u_1, u_3, \dots, u_{n-3}\}$, $x_2 \mapsto \{u_2, u_4, \dots, u_{n-2}\}$. If x_1 or x_2 dominates any vertex other than those described above, then by repeating the argument above we see that either the desired path exists or $\{x_1, x_2\} \mapsto V(C)$, which is impossible since D is strong. Hence we may assume that

$$\begin{aligned} \{u_2, u_4, \dots, u_{n-2}\} &\mapsto x_1 \mapsto \{u_1, u_3, \dots, u_{n-3}\}, \\ \{u_1, u_3, \dots, u_{n-3}\} &\mapsto x_2 \mapsto \{u_2, u_4, \dots, u_{n-2}\} \end{aligned} \tag{6.1}$$

If $n = 6$, then using that (c) does not hold, it is easy to see that the desired path exists. So we may assume that $n \geq 8$. By induction, the theorem and hence also Corollary 6.2.2 holds for all tournaments on $n-2$ vertices. Thus D' has a hamiltonian path P which starts and ends in the set $\{u_1, u_3, u_5\}$ and by (6.1), P can be extended to an (x_1, x_2)-hamiltonian path of D. □

We now turn to extended tournaments. An extended tournament D does not always have a hamiltonian path, but, as we saw in Theorem 5.7.1, it does when the following obviously necessary condition is satisfied: there is a 1-path-cycle factor in D. Thus if we are looking for a sufficient condition for the existence of an $[x, y]$-hamiltonian path, we must require the existence on an $[x, y]$-path P such that $D - P$ has a cycle factor (this includes the case when P is already hamiltonian). Checking for such a path factor in an arbitrary digraph can be done in polynomial time using flows, see Exercise 3.62.

The next result is similar to the structure we found in the last part of the proof of Theorem 6.2.1.

Lemma 6.2.3 *[67] Suppose that D is a strong extended tournament containing two adjacent vertices x and y such that $D - \{x, y\}$ has a hamiltonian cycle C but D has no hamiltonian $[x, y]$-path. Then C is an even cycle, $N^+(x) \cap V(C) = N^-(y) \cap V(C)$, $N^-(x) \cap V(C) = N^+(y) \cap V(C)$, and the neighbours of x alternate between in-neighbours and out-neighbours around C.*

Proof: Exercise 6.5. □

Bang-Jensen, Gutin and Huang obtained the following characterization for the existence of an $[x, y]$-hamiltonian path in an extended tournament. Note the strong similarity with Theorem 6.2.1.

Theorem 6.2.4 *[67] Let D be an extended tournament and x_1, x_2 be distinct vertices of D. Then D has an $[x_1, x_2]$-hamiltonian path if and only if D has an $[x_1, x_2]$-path P such that $D - P$ has a cycle factor and D does not satisfy any of the conditions below:*

(a) D is not strong and either the initial or the terminal component of D (or both) contains none of x_1 and x_2;
(b) D is strong and the following holds for $i = 1$ or $i = 2$: $D - x_i$ is not strong and either x_{3-i} belongs to neither the initial nor the terminal component of $D - x_i$, or x_{3-i} does belong to the initial (terminal) component of $D - x_i$ but there is no (x_{3-i}, x_i)-path $((x_i, x_{3-i})$-path$)$ P' such that $D - P'$ has a cycle factor.
(c) $D, D - x_1$, and $D - x_2$ are all strong and D is isomorphic to one of the tournaments in Figure 6.1. □

The proof of this theorem in [67] is constructive and implies the following result (the proof is much more involved than that of Theorem 6.2.1). We point out that the proof in [67] makes explicit use of the fact that the digraphs have no 2-cycles. Hence the proof is only valid for extended tournaments and not for general extended semicomplete digraphs, for which the problem is still open.

Theorem 6.2.5 *[67] There exists an $O(\sqrt{n}m)$ algorithm to decide if a given extended tournament has a hamiltonian path connecting two specified vertices x and y. Furthermore, within the same time bound a hamiltonian $[x, y]$-path can be found if it exists.* □

Theorem 6.2.4 implies the following characterization of extended tournaments which are weakly hamiltonian-connected (see Exercise 6.7).

Theorem 6.2.6 *[67] Let D be an extended tournament. Then D is weakly hamiltonian-connected if and only if it satisfies each of the conditions below.*

(a) D is strongly connected.
(b) For every pair of distinct vertices x and y of D, there is an $[x, y]$-path P such that $D - P$ has a cycle factor.
(c) For each vertex x of D, $D - x$ has at most two strong components and if $D - x$ is not strong, then for each vertex y in the initial (respectively terminal) strong component, there is a (y, x)-path (respectively an (x, y)-path) P' such that $D - P'$ has a cycle factor.
(d) D is not isomorphic to any of the two tournaments in Figure 6.1. □

The following result generalizes Corollary 6.2.2. Note that we must assume the existence of the paths described below in order to have any chance of having a hamiltonian path with end vertices in the set $\{x, y, z\}$. The proof below illustrates how to argue with extended tournament structure.

Corollary 6.2.7 *[67] Let x, y and z be three vertices of a strong extended tournament D. Suppose that, for every choice of distinct vertices $u, v \in \{x, y, z\}$, there is a $[u, v]$-path P in D so that $D - P$ has a cycle factor. Then there is a hamiltonian path connecting two of the vertices in $\{x, y, z\}$.*

Proof: If both $D - x$ and $D - y$ are strong, then, by Theorem 6.2.4, either D has a hamiltonian path connecting x and y, or D is isomorphic to one of the tournaments in Figure 6.1, in which case there is a hamiltonian path connecting x and z. There is a similar argument if both $D - x$ and $D - z$, or $D - y$ and $D - z$ are strong. So, without loss of generality, assume that neither $D - x$ nor $D - y$ is strong. Let $S_1, S_2, \ldots, S_t$ be an acyclic ordering of the strong components of $D - x$. Note that S_t has an arc to x, since D is strong.

Suppose first that $y \in V(S_i)$ for some $1 < i < t$. We show that this implies that $D - y$ is strong, contradicting our assumption. Consider an $[x, y]$-path P and a cycle factor $\mathcal{F}$ of $D - P$. It is easy to see that P cannot contain any vertex of $S_{i+1}, \ldots, S_t$. Hence each of these strong components contains a cycle factor consisting of those cycles from $\mathcal{F}$ that are in S_j for $j = i+1, \ldots, t$. In particular (since it contains a cycle), each S_j has size at least 3 for $j = i+1, \ldots, t$. It also follows from the existence of P and $\mathcal{F}$ that every vertex in S_i is dominated by at least one vertex from $U = V(S_1) \cup \ldots \cup V(S_{i-1})$. Indeed, if some vertex $z \in V(S_i)$ is not dominated by any vertex from U, then using that $S_r \Rightarrow S_p$ for all $1 \leq r < p \leq t$ we get that z is similar to all vertices in U. However, this contradicts the existence of P and $\mathcal{F}$. Now it is easy to see that $D - y$ is strong since every vertex of $S_i - y$ is dominated by some vertex from $V(S_1) \cup \ldots \cup V(S_{i-1})$ and dominates a vertex in $V(S_{i+1}) \cup \ldots \cup V(S_t)$. Hence we may assume that y belongs to S_1 or S_t.

By considering the converse of D if necessary, we may assume that $y \in V(S_1)$. By Theorem 6.2.4(b) we may assume that there is no (y, x)-path W such that $D - W$ has a cycle factor. Thus it follows from the assumption of the corollary that there is an (x, y)-path $P' = v_1 v_2 \ldots v_r$, $v_1 = x, v_r = y$ such that $D - P'$ has a cycle factor $\mathcal{F}'$. Since $P' - x$ is contained in S_1, we can argue as above that each S_i, $i > 1$, has a cycle factor (inherited from $\mathcal{F}'$) and hence each S_i contains a hamiltonian cycle C_i, by Theorem 5.7.7.

Note that every vertex of S_1 which is not on P' belongs to some cycle of $\mathcal{F}'$ that lies entirely inside S_1. Hence, if $r = 2$ (that is, P' is just the arc $x \to y$), then it follows from Proposition 6.1.1 (which is also valid when the path in question has length zero) that S_1 contains a hamiltonian path starting at y. This path can easily be extended to a (y, x)-hamiltonian path in D, since each S_i, $i > 1$, is hamiltonian. Thus we may assume that $r \geq 3$.

If $S_1 - y$ is strong then $D - y$ is strong, contradicting our assumption above. Let $T_1, T_2, \ldots, T_s$, $s \geq 2$, be an acyclic ordering of the strong components of $S_1 - y$. Note that each $V(T_i)$ is either covered by some cycles from the cycle factor $\mathcal{F}'$ of $D - P'$ and hence T_i has a hamiltonian cycle (by Theorem 5.7.5), or is covered by a subpath of $P'[v_2, v_{r-1}]$ and some cycles (possibly zero) from $\mathcal{F}'$ and hence T_i has a hamiltonian path (by Theorem 5.7.1). Note also that there is at least one arc from y to T_1 and at least one arc from T_s to y. If T_1 contains a portion of $P'[v_2, v_{r-1}]$, then it is clear that T_1 contains v_2. But then $D - y$ is strong since $x \to v_2$, contradicting our assumption. So T_1 contains no vertices of $P'[v_1, v_{r-1}]$ and hence, by the remark above, T_1 has a hamiltonian cycle to which there is at least one arc from y. Using the structure derived above, it is easy to show that D has a (y, x)-hamiltonian path (Exercise 6.6). □

It can be seen from the results above that, when we consider weak hamiltonian-connectedness, extended tournaments have a structure which is closely related to that of tournaments. To see that Theorem 6.2.4 does not extend to general multipartite tournaments, consider the multipartite tournament D obtained from a hamiltonian bipartite tournament B with classes X and Y, by adding two new vertices x and y along with the following arcs: all arcs from x to X and from Y to x, all arcs from y to Y and X to y and an arc between x and y in any direction. It is easy to see that D satisfies none of the conditions (a)-(c) in Theorem 6.2.4, yet there can be no hamiltonian path with end vertices x and y in D because any such path would contain a hamiltonian path of B starting and ending in X or starting and ending in Y. Such a path cannot exist for parity reasons ($|X| = |Y|$). Note also that we can choose B so that the resulting multipartite tournament is highly connected.

Bang-Jensen and Manoussakis [86] characterized weakly hamiltonian-connected bipartite tournaments. In particular, they proved a necessary and sufficient condition for the existence of an $[x, y]$-hamiltonian path in a bipartite tournament. The statement of this characterization turns out to be quite similar to that of Theorem 6.2.4. The only difference between the statements of these two characterizations is in Condition (c): in the characterization for bipartite tournaments the set of forbidden digraphs is absolutely different and moreover infinite.

6.2.2 Results for Locally Semicomplete Digraphs

Our next goal is to describe the solution of the $[x, y]$-hamiltonian path problem for locally semicomplete digraphs. Notice that this solution also covers the case of semicomplete digraphs and so, in particular, it generalizes Theorem 6.2.1 to semicomplete digraphs.

We start by establishing notation for some special locally semicomplete digraphs. Up to isomorphism there is a unique strong tournament with four vertices. We denote this by T_4^1. It has the following vertices and arcs:

$$V(T_4^1) = \{a_1, a_2, a_3, a_4\}, A(T_4^1) = \{a_1a_2, a_2a_3, a_3a_4, a_4a_1, a_1a_3, a_2a_4\}.$$

The semicomplete digraphs T_4^2, T_4^3, and T_4^4 are obtained from T_4^1 by adding some arcs, namely:

$$A(T_4^2) = A(T_4^1) \cup \{a_3a_1, a_4a_2\},$$

$$A(T_4^3) = A(T_4^1) \cup \{a_3a_1\}, A(T_4^4) = A(T_4^1) \cup \{a_1a_4\}.$$

Let $\mathcal{T}_4 = \{T_4^1, T_4^2, T_4^3, T_4^4\}$. It is easy to see that every digraph of $\mathcal{T}_4$ has a unique hamiltonian cycle and has no hamiltonian path between two vertices which are not consecutive on this hamiltonian cycle (such two vertices are called **opposite**).

Let $\mathcal{T}_6$ be the set of semicomplete digraphs with the vertex set $\{x_1, x_2, a_1, a_2, a_3, a_4\}$, each member D of $\mathcal{T}_6$ has a cycle $a_1a_2a_3a_4a_1$ and the digraph $D\langle\{a_1, a_2, a_3, a_4\}\rangle$ is isomorphic to one member of $\mathcal{T}_4$, in addition, $x_i \rightarrow \{a_1, a_3\} \rightarrow x_{3-i} \rightarrow \{a_2, a_4\} \rightarrow x_i$ for $i = 1$ or $i = 2$. It is straightforward to verify that $\mathcal{T}_6$ contains only two tournaments (denoted by T_6' and T_6''), namely the ones shown in Figure 6.1, and that $|\mathcal{T}_6| = 11$. Since none of the digraphs of $\mathcal{T}_4$ has a hamiltonian path connecting any two opposite vertices, no digraph of $\mathcal{T}_6$ has a hamiltonian path between x_1 and x_2.

For every even integer $m \geq 4$ there is only one 2-strong, 2-regular locally semicomplete digraph on m vertices, namely the second power $\vec{C}_m^2$ of an m-cycle (Exercise 6.8). We define

$$\mathcal{T}^* = \{\ \vec{C}_m^2 \mid m \text{ is even and } m \geq 4\}.$$

It is not difficult to prove that every digraph of $\mathcal{T}^*$ has a unique hamiltonian cycle and is not weakly hamiltonian-connected (Exercise 6.9, see also [47]). For instance, if the unique hamiltonian cycle of $\vec{C}_6^2$ is denoted by $u_1u_2u_3u_4u_5u_6u_1$, then $u_1u_3u_5u_1$ and $u_2u_4u_6u_2$ are two cycles of $\vec{C}_6^2$ and there is no hamiltonian path between any two vertices of $\{u_1, u_3, u_5\}$ or of $\{u_2, u_4, u_6\}$.

Let T_8^1 be the digraph consisting of $\vec{C}_6^2$ together with two new vertices x_1 and x_2 such that $x_1 \rightarrow \{u_1, u_3, u_5\} \rightarrow x_2 \rightarrow \{u_2, u_4, u_6\} \rightarrow x_1$. Furthermore, T_8^2 (T_8^3, respectively) is defined as the digraph obtained from T_8^1 by adding the arc x_1x_2 (the arcs x_1x_2 and x_2x_1, respectively). Let $\mathcal{T}_8 = \{T_8^1, T_8^2, T_8^3\}$. It is easy to see that every element of $\mathcal{T}_8$ is a 3-strong locally semicomplete digraph and has no hamiltonian path between x_1 and x_2.

Before we present the main result, we state the following two lemmas that were used in the proof of Theorem 6.2.10 by Bang-Jensen, Guo and Volkmann in [56]. The first lemma generalizes the structure found in the last part of the proof of Theorem 6.2.1.

Lemma 6.2.8 *[56] Let D be a strong locally semicomplete digraph on $n \geq 4$ vertices and x_1, x_2 two distinct vertices of D. If $D - \{x_1, x_2\}$ is strong, and $N^+(x_1) \cap N^+(x_2) \neq \emptyset$ or $N^-(x_1) \cap N^-(x_2) \neq \emptyset$, then D has a hamiltonian path connecting x_1 and x_2.*

Proof: Exercise 6.10. □

Another useful ingredient in the proof of Theorem 6.2.10 is the following linking result. An **odd chain** is the second power, $\vec{P}^2_{2k+1}$ for some $k \geq 1$, of a path on an odd number of vertices.

Lemma 6.2.9 *[56] Let D be a connected, locally semicompletedigraph with $p \geq 4$ strong components and acyclic ordering $D_1, D_2, ..., D_p$ of these. Suppose that $V(D_1) = \{u_1\}$ and $V(D_p) = \{v_1\}$ and that $D - x$ is connected for every vertex x. Then for every choice of $u_2 \in V(D_2)$ and $v_2 \in V(D_{p-1})$, D has two vertex disjoint paths P_1 from u_2 to v_1 and P_2 from u_1 to v_2 with $V(P_1) \cup V(P_2) = V(D)$ if and only if D is not an odd chain from u_1 to v_1.*

Proof: If D is an odd chain, it is easy to see that D has no two vertex disjoint (u_i, v_{3-i})-path for $i = 1, 2$ (Exercise 6.11). We prove by induction on p that the converse is true as well. Suppose that D is not an odd chain from u_1 to v_1. Since the subdigraph $D - x$ is connected for every vertex x, $|N^+(D_i)| \geq 2$ for all $i \leq p-2$ and $|N^-(D_j)| \geq 2$ for all $j \geq 3$. If $p = 4$, then it is not difficult see that D has two vertex disjoint paths P_1 from u_2 to v_1 and P_2 from u_1 to v_2 with $V(P_1) \cup V(P_2) = V(D)$ (Exercise 6.13). If $p = 5$, it is also not difficult to check that D has the desired paths, unless D is a chain on five vertices. So we assume that $p \geq 6$. Now we consider the digraph D', which is obtained from D by deleting the vertex sets $\{u_1, v_1\}$, $V(D_2 - u_2)$ and $V(D_{p-1} - v_2)$.

Using the assumption on D, it is not difficult to show that D' is a connected, but not strongly connected locally semicompletedigraph with the acyclic ordering $\{u_2\}, D_3, D_4, \ldots, D_{p-2}, \{v_2\}$ of its strong components. Furthermore, for every vertex y of D', the subdigraph $D' - y$ is still connected. Let u be an arbitrary vertex of D_3 and v an arbitrary vertex of D_{p-2}. Note that there is a (u_1, u)-hamiltonian path P in $D\langle\{u_1, u\} \cup V(D_2 - u_2)\rangle$ and similarly there is a (v, v_1)-hamiltonian path Q in $D\langle\{v, v_1\} \cup V(D_{p-1} - v_2)\rangle$. Hence if D' has disjoint (u_2, v)-, (u, v_2)-paths which cover all vertices of D', then D has the desired paths. So we can assume D' has no such paths. By induction, D' is an odd chain from u_2 to v_2. Now using that D is not an odd chain from u_1 to v_1 it is easy to see that D has the desired paths. We leave the details to the reader. □

A weaker version of Lemma 6.2.9 was proved in [47, Theorem 4.5].

Below we give a characterization, due to Bang-Jensen, Guo and Volkmann for the existence of an $[x, y]$-hamiltonian path in a locally semicomplete digraph. Note again the similarity to Theorem 6.2.1.

Theorem 6.2.10 *[56] Let D be a connected locally semicomplete digraph on n vertices and x_1 and x_2 be two distinct vertices of D. Then D has no hamiltonian $[x_1, x_2]$-path if and only if one of the following conditions is satisfied:*

(1) D is not strong and either the initial or the terminal component of D (or both) contains none of x_1, x_2.
(2) D is strongly connected, but not 2-strong,
  (2.1) there is an $i \in \{1,2\}$ such that $D - x_i$ is not strong and x_{3-i} belongs to neither the initial nor the terminal component of $D - x_i$;
  (2.2) $D - x_1$ and $D - x_2$ are strong, s is a separating vertex of D, $D_1, D_2, ..., D_p$ is the acyclic ordering of the strong components of $D - s$, $x_i \in V(D_\alpha)$ and $x_{3-i} \in V(D_\beta)$ with $\alpha \leq \beta - 2$. Furthermore, $V(D_{\alpha+1}) \cup V(D_{\alpha+2}) \cup ... \cup V(D_{\beta-1})$ contains a separating vertex of D, or $D' = D\langle V(D_\alpha) \cup V(D_{\alpha+1}) \cup ... \cup V(D_\beta)\rangle$ is an odd chain from x_i to x_{3-i} with $N^-(D_{\alpha+2}) \cap V(D - V(D')) = \emptyset$ and $N^+(D_{\beta-2}) \cap V(D - V(D')) = \emptyset$.
(3) D is 2-strong and is isomorphic to T_4^2 or to one member of $\mathcal{T}_6 \cup \mathcal{T}_8 \cup \mathcal{T}^*$ and x_1, x_2 are the corresponding vertices in the definitions. □

As an easy consequence of Theorem 6.2.10, we obtain a characterization of weakly hamiltonian-connected locally semicomplete digraphs. The proof is left to the interested reader as Exercise 6.12.

Theorem 6.2.11 *[56] A locally semicomplete digraph D with at least three vertices is weakly hamiltonian–connected if and only if it satisfies (a), (b) and (c) below:*

(a) D is strong,
(b) the subdigraph $D - x$ has at most two components for each vertex x of D,
(c) D is not isomorphic to any member of $\mathcal{T}_6 \cup \mathcal{T}_8 \cup \mathcal{T}^*$. □

6.3 Hamiltonian-Connected Digraphs

We now turn to hamiltonian paths with specified initial and terminal vertices. An (x, y)**-hamiltonian path** is a hamiltonian path from x to y. Clearly, asking for such a path in an arbitrary digraph is an even stronger requirement than asking for an $[x, y]$-hamiltonian path[3]. A digraph $D = (V, A)$ is **hamiltonian-connected** if D has an (x, y)-hamiltonian path for every choice of distinct vertices $x, y \in V$.

[3] We know of no class of digraphs for which the $[x, y]$-hamiltonian path problem is polynomially solvable, but the (x, y)-hamiltonian path problem is $\mathcal{NP}$-complete. For arbitrary digraphs they are equivalent from a complexity point of view (see Exercise 6.3).

No characterization for the existence of an (x, y)-hamiltonian path is known even for the case of tournaments[4]. Note however, that we sketch a polynomial algorithm for the problem in the next section, so in the algorithmic sense a good characterization does exist. The following very important partial result due to Thomassen will be used in the algorithm of the next section.

Theorem 6.3.1 (Thomassen) *[698] Let $D = (V, A)$ be a 2-strong semicomplete digraph with distinct vertices x, y. Then D contains an (x, y)-hamiltonian path if either (a) or (b) below is satisfied.*

(a) D contains three internally disjoint (x, y)-paths each of length at least two,

(b) D contains a vertex z which is dominated by every vertex of $V - x$ and D contains two internally disjoint (x, y)-paths each of length at least two.

□

In his proof Thomassen explicitly uses the fact that the digraph is allowed to have cycles of length 2. This simplifies the proof (which is still far from trivial), since one can use contraction to reduce to a smaller instance and then use induction.

An important ingredient in the proof of Theorem 6.3.1 as well as in several other proofs concerning the existence of an (x, y)-hamiltonian path in a semicomplete digraph D is to prove that D contains a spanning acyclic graph in which x can reach all other vertices and y can be reached by all other vertices. The reason for this can be seen from the following result which generalizes an observation by Thomassen in [698].

Proposition 6.3.2 *[50] Let D be a path-mergeable digraph. Then D has a hamiltonian (x, y)-path if and only if D contains a spanning acyclic digraph H in which $d^-_H(x) = d^+_H(y) = 0$ and so that, for every vertex $z \in V(D)$, H contains an (x, z)-path and a (z, y)-path.*

Proof: Exercise 6.15. □

Theorem 6.3.1 and Menger's theorem (see Theorem 7.3.1) immediately imply the following result. For another nice consequence see Exercise 6.16.

Theorem 6.3.3 *[698] If a semicomplete digraph D is 4-strong, then D is hamiltonian-connected.* □

Thomassen constructed an infinite family of 3-strongly connected tournaments with two vertices x, y for which there is no (x, y)-hamiltonian path [698]. Hence, from a connectivity point of view, Theorem 6.3.3 is the best possible.

---

[4] By this we mean a structural characterization involving only conditions that can be checked in polynomial time.

Theorem 6.3.3 is a very important result with several consequences. Thomassen has shown in several papers how to use Theorem 6.3.3 to obtain results on spanning collections of paths and cycles in semicomplete digraphs. See e.g. the papers [699, 701] by Thomassen and also Section 6.7. The following extension of Theorem 6.3.3 to extended tournaments has been conjectured by Bang-Jensen, Gutin and Huang:

Conjecture 6.3.4 *[67] If D is a 4-strong extended tournament with an (x, y)-path P such that $D - P$ has a cycle factor, then D has an (x, y)-hamiltonian path.*

Extending Theorem 6.3.3 to locally semicomplete digraphs, Guo [342] proved the following:

Theorem 6.3.5 (Guo) *[342] Let D be a 2-strong locally semicomplete digraph and let x, y be two distinct vertices of D. Then D contains a hamiltonian path from x to y if (a) or (b) below is satisfied.*

(a) There are three internally disjoint (x, y)-paths in D, each of which is of length at least 2 and D is not isomorphic to any of the digraphs T_8^1 and T_8^2 (see the definition in the preceding section).
(b) The digraph D has two internally disjoint (x, y)-paths P_1, P_2, each of which is of length at least 2 and a path P which either starts at x, or ends at y and has only x or y in common with P_1, P_2 such that $V(D) = V(P_1) \cup V(P_2) \cup V(P)$. Furthermore, for any vertex $z \notin V(P_1) \cup V(P_2)$, z has a neighbour on $P_1 - \{x, y\}$ if and only if it has a neighbour on $P_2 - \{x, y\}$. $\square$

Since neither of the two exceptions in (a) is 4-strong, Theorem 6.3.5 implies the following:

Corollary 6.3.6 *[342] If a locally semicomplete digraph is 4-strong, then it is hamiltonian-connected.* $\square$

In [341] Guo used Theorem 6.3.5 to give a complete characterization of those 3-strongly connected arc-3-cyclic (that is, every arc is in a 3-cycle) locally tournament digraphs with no hamiltonian path from x to y for specified vertices x and y. In particular this characterization shows that there exist infinitely many 3-strongly connected digraphs which are locally tournament digraphs (but not semicomplete digraphs) and are not hamiltonian-connected. Thus, as far as this problem is concerned, it is not only the subclass of semicomplete digraphs which contain difficult instances within the class of locally semicomplete digraphs. It should be noted that Guo's proof does not rely on Theorem 6.3.3. However, due to the non-semicomplete exceptions mentioned above, it seems unlikely that a much simpler proof of Corollary 6.3.6 can be found using Theorem 6.3.3 and Theorem 4.11.15.

Not surprisingly, there are also several results, such as the following by Lewin, on hamiltonian-connectivity in digraphs with many arcs.

Theorem 6.3.7 *[514] Every digraph on $n \geq 3$ vertices and at least $(n-1)^2+1$ arcs is hamiltonian-connected.* □

If a digraph D is hamiltonian-connected, then D is also hamiltonian (since every arc is in a hamiltonian cycle). The next result, due to Bermond, shows that we only need a slight strengthening of the degree condition in Theorem 5.6.3 to get a sufficient condition for strong hamiltonian-connectivity.

Theorem 6.3.8 *[108] Every digraph D on n vertices which satisfies $\delta^0(D) \geq \frac{n+1}{2}$ is hamiltonian-connected.* □

If we just ask for weakly hamiltonian-connectness then Overbeck-Larisch showed that we can replace the condition on the semi-degrees by a condition on the degrees:

Theorem 6.3.9 *[597] Every 2-strong digraph on n vertices and minimum degree at least $n+1$ is weakly hamiltonian-connected.* □

Thomassen asked whether all 3-strong digraphs $D = (V, A)$ on n vertices with $d^+(x) + d^-(x) \geq n+1$ for all $x \in V$ are necessarily hamiltonian-connected. However, this is not the case, as was shown by Darbinyan [179].

6.4 Finding a Hamiltonian (x, y)-Path in a Semicomplete Digraph

In this section we discuss algorithmic aspects of the (x, y)-hamiltonian path problem for semicomplete digraphs. The main result is the following by Bang-Jensen, Manoussakis and Thomassen:

Theorem 6.4.1 *[87] The (x, y)-hamiltonian path problem is polynomially solvable for semicomplete digraphs.* □

We will not give the proof of this difficult result here, but rather outline the most interesting ingredients in the non-trivial proof in [87]. As usual, we will always use n to denote the number of vertices of the digraph in question.

The first lemma is quite simple to prove, but it turns out to be very useful for the design of the algorithm of Theorem 6.4.1.

If x, w, z are distinct vertices of a digraph D, then we use the notation $Q_{x,z}$, $Q_{.,w}$ to denote two disjoint paths such that the first path is an (x, z)-path, the second path has terminal vertex w, and $V(Q_{x,z}) \cup V(Q_{.,w}) = V(D)$. Similarly $Q_{z,x}$ and $Q_{w,.}$ denote two disjoint paths, such that the first path is a (z, x)-path, the second path has initial vertex w, and $V(Q_{z,x}) \cup V(Q_{w,.}) = V(D)$.

Lemma 6.4.2 *[87] Let x, w, z be distinct vertices in a semicomplete digraph T, such that there exist internally disjoint (x, w)-, (x, z)-paths P_1, P_2 in T. Let $R = T - V(P_1) \cup V(P_2)$.*

(a) There are either $Q_{x,w}$, $Q_{\cdot,z}$ or $Q_{x,z}$, $Q_{\cdot,w}$ in T, unless there is no arc from R_t to $V(P_1) \cup V(P_2) - x$, where R_t is the terminal component of $T\langle R\rangle$.

(b) In the case when there is an arc from R_t to $V(P_1) \cup V(P_2) - x$ we can find one of the pairs of paths, such that the path with only one end vertex specified has length at least one, unless $V(P_1) \cup V(P_2) = \{w, x, z\}$.

(c) Moreover there is an $O(n^2)$ algorithm to find one of the pairs of paths above if they exist.

Proof: If $R = \emptyset$ then both pairs of paths exist. Hence we may assume that $R \neq \emptyset$. Assume there is an arc uv where $u \in R_t$ and $v \in (V(P_1) \cup V(P_2)) - x$. Assume without loss of generality that $v \in P_1$. Since $u \in R_t$, $T\langle R\rangle$ has a hamiltonian path Q ending at u and starting at some vertex y. By Proposition 4.10.2, the semicomplete digraph $T\langle R \cup V(P_1) - x\rangle$ has a hamiltonian path starting either at y or the successor of x on P_1 and ending in w. This path together with P_2 forms the desired pair of paths $Q_{x,z}, Q_{\cdot,w}$. This proves (a). It is easy to verify (b) by the same argument. As the strong components of $T\langle R\rangle$ and a hamiltonian cycle in each of them can be found in $O(n^2)$ time (Theorem 5.5.2), we can find Q and $Q_{x,z}, Q_{\cdot,w}$ in $O(n^2)$ time. □

We point out that the proof above shows that Lemma 6.4.2 is valid also for digraphs that are locally in-semicomplete.

The following lemma allows one to use symmetry and thereby reduces the number of cases to consider when looking for an (x, y)-hamiltonian path.

Lemma 6.4.3 *Let T be a semicomplete digraph and x, y vertices of T, such that there exist 2 internally disjoint (x, y)-paths and an (x, y)-separator $\{u, v\}$ in T. Suppose that u, v do not induce a 2-cycle, say, $v \not\rightarrow u$. Let T' denote the semicomplete digraph obtained from T, by adding the arc $v \rightarrow u$. Then T has an (x, y)-hamiltonian path if and only if T' has an (x, y)-hamiltonian path.*

Proof: Exercise 6.18. □

The next result shows that either T is 2-strong or we can reduce the problem to smaller instances.

Lemma 6.4.4 *[87] If T is not 2-strong then either the desired path exists in T, or we can reduce the problem to one or two smaller problems, such that in the latter case the total size of the subproblems is at most $n + 1$.* □

We now outline the major steps of the algorithm in [87] for the (x, y)-hamiltonian path problem. First we make some assumptions which do not change the problem.

We assume that there is no arc from x to y and that neither x nor y are contained in a 2-cycle (if there is such a cycle containing x (y), then delete the arc entering x (leaving y)). It is easy to see that the new semicomplete digraph has an (x, y)-hamiltonian path if and only if the original digraph has one. So we assume that the input is a semicomplete digraph T which has the form above. In order to refer to smaller versions of the same problem we refer to the problem as the **hamiltonian problem**. Note that by Lemma 6.4.4 we may assume that T is 2-strong (otherwise we just consider smaller subproblems).

With the assumptions above it follows from Theorem 6.3.1 that, if there are three internally disjoint (x, y)-paths in T, then the desired hamiltonian path exists. Thus, by Lemma 6.4.4, the interesting part is when T is 2-strong and there are two but not three internally disjoint (x, y)-paths. By Menger's theorem (which we study in Chapter 7) we may thus assume that there exists an (x, y)-separator of size two in T.

The next theorem by Bang-Jensen, Manoussakis and Thomassen generalizes Theorem 6.3.1. It is very important for the proof of Theorem 6.4.1, because it corresponds to a case when no reduction is possible (see the description of the algorithm below) and hence one has to prove the existence of the desired path directly. Recall that for specified distinct vertices s, t, an (s, t)-separator is a subset $S \subseteq V - \{s, t\}$ such that $D - S$ has no (s, t)-path. An (s, t)-separator is **trivial** if either s has out-degree zero or t has in-degree zero in $D - S$.

Theorem 6.4.5 *[87] Let T be a 2-strong semicomplete digraph on at least 10 vertices and let x, y be vertices of T such that $y \mapsto x$. Suppose that $T - x, T - y$ are both 2-strong. If all (x, y)-separators consisting of two vertices (if any exist) are trivial, then T has an (x, y)-hamiltonian path.* $\square$

Besides the results mentioned above the algorithm uses the following results:

Lemma 6.4.6 *[87] Suppose T is 2-strong and there exists a non-trivial separator $\{u, v\}$ of x, y. Let A, B denote a partition of $T - \{u, v\}$ such that $y \in A, x \in B$ and $A \mapsto B$. Let $T' = T\langle A \cup \{u, v\}\rangle, T'' = T\langle B \cup \{u, v\}\rangle$. We can reduce the hamiltonian problem to at most four hamiltonian problems such that one has size* $\max\{|A|, |B|\} + 2$ *or* $\max\{|A|, |B|\} + 3$ *and the others (if any) have size at most* $\min\{|A|, |B|\} + 3$. $\square$

Lemma 6.4.7 *[87] Suppose that T is 2-strong, $n \geq 6$, and all (x, y)-separators of size 2 x, y are trivial. If $T - x$ or $T - y$ is not 2-strong, then either the desired path exists in T, or we can reduce the problem to one or two smaller problems, such that in the latter case, the total size of the subproblems is at most $n + 2$.* $\square$

The hamiltonian algorithm

1. If $n \leq 9$, then settle the problem in constant time.
2. If T is not 2-strong, then using Lemma 6.4.4 we settle the problem, or reduce to smaller instances of the hamiltonian problem.

3. If there are no (x, y)-separators of size 2, then T has the desired path, by Theorem 6.3.1.
4. If all (x, y)-separators of size 2 are trivial, we check if $T - x$ and $T - y$ are 2-strong. Then we settle or reduce the problem using Theorem 6.4.5 or Lemma 6.4.7.
5. Let $\{u, v\}$ be a non-trivial (x, y)-separator and let A, B form a partition of $T - \{u, v\}$, such that $y \in A, x \in B$ and $A \mapsto B$. (Such a partition can be found in time $O(n^2)$, by letting B be the vertices which in $T - \{u, v\}$ can be reached from x by a directed path and then taking $A = V - B - \{u, v\}$.) Also, if necessary, add an arc to make u, v induce a 2-cycle. This does not change the problem, by Lemma 6.4.3.
6. Use the algorithmic version of Lemma 6.4.2 to find $Q_{x,u}$, $Q_{.,v}$ or $Q_{x,v}$, $Q_{.,u}$ in $T'' = T(B \cup \{u, v\})$, and use an analogous algorithm to find $Q_{u,y}$, $Q_{v,.}$ or $Q_{v,y}$, $Q_{u,.}$ in $T' = T(A \cup \{u, v\})$. These paths exist, since T is 2-strong, and the paths with one end vertex unspecified can be chosen of length at least one, since A, B both have size at least 2 (here we used that $\{u, v\}$ is a non-trivial separator).
7. If these paths match then T has the desired (x, y)-hamiltonian path. So suppose (by renaming u, v if necessary) that we find $Q_{x,u}$, $Q_{.,v}$ in T'' and $Q_{u,y}$, $Q_{v,.}$ in T'.
8. Using Lemma 6.4.6 we can now reduce the problem to smaller instances of the hamiltonian problem.

In Step 7 we say that the two sets of paths in T'' and T' **match** if the following holds: the paths are P_1 from x to w and P_2 from p to z in T'' and R_1 from r to y and R_2 from s to q in T' where $\{w, z\} = \{r, s\} = \{u, v\}$ and $w = s$ and $z = r$. In this case the path $P_1 R_2 P_2 R_1$ is the desired hamiltonian path since $q \rightarrow p$ by the definition of B in Step 5.

The complexity of the algorithm outlined above is $O(n^5)$ (in fact, it is $O(n^{4+\epsilon})$ for every $\epsilon > 0$). No attempt was made in [87] to improve the complexity, but it seems quite difficult to improve it very much.

It is interesting to note that the algorithm described above cannot be easily modified to solve the problem of finding the longest path with specified initial and terminal vertex in a semicompletedigraph. In several places we explicitly use that we are searching for a hamiltonian path. There also does not seem to be any simple reduction of this problem to the problem of deciding the existence of a hamiltonian path from x to y.

Conjecture 6.4.8 *[65] There exists a polynomial algorithm which, given a semicomplete digraph D and two distinct vertices x and y of D, finds a longest (x, y)-path.*

Note that, if we ask for the longest $[x, y]$-path in a tournament, then this can be answered using Theorem 6.2.1 (see Exercise 6.19).

Conjecture 6.4.9 *[65] There exists a polynomial algorithm which, given a digraph D that is either extended semicomplete or locally semicomplete, and two distinct vertices x and y of D, decides whether D has an (x,y)-hamiltonian path and finds such a path if one exists.*

6.5 Pancyclicity of Digraphs

A digraph D of order n is **pancyclic** if it has cycles of all lengths $3, 4, \dots, n$. We say that D is **vertex-pancyclic** if for any $v \in V(D)$ and any $k \in \{3, 4, ..., n\}$ there is a cycle of length k containing v. We also say that D is **(vertex-)m-pancyclic** if D contains a k-cycle (every vertex of D is on a k-cycle) for each $k = m, m+1, \dots, n$. Note that some early papers on pancyclicity in digraphs require that D is (vertex-)2-pancyclic in order to be (vertex-)pancyclic (see e.g. the survey [115] by Bermond and Thomassen). We feel that this definition is too restrictive, since often one can prove pancyclicity results for much broader classes of digraphs when the 2-cycle is omitted from the requirement.

6.5.1 (Vertex-)Pancyclicity in Degree-Constrained Digraphs

The following claim is due to Alon and Gutin:

Lemma 6.5.1 *[11] Every directed graph $D = (V, A)$ on n vertices for which $\delta^0(D) \geq n/2 + 1$ is vertex-2-pancyclic.*

Proof: Let $v \in V$ be arbitrary. By Corollary 5.6.3 there is a Hamilton cycle $u_1 u_2 \dots u_{n-1} u_1$ in $D - v$. If there is no cycle of length k through v then for every i, $|N^+(v) \cap \{u_i\}| + |N^-(v) \cap \{u_{i+k-2}\}| \leq 1$, where the indices are computed modulo $n-1$. By summing over all values of i, $1 \leq i \leq n-1$, we conclude that $|N^-(v)| + |N^+(v)| \leq n-1$, contradicting the assumption that all in-degrees and out-degrees exceed $n/2$. □

Thomassen [696] proved that just by adding one to the degree condition for hamiltonicity in Theorem 5.6.7 one obtains cycles of all possible lengths in the digraphs satisfying the degree condition.

Theorem 6.5.2 *[696] Let D be a strong digraph on n vertices such that $d(x) + d(y) \geq 2n$ whenever x and y are nonadjacent. Then either D has cycles of all lengths $2, 3, \dots, n$, or D is a tournament (in which case it has cycles of all lengths $3, 4, \dots, n$) or n is even and D is isomorphic to $\overleftrightarrow{K}_{\frac{n}{2}, \frac{n}{2}}$.* □

The following example from [696] shows that $2n$ cannot be replaced by $2n-1$ in Theorem 6.5.2. For some $m \leq n$ let $D_{n,m} = (V, A)$ be the digraph with vertices $V = \{v_1, v_2, \ldots, v_n\}$ and arcs $A = \{v_iv_j | i < j \text{ or } i = j+1\} - \{v_iv_{i+m-1} | 1 \leq i \leq n-m+1\}$. We leave it as Exercise 6.20 to show that $D_{n,m}$ is strong, has no m-cycle and if $m > (n+1)/2$, then $D_{n,m}$ satisfies Meyniel's condition for hamiltonicity (Theorem 5.6.7). In [176] Darbinyan characterizes those digraphs which satisfy Meyniel's condition, but are not pancyclic.

Theorem 6.5.2 extends Moon's theorem (Theorem 1.5.1) and Corollaries 5.6.2 and 5.6.6. However, as pointed out by Bermond and Thomassen in [115], Theorem 6.5.2 does not imply Meyniel's theorem (Theorem 5.6.7). The following result is due to Häggkvist:

Theorem 6.5.3 *[391] Every hamiltonian digraph on n vertices and at least $\frac{1}{2}n(n+1) - 1$ arcs is pancyclic.* □

Song [679] generalized the result of Jackson given in Theorem 5.12.5 and proved the following theorem.

Theorem 6.5.4 *[679] Let $D = (V, A)$ be an oriented graph on $n \geq 9$ vertices with minimum degree $n - 2$. Suppose that D satisfies the following property:*

$$xy \notin A \Rightarrow d^+(x) + d^-(y) \geq n - 3. \tag{6.2}$$

Then D is pancyclic. □

Song [679] pointed out that, if the minimum degree condition in Theorem 6.5.4 is relaxed, then it is no longer guaranteed that D is hamiltonian.

Using Theorem 6.5.4 and Theorem 10.7.3, Bang-Jensen and Guo proved that under the same conditions as in Theorem 6.5.4 the digraph is in fact vertex-pancyclic.

Theorem 6.5.5 *[54] Let D be an oriented graph on $n \geq 9$ vertices and suppose that D satisfies the conditions in Theorem 6.5.4. Then D is vertex pancyclic.* □

It should be noted that every digraph which satisfies the condition of Theorem 6.5.4 is a multipartite tournament with independence number at most 2.

There are several other results on pancyclicity of digraphs with large minimum degrees, see e.g. the papers [174, 175, 178] by Darbinyan.

6.5.2 Pancyclicity in Extended Semicomplete and Quasi-Transitive Digraphs

In this subsection we show how to use the close relationship between the class of quasi-transitive digraphs and the class of extended semicomplete digraphs to derive results on pancyclic and vertex-pancyclic quasi-transitive digraphs from analogous results for extended semicomplete digraphs.

A digraph D is **triangular with partition** V_0, V_1, V_2, if the vertex set of D can be partitioned into three disjoint sets V_0, V_1, V_2 with $V_0 \mapsto V_1 \mapsto V_2 \mapsto V_0$. Note that this is equivalent to saying that $D = \vec{C}_3[D\langle V_0\rangle, D\langle V_1\rangle, D\langle V_2\rangle]$.

Gutin [367] characterized pancyclic and vertex-pancyclic extended semicomplete digraphs. Clearly no extended semicomplete digraph of the form $D = \vec{C}_2[\overline{K}_{n_1}, \overline{K}_{n_2}]$ with at least 3 vertices is pancyclic since all cycles are of even length. Hence we must assume that there are at least 3 partite sets in order to get a pancyclic extended semicomplete digraph. It is also easy to see that the (unique) strong 3-partite extended semicomplete digraph on 4 vertices is not pancyclic (since it has no 4-cycle). These observations and the following theorem completely characterize pancyclic and vertex-pancyclic extended semicomplete digraphs.

Theorem 6.5.6 *[367] Let D be a hamiltonian extended semicomplete digraph of order $n \geq 5$ with k partite sets ($k \geq 3$). Then*

1. *(a) D is pancyclic if and only if D is not triangular with a partition V_0, V_1, V_2, two of which induce digraphs with no arcs, such that either $|V_0| = |V_1| = |V_2|$ or no $D\langle V_i\rangle$ ($i = 0, 1, 2$) contains a path of length 2.*
2. *(b) D is vertex-pancyclic if and only if it is pancyclic and either $k > 3$ or $k = 3$ and D contains two cycles Z, Z' of length 2 such that $Z \cup Z'$ has vertices in the three partite sets.* □

It is not difficult to see that Theorem 6.5.6 extends Theorem 1.5.1, since no semicomplete digraph on $n \geq 5$ vertices satisfies any of the exceptions from (a) and (b).

The next two lemmas by Bang-Jensen and Huang [79] concern cycles in triangular digraphs. They are used in the proof of Theorem 6.5.9 which characterizes pancyclic and vertex-pancyclic quasi-transitive digraphs.

Lemma 6.5.7 *[79] Suppose that D is a triangular digraph with a partition V_0, V_1, V_2 and suppose that D is hamiltonian. If $D\langle V_1\rangle$ contains an arc xy and $D\langle V_2\rangle$ contains an arc uv, then every vertex of $V_0 \cup \{x, y, u, v\}$ is on cycles of lengths $3, 4, \ldots, n$.*

Proof: Let C be a hamiltonian cycle of D. We construct an extended semicomplete digraph D' from D in the following way. For each of $i = 0, 1, 2$, first path-contract[5] each maximal subpath of C which is contained in $D\langle V_i\rangle$ and then delete the remaining arcs of $D\langle V_i\rangle$. It is clear that D' is a subdigraph of D, and in this process, C is changed to a hamiltonian cycle C' of D'. Hence D' is also triangular with a partition V_0', V_1', V_2' such that $|V_0'| = |V_1'| = |V_2'| = r$, for some r (the last fact follows from the existence of a hamiltonian cycle in D'). Then each vertex of D is on a cycle of length k with $3r \leq k \leq |V(D)|$ (to see this, just use suitable pieces of the r subpaths of C in each V_i).

[5] Recall the definition of path-contraction from Subsection 5.1.1.

Now we may assume that $r \geq 2$ and we show that each vertex of $V_0 \cup \{x, y, u, v\}$ is on a cycle of length k with $3 \leq k \leq 3r-1$. To see this, we modify D' to another digraph D'' as follows. If x and y are in distinct maximal subpaths P_x, P_y of C in $D\langle V_1 \rangle$, then we add (in D') an arc from the vertex to which P_x was contracted to the vertex to which P_y was contracted. If x and y are in the same maximal subpath P of C in $D\langle V_1 \rangle$, then we add (in D') an arc from the vertex to which P was contracted to an arbitrary other vertex of V_1'. For the vertices u and v we make a similar modification. Hence we obtain a digraph D'' which is isomorphic to a subdigraph of D. The digraph D'' is also triangular with a partition V_0'', V_1'', V_2'' such that $|V_0''| = |V_1''| = |V_2''| = r$. Moreover $D''\langle V_1'' \rangle$ contains an arc $x'y'$ and $D''\langle V_2'' \rangle$ contains an arc $u'v'$. It is clear now that each vertex of $V_0'' \cup \{x', y', u', v'\}$ is on a cycle of length k where $3 \leq k \leq 3r - 1$. Using the same structure as for these cycles we can see that in D each vertex of $V_0 \cup \{x, y, u, v\}$ is on a cycle of length k with $3 \leq k \leq 3r - 1$. □

Lemma 6.5.8 *[79] Suppose that D is a triangular digraph with a partition V_0, V_1, V_2 and D has a hamiltonian cycle C. If $D\langle V_0 \rangle$ contains an arc of C and a path P of length 2, then every vertex of $V_1 \cup V_2 \cup V(P)$ is on cycles of lengths $3, 4, \ldots, n$.*

Proof: Exercise 6.24. □

It is easy to check that a strong quasi-transitive digraph on 4 vertices is pancyclic if and only if it is a semicomplete digraph. For $n \geq 5$ we have the following characterization due to Bang-Jensen and Huang:

Theorem 6.5.9 *[79] Let $D = (V, A)$ be a hamiltonian quasi-transitive digraph on $n \geq 5$ vertices.*

1. *(a) D is pancyclic if and only if it is not triangular with a partition V_0, V_1, V_2, two of which induce digraphs with no arcs, such that either $|V_0| = |V_1| = |V_2|$, or no $D\langle V_i \rangle$ $(i = 0, 1, 2)$ contains a path of length 2.*
2. *(b) D is not vertex-pancyclic if and only if D is not pancyclic or D is triangular with a partition V_0, V_1, V_2 such that one of the following occurs:*
   - *(b1) $|V_1| = |V_2|$, both $D\langle V_1 \rangle$ and $D\langle V_2 \rangle$ have no arcs, and there exists a vertex $x \in V_0$ such that x is not contained in any path of length 2 in $D\langle V_0 \rangle$ (in which case x is not contained in a cycle of length 5).*
   - *(b2) one of $D\langle V_1 \rangle$ and $D\langle V_2 \rangle$ has no arcs and the other contains no path of length 2, and there exists a vertex $x \in V_0$ such that x is not contained in any path of length 1 in $D\langle V_0 \rangle$ (in which case x is not contained in a cycle of length 5).*

Proof: To see the necessity of the condition in (a), suppose that D is triangular with a partition V_0, V_1, V_2, two of which induce digraphs with no arcs. If $|V_0| = |V_1| = |V_2|$, then D contains no cycle of length $n - 1$. If no $D\langle V_i \rangle$

$(i = 0, 1, 2)$ contains a directed path of length 2, then D contains no cycle of length 5.

Now we prove the sufficiency of the condition in (a). According to Theorem 4.8.5, there exists a semicomplete digraph T on k vertices for some $k \geq 3$ such that D is obtained from T by substituting a quasi-transitive digraph H_v for each vertex $v \in V(T)$ (here H_v is non-strong if it has more than one vertex). Let C be a hamiltonian cycle of D. We construct an extended semicomplete digraph D' from D in the following way: for each $H_v, v \in V(T)$, first path-contract each maximal subpath of C which is contained in H_v and then delete the remaining arcs of H_v. In this process C is changed to a hamiltonian cycle C' of D'.

Suppose D is not pancyclic. Then it is easy to see that D' is not pancyclic. By Theorem 6.5.6, D' is triangular with a partition V'_0, V'_1, V'_2. Let $V_i \subset V$ be obtained from V'_i, $i = 0, 1, 2$, by substituting back all vertices on contracted subpaths of C. Then D is triangular with partition V_0, V_1, V_2. Moreover each $D\langle V_i \rangle$ is covered by r disjoint subpaths of C for some r.

By Lemma 6.5.7, two of V_0, V_1, V_2, say V_1 and V_2, induce subdigraphs with no arcs in D. If $|V_0| = |V_1| = |V_2|$ we have the first exception in (a). Hence we may assume that $|V_0| > |V_1| = |V_2|$. Then $D\langle V_0 \rangle$ contains an arc of C. From Lemma 6.5.8, we see that $D\langle V_0 \rangle$ contains no path of length 2. This completes the proof of (a).

The proof of (b) is left to the reader as Exercise 6.25. □

6.5.3 Pancyclic and Vertex-Pancyclic Locally Semicomplete Digraphs

We saw in the last subsection how the structure theorem for quasi-transitive digraphs (i.e., Theorem 4.8.5) was helpful in finding a characterization for (vertex-)pancyclic quasi-transitive digraphs. Now we show that the structure theorem for locally semicomplete digraphs (Theorem 4.11.15) is also very useful for finding a characterization of those locally semicomplete digraphs which are (vertex-)pancyclic. Our first goal (Lemma 6.5.13) is a characterization of those round decomposable locally semicomplete digraphs which are (vertex-)pancyclic.

Lemma 6.5.10 *Let R be a strong round local tournament and let C be a shortest cycle of R and suppose C has $k \geq 3$ vertices. Then for every round labelling $v_0, v_1, \ldots, v_{n-1}$ of R such that $v_0 \in V(C)$ there exist indices $0 < a_1 < a_2 < \ldots < a_{k-1} < n$ so that $C = v_0 v_{a_1} v_{a_2} \ldots v_{a_{k-1}} v_0$.*

Proof: Let C be a shortest cycle and let $\mathcal{L} = v_0, v_1, \ldots, v_{n-1}$ be a round labelling of R so that $v_0 \in V(C)$. If the claim is not true, then there exists a number $2 \leq l < k-1$ so that $C = v_0 v_{a_1} v_{a_2} \ldots v_{a_{k-1}} v_0$, where $0 < a_1 < \ldots < a_{l-1}$ and $a_l < a_{l-1}$. Now the fact that $\mathcal{L}$ is a round labelling of R implies that $v_{l-1} \rightarrow v_0$, contradicting the fact that C is a shortest cycle. □

Recall that the girth $g(D)$ of a digraph is the length of a shortest cycle in $D = (V, A)$. For a vertex $v \in V$ we let $g_v(D)$ denote the length of a shortest cycle in D that contains v. The next lemma shows that every round local tournament R is $g(R)$-pancyclic.

Lemma 6.5.11 *A strong round local tournament digraph R on r vertices has cycles of length $k, k+1, ..., r$, where $k = g(R)$.*

Proof: By Lemma 6.5.10 we may assume that R contains a cycle of the form $v_{i_1}v_{i_2}...v_{i_k}v_{i_1}$, where $0 = i_1 < i_2 < ... < i_k < r$. Because D is strong, v_{i_m} dominates all the vertices $v_{i_m+1}, \ldots, v_{i_{m+1}}$ for $m = 1, 2, ..., k$. Now it is easy to see that D has cycles of lengths $k, k+1, ..., r$ through the vertices $v_{i_1}, v_{i_2}, ..., v_{i_k}$. □

There is also a very nice structure on cycles through a given vertex in a round local tournament digraph. We leave the proof as Exercise 6.26.

Lemma 6.5.12 *If a strong round locally tournament digraph with r vertices has a cycle of length k through a vertex v, then it has cycles of all lengths $k, k+1, \ldots, r$ through v.* □

Lemma 6.5.13 *[55] Let D be a strongly connected round decomposable locally semicomplete digraph with round decomposition $D = R[S_1, \ldots, S_p]$. Let $V(R) = \{r_1, r_2, \ldots, r_p\}$, where r_i is the vertex of R corresponding to S_i. Then*

(1) D is pancyclic if and only if either the girth of R is 3 or $g(R) \leq \max_{1 \leq i \leq p} |V(S_i)| + 1$.
(2) D is vertex-pancyclic if and only if, for each $i = 1, ..., p$, either $g_{r_i}(R) = 3$ or $g_{r_i}(R) \leq |V(S_i)| + 1$.

Proof: As each S_i is semicomplete, it has a hamiltonian path P_i. Furthermore, since R is a strong locally semicomplete digraph, it is hamiltonian by Theorem 5.5.1. Thus, starting from a p-cycle with one vertex from each S_i, we can get cycles of all lengths $p+1, p+2, \ldots, n$, by taking appropriate pieces of hamiltonian paths $P_1, P_2, \ldots, P_p$ in $S_1, \ldots, S_p$. Thus, if $g(R) = 3$ then D is pancyclic by Lemma 6.5.11. If $g(R) \leq \max_{1 \leq i \leq r} |V(S_i)| + 1$, then D is pancyclic by Lemma 6.5.11 and the fact that (by Moon's theorem) every S_i has cycles of lengths $3, 4, \ldots, |V(S_i)|$. If $g(R) > 3$ and, for every $i = 1, ..., r$, $g(R) > |V(S_i)| + 1$, then D is not pancyclic since it has no $(g(R)-1)$-cycle. The second part of the lemma can be proved analogously by first proving that for each $i = 1, 2, \ldots, p$, every vertex in S_i is on cycles of all lengths $g_{r_i}(R), g_{r_i}(R) + 1, \ldots, n$ (using Lemma 6.5.12) and then applying Theorem 1.5.1. □

The main part of the characterization of (vertex-)pancyclic locally semicomplete digraphs is to prove the following lemma (recall Theorem 4.11.15).

Lemma 6.5.14 *[55] Let D be a strong locally semicomplete digraph on n vertices which is not round decomposable. Then D is vertex-pancyclic.*

Proof: If D is semicomplete, then the claim follows from Moon's theorem. So we assume that D is not semicomplete. Thus, D has the structure described in Lemma 4.11.14.

Let S be a minimal separating set of D such that $D - S$ is not semicomplete and let $D_1, D_2, ..., D_p$ be the acyclic ordering of the strong components of $D - S$. Since the subdigraph $D\langle S\rangle$ is semicomplete, it has a unique acyclic ordering $D_{p+1}, ..., D_{p+q}$ with $q \geq 1$ of its strong components. Recalling Lemma 4.11.14 (a), the semicomplete decomposition of $D - S$ contains exactly three components D'_1, D'_2, D'_3. Recall that the index of the initial component of D'_2 is λ_2. From Theorem 4.11.8 and Lemma 4.11.12, we see that $D'_2 \Rightarrow D'_1 \Rightarrow S \Rightarrow D_1$ and there is no arc between D'_1 and D'_3.

We first consider the spanning subdigraph D^* of D which is obtained by deleting all the arcs between S and D'_2. By Lemma 4.11.14, D^* is a round decomposable locally semicompletedigraph and $D^* = R^*[D_1, D_2, \ldots, D_{p+q}]$, where R^* is the round locally semicomplete digraph obtained from D^* by contracting each D_i to one vertex (or, equivalently, R^* is the digraph obtained by keeping an arbitrary vertex from each D_i and deleting the rest). It can be checked easily that $g_v(R^*) \leq 5$ for every $v \in V(R^*)$. Thus D^* is vertex 5-pancyclic by the remark in the proof of Lemma 6.5.13 (in the case when $n = 4$, D is easily seen to be vertex-pancyclic so we may assume $n \geq 5$). Thus, it remains to show that every vertex of D lies on a 3-cycle and a 4-cycle.

We define

$$t = \max\{\ i\ |N^+(S) \cap V(D_i) \neq \emptyset, \lambda_2 \leq i < p\},$$

$$A = V(D_{\lambda_2}) \cup ... \cup V(D_t),$$

$$t' = \min\{\ j\ |N^+(D_j) \cap V(D'_2) \neq \emptyset, p+1 \leq j \leq p+q\}$$

$$\text{and } B = V(D_{t'}) \cup ... \cup V(D_{p+q}).$$

It follows from Proposition 4.11.16 that $B {\mapsto} D'_3 {\mapsto} A$.

Since we have $S {\mapsto} D_1 {\mapsto} D_{\lambda_2} {\mapsto} D'_1 {\mapsto} S$, every vertex of S is in a 4-cycle and since we have $B {\mapsto} D'_3 {\mapsto} A {\mapsto} D'_1 {\mapsto} S$, each vertex of $V(D'_3) \cup A \cup V(D'_1)$ is contained in a 4-cycle.

By the definition of t' and A, there is an arc sa from $D_{t'}$ to A. It follows from Lemma 4.11.14 (b) that there is an arc $a's'$ from A to B. Let $v \in V(D'_1)$ and $w \in V(D'_3)$ be arbitrarily chosen. Then $savs$ and $s'wa's'$ are 3-cycles.

Suppose D'_2 contains a vertex x that is not in A, then $A {\mapsto} x$. We also have $x, s' \in N^+(a')$ and this implies that $x {\to} s'$. From this we get that $x {\mapsto} D_{t'}$, in particular, $x {\to} s$. Hence $xsax$ is a 3-cycle and $xvsax$ is a 4-cycle. Thus, there only remains to show that every vertex of $S \cup A$ is contained in a 3-cycle.

Let u be a vertex of S and let D_ℓ be the strong component containing u. If D_ℓ has at least three vertices, then u lies on a 3-cycle by Theorem 1.5.1.

So we assume $|V(D_\ell)| \leq 2$. If $\ell < t'$, then u and a' are adjacent because D_ℓ dominates the vertex s' of B. If $\ell \geq t'$, then either $u = s$ or $s \rightarrow u$ (if $V(D_\ell) = \{s, u\}$, then usu is a 2-cycle) and hence u, a are adjacent. Therefore, in any case, u is adjacent to one of $\{a, a'\}$. Assume without loss of generality that a and u are adjacent. If $u \rightarrow a$, then $uavu$ is a 3-cycle. If $a \rightarrow u$, then $uwau$ is a 3-cycle because of $D'_3 \rightarrow A$. Hence, every vertex of S has the desired property.

Finally, we note that $S' = N^+(D'_3)$ is a subset of $V(D'_2)$ and it is also a minimal separating set of D. Furthermore, $D - S'$ is not semicomplete. From the proof above, every vertex of S' is also in a 3-cycle. So the proof of the theorem is completed by the fact that $A \subseteq S'$. $\square$

Combining Lemmas 6.5.13 and 6.5.14 we have the following characterization of pancyclic and vertex-pancyclic locally semicomplete digraphs due to Bang-Jensen, Guo, Gutin and Volkmann:

Theorem 6.5.15 *[55] A strong locally semicomplete digraph D is pancyclic if and only if it is not of the form $D = R[S_1, \ldots, S_p]$, where R is a round local tournament digraph on p vertices with $g(R) > \max\{2, |V(S_1)|, ..., |V(S_p)|\} + 1$. D is vertex-pancyclic if and only if D is not of the form $D = R[S_1, \ldots, S_p]$, where R is a round local tournament digraph with $g_{r_i}(R) > \max\{2, |V(S_i)|\} + 1$ for some $i \in \{1, ..., p\}$, where r_i is the vertex of R corresponding to S_i.* $\square$

6.5.4 Further Pancyclicity Results

To characterize pancyclic locally in-semicomplete digraphs seems a much harder problem than that of characterizing pancyclic locally semicomplete digraphs. Tewes [692] studied this problem and obtained several partial results of which we will state a few below.

Theorem 6.5.16 *[692, Theorem 4.4] Let D be a locally in-tournament digraph on n vertices and let $3 \leq k \leq n$ be an integer such that $\delta^-(D) > \frac{3n}{2(k+1)} - \frac{1}{2}$. Furthermore, let D be strong if $k \geq 2\delta^-(D) + 2$. Then D has a cycle of length k. For $k \geq \sqrt{n+1}$ this bound is sharp.* $\square$

Let the function $f(k)$ be defined as follows for fixed n:

$$f(k) = \begin{cases} \frac{n+1}{k} + \frac{k-1}{2} & \text{if } k \text{ is even} \\ \frac{n+2}{k} + \frac{k-5}{2} & \text{if } k \text{ is odd.} \end{cases}$$

Theorem 6.5.17 *[692, Theorem 4.13] Let D be a strongly connected locally in-tournament digraph on n vertices such that $\delta^-(D) > f(k)$ for some integer $3 \leq k \leq \sqrt{n+1}$. Then D has cycles of all lengths $k, k+1, \ldots, n$.* $\square$

Since every regular tournament is strong (Exercise 6.23) it is also pancyclic by Moon's theorem. Note that by Theorem 5.7.23, every regular multipartite tournament is hamiltonian. This motivated Volkmann to make the following conjecture.

Conjecture 6.5.18 *[728] Every regular p-partite tournament with $p \geq 4$ is pancyclic.*

Note that in the 3-partite tournament $D = \vec{C}_3[\overline{K}_k, \overline{K}_k, \overline{K}_k]$ all cycles have length some multiple of 3. Hence the condition $p \geq 4$ above is necessary.

For $p \geq 5$ Conjecture 6.5.18 follows from the following stronger result due to Yeo [747] (For an outline of Yeo's proof see [728]).

Theorem 6.5.19 *[747] Every regular multipartite tournament with at least 5 partite sets is vertex-pancyclic.* □

Using a probabilistic approach, Yeo [749] also proved that all, except possibly a finite number of exceptions, regular 4-partite tournaments are vertex-pancyclic (in particular, every regular 4-partite tournament on at least 488 vertices is vertex-pancyclic). The infinite family of regular and non-pancyclic 3-partite tournaments described above shows that no such result holds for 3-partite tournaments.

Clearly, the results above give strong support for the following conjecture by Yeo:

Conjecture 6.5.20 *[749] If a 4-partite tournament is regular, then it is vertex-pancyclic.*

We conjecture that the only non-vertex-pancyclic regular 3-partite tournaments are the triangular ones:

Conjecture 6.5.21 *Every 3-regular semicomplete multipartite digraph D which is not of the form $D = \vec{C}_3[\overline{K}_k, \overline{K}_k, \overline{K}_k]$ for any k is vertex-pancyclic.*

There are also many results on sufficient conditions in terms of the number of arcs for a digraph to contain a cycle of length precisely k. We refer the reader to the survey of Bermond and Thomassen [115] for a number of references to such results.

Recall that for a given directed pseudograph $D = (V, A)$, the line digraph $L(D)$ of D has vertex set A and $a \rightarrow a'$ is an arc in $L(D)$ precisely when the head of a equals the tail of a' in D (note that a loop in D gives rise to a loop in $L(D)$). Let $D = (V, A)$ be a directed pseudograph; D is **pancircular** if it contains a closed trail of length q for every $q \in \{3, 4, ..., |A|\}$. Due to a natural bijection between the set of closed trails in D and the set of cycles in $L(D)$, we obtain the following:

Proposition 6.5.22 *$L(D)$ is pancyclic if and only if D is pancircular.* □

Imori, Matsumoto and Yamada [445], who introduced the notion of pancircularity, proved the following theorem.

Theorem 6.5.23 *Let D be a regular and pancircular directed pseudograph. Then, $L(D)$ is also regular and pancircular.* $\square$

This theorem was used in [445] to show that de Bruijn digraphs are pancyclic and pancircular.

Theorem 6.5.24 *[445] Every de Bruijn digraph $D_B(d,t)$ is pancyclic and pancircular.*

Proof: de Bruijn digraphs $D_B(d,t)$ were introduced for $d \geq 2$ and $t \geq 1$. Let $D_B(d,0)$ be the directed pseudograph consisting of a singular vertex and d loops. Clearly, $D_B(d,1) = L(D_B(d,0))$. Since

$$D_B(d,t+1) = L(D_B(d,t)) \tag{6.3}$$

for $t \geq 1$ by Proposition 4.6.1, we conclude that (6.3) holds for all $t \geq 0$. We prove the theorem by induction on $t \geq 0$. Clearly, $D_B(d,0)$ is pancyclic and pancircular. Assume that $D_B(d,t)$ is pancyclic and pancircular. By Theorem 6.5.23, $L(D_B(d,t))$ is pancircular. By Proposition 6.5.22, $L(D_B(d,t))$ is pancyclic. By (6.3), $D_B(d,t+1) = L(D_B(d,t))$. Thus, $D_B(d,t+1)$ is pancyclic and pancircular. $\square$

6.5.5 Cycle Extendability in Digraphs

The following definitions are due to Hendry [420]. A non-hamiltonian cycle C in a digraph D is **extendable** if there is some cycle C' with $V(C') = V(C) \cup \{y\}$ for some vertex $y \in V - V(C)$. A digraph D which has at least one cycle is **cycle extendable** if every non-hamiltonian cycle of D is extendable. Clearly a cycle extendable digraph is pancyclic if and only if it contains a 3-cycle and vertex-pancyclic if and only if every vertex is in a 3-cycle.

The following is an easy consequence of the proof of Theorem 1.5.1:

Theorem 6.5.25 *[571] A strong tournament $T = (V,A)$ is cycle extendable unless V can be partitioned into sets U, W, Z such that $W \mapsto U \mapsto Z$ and $T\langle U \rangle$ is strong.* $\square$

Hendry [420] studied cycle extendability in digraphs with many arcs and obtained the next two results.

Theorem 6.5.26 *[420] Every strong digraph on n vertices and at least $n^2 - 3n + 5$ arcs is cycle extendable.* $\square$

Hendry showed that digraphs may have very large in- and out-degree and still not be cycle extendable. This contrasts to the situation for undirected graphs. Hendry has shown in [421, Corollary 8] that, apart from certain exceptions, every graph satisfying Dirac's condition for hamiltonicity ($d(x) \geq n/2$ for every vertex [198]) is also cycle extendable (with the obvious analogous definition of cycle extendability for undirected graphs). The main result of [420] is the following.

Theorem 6.5.27 *[420] Let D be a digraph on $n \geq 7$ vertices such that $\delta^0(D) \geq \frac{2n-3}{3}$. Then D is cycle extendable unless $n = 3r$ for some r and D contains F_n as a spanning subdigraph and D is a spanning subdigraph of G_n. See Figure 6.2 for the definition of F_n, G_n.* □

$\overleftrightarrow{K}_k$ $\overleftrightarrow{K}_k$

$\overleftrightarrow{K}_k$ $\overleftrightarrow{K}_k$ $\overleftrightarrow{K}_k$ $\overleftrightarrow{K}_k$

F_{3k} G_{3k}

Figure 6.2 The digraphs F_n and G_n. All arcs indicate complete domination in the direction shown.

6.6 Arc-Pancyclicity

A digraph D of order n is **arc-k-cyclic** for some $k \in \{3, 4, \ldots, n\}$ if each arc of D is contained in a cycle of length k. A digraph $D = (V, A)$ is **arc-pancyclic** if it is arc-k-cyclic for every $k = 3, 4, \ldots, n$. Demanding that a digraph is arc-pancyclic is a very strong requirement, since in particular every arc must be in a hamiltonian cycle. Hence it is not surprising that most results on arc-pancyclic digraphs are for tournaments and generalizations of tournaments. However, Moon proved that almost all tournaments are arc-3-cyclic [571], so for tournaments this is not such a hard requirement, in particular in the light of Theorem 6.6.1 below.

Tian, Wu and Zhang characterized all tournamentsthat are arc-3-cyclic but not arc-pancyclic. See Figure 6.3 for the definition of the classes $\mathcal{D}_6, \mathcal{D}_8$.

Theorem 6.6.1 *[718] An arc-3-cyclic tournament is arc-pancyclic unless it belongs to one of the families $\mathcal{D}_6, \mathcal{D}_8$ (in which case the arc yx does not belong to a hamiltonian cycle).* □

U W

x y x y

$\mathcal{D}_6$ $\mathcal{D}_8$

Figure 6.3 The two families of non-arc-pancyclic arc-3-cyclic tournaments. Each of the sets U and W induce an arc-3-cyclic tournament. All edges that are not already oriented may be oriented arbitrarily, but all arcs between U and W have the same direction.

It is not difficult to derive the following two corollaries from this result:

Corollary 6.6.2 *[718] At most one arc of every arc-3-cyclic tournament is not in cycles of all lengths* $3, 4, \dots, n$.

Proof: Exercise 6.31. □

Corollary 6.6.3 *[741] A tournament is arc-pancyclic if and only if it is arc-3-cyclic and arc-n-cyclic.*

Proof: Exercise 6.32. □

The following result due to Alspach is also an easy corollary:

Corollary 6.6.4 *[19] Every regular tournament is arc-pancyclic.* □

Finally, observe that since each tournament in the infinite family $\mathcal{D}_6$ is 2-strong and the arc yx is not in any hamiltonian cycle we have the following result due to Thomassen:

Theorem 6.6.5 *[698] There exist infinitely many 2-strong tournaments containing an arc which is not in any hamiltonian cycle.* □

In [341, 343] Guo studied arc-pancyclic locally tournament digraphs and obtained several results which generalize those above. In particular he made the important observation that one can in fact get a more general result by studying paths from x to y for all such pairs where the arc xy is not present rather than just those for which the arc yx is present (which is the case for tournaments of course).

Theorem 6.6.6 *[343] Let D be an arc-3-cyclic local tournament and let x, y be distinct vertices such that there is no arc from x to y. Then D contains an (x, y)-path of length k for every k such that $2 \leq k \leq n-1$ unless D is isomorphic to one of the local tournaments T_8^1, T_8^2 (from Section 6.2) or D belongs to one of the families $\mathcal{D}_6$ or $\mathcal{D}_8$, possibly with the arc from y to x missing.* $\square$

The proofs of Theorems 6.6.1 and 6.6.6 are very technical and consist of a long case analysis. Hence it makes no sense to give any of these proofs here. However, we will finish the section with a proof of the following partial result which Guo used in his proof of Theorem 6.6.6.

Theorem 6.6.7 *[343] Let D be a connected, arc-3-cyclic local tournament which is not 2-strong. Then D is isomorphic to $\vec{C}_3[T_1, T_2, \{s\}]$ where T_i is an arc-3-cyclic tournament for $i = 1, 2$ and s is a vertex. Furthermore, D is arc-pancyclic.*

Proof: First observe that D is strongly connected since it is connected and arc-3-cyclic. Since D is not 2-strong, it has a separating vertex s. Let $T_1, T_2, \ldots, T_k$ denote the acyclic ordering of the strong components of $D - s$. If there is an arc xs from $V(T_1)$ to s, then no arc from x to $V(T_2)$ can be in a 3-cycle. Hence we must have $s \mapsto V(T_1)$ and similarly $V(T_k) \mapsto s$. Since D is arc-3-cyclic, each of T_1, T_k must be an arc-3-cyclic tournament.

If $k \geq 3$ then for every vertex $u \in V(T_2)$, either no arc from $V(T_1)$ to u or no arc from u to $V(T_3)$ can be in a 3-cycle, contradicting our assumption. Thus we must have $k = 2$ and we have proved that $D = \vec{C}_3[T_1, T_2, \{s\}]$.

It remains to prove that D is arc-pancyclic. Since T_1 and T_2 have hamiltonian paths, it is easy to see that each arc which does not belong to either T_1 or T_2 is on cycles of all possible lengths. So we just have to consider arcs inside T_1, T_2. If $|V(T_1)| = |V(T_2)| = 1$ there is nothing more to prove. So suppose without loss of generality that $|V(T_1)| \geq 3$. Let $u_1 u_2 \ldots u_r u_1$, $r \geq 3$, be a hamiltonian cycle of T_1. Let $u_i u_j$ be an arbitrary arc of T_1. If $T_1 - u_i$ is strong, then $T_1 - u_i$ has a hamiltonian cycle and hence T_1 has a hamiltonian path starting with the arc $u_i u_j$. Using this and a hamiltonian path in T_2 we can easily obtain cycles of all lengths $3, 4, \ldots, n$ through $u_i u_j$ in D. Suppose now that $T_1 - u_i$ is not strong. Then $T_1 - u_i$ satisfies the assumption of the theorem, so by induction it has the same structure as D and u_j must belong to the initial component of $T_1 - u_i$. Hence again we find a hamiltonian path starting with the arc $u_i u_j$ in T_1 and finish as above.

Similarly, if $|V(T_2)| \geq 3$ the same proof as above can be applied to every arc of T_2. Thus we have shown that D is arc-pancyclic. □

It is interesting to note that the problem of characterizing arc-pancyclic semicomplete digraphs is still open and seems quite difficult. A partial result was obtained by Darrah, Liu and Zhang [181].

6.7 Hamiltonian Cycles Containing or Avoiding Prescribed Arcs

We now turn our attention to hamiltonian cycles in digraphs with the extra condition that these cycles must either contain or avoid all arcs from a prescribed subset A' of the arcs. Not surprisingly, problems of this type are quite difficult even for semicomplete digraphs. If we have no restriction on the size of A', then we may easily formulate the hamiltonian cycle problem for arbitrary digraphs as an avoiding problem for semicomplete digraphs. Hence the avoiding problem without any restrictions is certainly $\mathcal{NP}$-complete. Below, we study both types of problems from a connectivity as well as from a complexity point of view. We also show that when the number of arcs to be avoided respectively, contained in a hamiltonian cycle is some constant, then, from a complexity point of view, the avoiding version is no harder than the containing version. Finally, we show that for digraphs which can be obtained from a semicomplete digraph by adding a few new vertices and some arcs, the hamiltonian cycle problem is very hard and even if we just added one new vertex, the problem is highly non-trivial.

6.7.1 Hamiltonian Cycles Containing Prescribed Arcs

We start by studying the problem of finding a hamiltonian cycle that contains certain prescribed arcs $e_1, e_2, \ldots, e_k$. This problem, which we call the ***k*-HCA problem**, is clearly very hard for general digraphs. We show below that even for semicomplete digraphs this is a difficult problem. For $k = 1$ the k-HCA problem is a special case of the (x, y)-hamiltonian path problem and it follows from the result in Section 6.4 that there is a polynomial algorithm to decide the existence of a hamiltonian cycle containing one prescribed arc in a semicomplete digraph.

Based on the evidence from Theorem 6.4.1, Bang-Jensen, Manoussakis and Thomassen raised the following conjecture. As mentioned above, when $k = 1$ the conjecture follows from Theorem 6.4.1.

Conjecture 6.7.1 *[87] For each fixed k, the k-HCA problem is polynomially solvable for semicomplete digraphs.*

When $k = 2$ the problem already seems very difficult. This is interesting, especially in view of the discussion below concerning hamiltonian cycles in digraphs obtained from semicomplete digraphs by adding a few new vertices. Bang-Jensen and Thomassen proved that when k is not fixed the k-HCA problem becomes $\mathcal{NP}$-complete even for tournaments [89]. The proof of this result in [89] contains an interesting idea which was generalized by Bang-Jensen and Gutin in [60]. Consider a digraph D containing a set W of k vertices such that $D - W$ is semicomplete. Construct a new semicomplete digraph D_W as follows. First, split every vertex $w \in W$ into two vertices w_1, w_2 such that all arcs entering w now enter w_1 and all arcs leaving w now leave w_2. Add all possible arcs from vertices of index 1 to vertices of index 2 (whenever the arcs in the opposite direction are not already present). Add all edges between vertices of the same index and orient them randomly. Finally, add all arcs of the kind w_1z and zw_2, where $w \in W$ and $z \in V(D) - W$. See Figure 6.4. It is easy to show that the following holds:

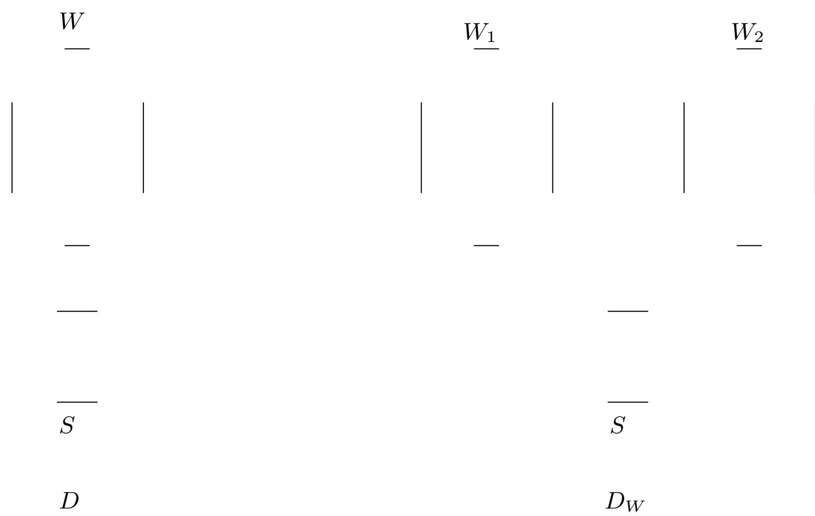


Figure 6.4 The construction of D_W from D and W. The fat arc from W_1 to W_2 indicates that all arcs not already going from W_2 to W_1 (as copies of arcs in D) go in the direction shown. The four other fat arcs indicate that all possible arcs are present in the direction shown.

Proposition 6.7.2 *[60] Let W be a set of k vertices of a digraph D such that $D - W$ is a semicomplete digraph. Then D has a cycle of length $c \geq k$ containing all vertices of W, if and only if the semicomplete digraph D_W has a cycle of length $c + k$ through the arcs $\{w_1w_2 : w \in W\}$.*

Proof: Exercise 6.36. $\square$

Let $D = (V, A)$ be a semicomplete digraph and $A' = \{u_1v_1, \ldots, u_kv_k\}$ be a subset of A. Let D' be the digraph obtained from D by replacing each arc $u_iv_i \in A'$ by a path $u_iw_iv_i$, $i = 1, 2, \ldots, k$, where w_i is a new vertex. Then every cycle C in D that uses all arcs in A' corresponds to a cycle C' in D' which contains all vertices of $W = \{w_1, w_2, \ldots, w_k\}$ and conversely. This observation and Proposition 6.7.2 allows us to study cycles through a specified set W of vertices in digraphs D such that $D - W$ is semicomplete instead of studying cycles containing $k = |W|$ fixed arcs in semicomplete digraphs.

Note that, if k is not fixed, then it is $\mathcal{NP}$-complete to decide the existence of a cycle through k given vertices in a digraph which can be obtained from a semicomplete digraph by adding k new vertices and some arcs. Indeed, take $k = |V(D)|$, then this is the Hamilton cycle problem for general digraphs. This proves that the k-HCA is $\mathcal{NP}$-complete for semicomplete digraphs.

Now we can reformulate Conjecture 6.7.1 to the following equivalent statement:

Conjecture 6.7.3 *[60] Let k be a fixed natural number. There exists a polynomial algorithm to decide if there is a hamiltonian cycle in a given digraph D which is obtained from a semicomplete digraph by adding at most k new vertices and some arcs.*

The truth of this conjecture when $k = 1$ follows from Proposition 6.7.2 and Theorem 6.4.1. Surprisingly, when $|W| = 2$ the problem already seems very difficult (recall from Section 6.4 and the remark above that even the case $|W| = 1$ is highly non-trivial).

We conclude this subsection with some results on the k-HCA problem for highly connected tournaments. Thomassen [701] obtained the following theorem for tournaments with large strong connectivity (the function $f(k)$ is defined recursively by $f(1) = 1$ and $f(k) = 2(k-1)f(k-1) + 3$ for $k \geq 2$). The proof is by induction on k and uses Theorem 6.3.3 to establish the case $k = 1$ (this is another illustration of the importance of Theorem 6.3.3).

Theorem 6.7.4 *[701] If $\{x_1, y_1, ..., x_k, y_k\}$ is a set of distinct vertices in an $h(k)$-strong tournament T, where $h(k) = f(5k)+12k+9$, then T has a k-path factor $P_1 \cup P_2 \cup ... \cup P_k$ such that P_i is an (x_i, y_i)-path for $i = 1, ..., k$.* □

Theorem 6.7.4 implies the following:

Theorem 6.7.5 *[701] If $a_1, ..., a_k$ are arcs with no common head or tail in an $h(k)$-strong tournament T, then T has a hamiltonian cycle containing $a_1, ..., a_k$ in that cyclic order.* □

Combining the ideas of avoiding and containing, Thomassen proved the following:

Theorem 6.7.6 *[701] For any set A_1 of at most k arcs in an $h(k)$-strong tournament T and for any set A_2 of at most k independent arcs of $T - A_1$, the digraph $T - A_1$ has a hamiltonian cycle containing all arcs of A_2.* □

6.7.2 Avoiding Prescribed Arcs with a Hamiltonian Cycle

How many arcs can we delete from a strong tournament and still have a hamiltonian cycle no matter what set of arcs is deleted? This is a difficult question, but it is easy to see that for some tournaments the answer is that even one missing arc may destroy all hamiltonian cycles. If some vertex has in- or out-degree 1, then deleting that arc clearly suffices to destroy all hamiltonian cycles. On the other hand, it is also easy to construct for every p an infinite set $\mathcal{S}$ of strong tournaments in which $\delta^0(T) \geq p$ for every $T \in \mathcal{S}$ and yet there is some arc of T which is on every hamiltonian cycle of T (see Exercise 6.35). It follows from Theorem 6.7.7 below that all such tournaments are strong but not 2-strong.

We can generalize the question to k-strong tournaments and again it is obvious that if some vertex v has in- or out-degree k (this is the smallest possible by the connectivity assumption), then deleting all k arcs out of or into v, we can obtain a digraph with no hamiltonian cycle. Thomassen [699] conjectured that in a k-strong tournament, k is the minimum number of arcs one can delete in order to destroy all hamiltonian cycles. The next theorem due to Fraisse and Thomassen answers this in the affirmative.

Theorem 6.7.7 *[249] For every k-strong tournament $D = (V, A)$ and every set $A' \subset A$ such that $|A| \leq k-1$, there is a hamiltonian cycle C in $D - A'$.* □

The proof is long and non-trivial; in particular it uses Theorem 6.3.3. Below we describe a stronger result due to Bang-Jensen, Gutin and Yeo [71]. The authors proved Theorem 6.7.8 using results on irreducible cycle factors in multipartite tournaments, in particular Yeo's irreducible cycle factor theorem (Theorem 5.7.21). This is just one more illustration of the power of Theorem 5.7.21.

Theorem 6.7.8 *[71] Let $T = (V, A)$ be a k-strong tournament on n vertices, and let $X_1, X_2, \ldots, X_p$ $(p \geq 1)$ be a partition of V such that $1 \leq |X_1| \leq |X_2| \leq \ldots \leq |X_p|$. Let D be the digraph obtained from T by deleting all arcs which have both head and tail in the same X_i (i.e. $D = T - \cup_{i=1}^{p} A(T\langle X_i \rangle)$). If $|X_p| \leq n/2$ and $k \geq |X_p| + \sum_{i=1}^{p-1} \lfloor |X_i|/2 \rfloor$, then D is hamiltonian. In other words, T has a hamiltonian cycle which avoids all arcs with both head and tail in some X_i.* □

We will not give the proof here since it is quite technical, but we give the main idea of the proof. The first observation is that D is a multipartite tournament, which follows from the way we constructed it. Our goal is to apply Theorem 5.7.21 to D. Hence we need to establish that D is strong (see Exercise 6.40) and has a cycle factor (Exercise 6.41). Now we can apply Theorem 5.7.21 to prove that every irreducible cycle factor in D is a hamiltonian cycle. This last step is non-trivial (Exercise 6.42).

The following result shows that the bound for k in Theorem 6.7.8 is sharp:

Theorem 6.7.9 *[71] Let $2 \leq r_1 \leq r_2 \leq \ldots \leq r_p$ be arbitrary integers. Then there exists a tournament T and a collection $X_1, X_2, ..., X_p$ of disjoint sets of vertices in T such that*

(a) T is $(r_p - 1 + \sum_{i=1}^{p-1} \lfloor r_i/2 \rfloor)$-strong;
(b) $|X_i| = r_i$ for $i = 1, 2, \ldots, p$;
(c) $D = T - \cup_{i=1}^{p} A(T\langle X_i \rangle)$ is not hamiltonian. □

In fact, the paper [71] is concerned with aspects of the following more general problem:

Problem 6.7.10 *[71] Which sets B of edges of the complete graph K_n have the property that every k-strong orientation of K_n induces a hamiltonian digraph on $K_n - B$?*

The Fraisse-Thomassen theorem says that this is the case whenever B contains at most $k-1$ edges. Theorem 6.7.8 says that a union of disjoint cliques of sizes $r_1, \ldots, r_p$ has the property whenever $\sum_{i=1}^{l} \lfloor r_i/2 \rfloor + \max_{1 \leq i \leq l}\{\lceil r_i/2 \rceil\} \leq k$. By Theorem 6.7.9, this is the best possible result for unions of cliques.

Let us show that Theorem 6.7.8 implies Theorem 6.7.7. Let T be a k-strong tournament on n vertices and let $A' = \{e_1, e_2, \ldots, e_{k-1}\}$ be a given set of $k-1$ arcs of T. In $UG(T)$ these arcs induce a number of connected components $X_1, X_2, \ldots, X_p$, $1 \leq p \leq k-1$. Denote by a_i, $i = 1, 2, \ldots, p$ the number of arcs form A' which join two vertices from X_i. Then we have $\sum_{i=1}^{p} a_i = k - 1$ and $|X_i| \leq a_i + 1$, $i = 1, 2, \ldots, p$. We may assume that the numbering is chosen so that $|X_1| \leq |X_2| \leq \ldots \leq |X_p|$. Note that $|X_p| \leq k < n/2$. Furthermore, since each $a_i \geq 1$ we also have $|X_p| \leq (k-1)-(p-1)+1 = k - p + 1$. Now we can make the following calculation:

$$\begin{aligned}
|X_p| + \sum_{i=1}^{p-1} \lfloor \frac{|X_i|}{2} \rfloor &= \lceil \frac{|X_p|}{2} \rceil + \sum_{i=1}^{p} \lfloor \frac{|X_i|}{2} \rfloor \\
&\leq \lceil \frac{|X_p|}{2} \rceil + \lfloor \frac{1}{2} \sum_{i=1}^{p} |X_i| \rfloor \\
&\leq \lceil \frac{k-p+1}{2} \rceil + \lfloor \frac{1}{2} \sum_{i=1}^{p} (a_i + 1) \rfloor \\
&= \lceil \frac{k-p+1}{2} \rceil + \lfloor \frac{k-1+p}{2} \rfloor \\
&= k.
\end{aligned}$$

Now it follows from Theorem 6.7.8 that T has a hamiltonian cycle which avoids every arc with both head and tail in some X_i and in particular it avoids all arcs in A'. This shows that Theorem 6.7.8 implies Theorem 6.7.7.

Note that if A' induces a tree and possibly some disjoint edges in $UG(T)$, then Theorem 6.7.8 is no stronger than Theorem 6.7.7. This can be seen from the fact that in this case we have equality everywhere in the calculation above. In all other cases Theorem 6.7.8 provides a stronger bound.

In relation to Problem 6.7.10, it seems natural to investigate bounds for k in different cases of the set B. In particular, what are sharp bounds for k when B is a spanning forest of K_n consisting of m disjoint paths containing $r_1, ..., r_m$ vertices, respectively ? The same question can be asked if we replace 'paths' by 'stars' or by 'cycles' (in the last case 'spanning forest' should also be replaced by 'spanning cycle subdigraph').

How easy is it to decide given a semicomplete digraph $D = (V, A)$ and a subset $A' \subseteq A$ whether D has a hamiltonian cycle C which avoids all arcs of A'? As we mentioned earlier, this problem is $\mathcal{NP}$-complete if we pose no restriction on the arcs in A'. In thecase when A' is precisely the set of those arcs that lie inside the sets of some partition $X_1, X_2, \ldots, X_r$ of V, then the existence of C can be decided in polynomial time. This follows from the fact that $D\langle A - A'\rangle$ is a semicomplete multipartite digraph and, by Theorem 5.7.9, the hamiltonian cycle problem is polynomially solvable for semicomplete multipartite digraphs. The same argument also covers the case when $k = 1$ in the conjecture below.

Conjecture 6.7.11 *For every k there exists a polynomial algorithm which, for a given semicomplete digraph $D = (V, A)$ and a subset $A' \subseteq A$ such that $|A'| = k$, decides whether D has a hamiltonian cycle that avoids all arcs in A'.*

At first glance, cycles that avoid certain arcs seem to have very little to do with cycles that contain certain specified arcs. Hence, somewhat surprisingly, if Conjecture 6.7.1 is true, then so is[6] Conjecture 6.7.11.

Suppose that Conjecture 6.7.1 is true. Then it follows from the discussion of Subsection 6.7.1 that also Conjecture 6.7.3 holds. Hence, for fixed k, there is a polynomial algorithm $\mathcal{A}_k$ which, given a digraph $D = (V, A)$ and a subset $W \subset V$ for which $D - W$ is semicomplete and $|W| \leq k$, decides whether or not D has a hamiltonian cycle. Let k be fixed and D be a semicomplete digraph and let A', $|A'| \leq k$, be a prescribed set of arcs in D. Let W be the set of all vertices such that at least one arc of A' has head or tail in W. Then $|W| \leq 2|A'|$ and D has a hamiltonian cycle avoiding all arcs in A' if and only if the digraph $D - A'$ has a hamiltonian cycle. By the remark above we can test this using the polynomial algorithm $\mathcal{A}_r$, where $r = |W|$.

6.7.3 Hamiltonian Cycles Avoiding Arcs in 2-Cycles

Recall from Chapter 4 that we call an arc xy ordinary if it is not contained in a 2-cycle. Deciding whether a given digraph has a hamiltonian cycle C such

[6] We thank Thomassen for pointing out this consequence to us (private communication, August 1999).

that all arcs of C are ordinary is of course an $\mathcal{NP}$-complete problem since the hamiltonian cycle problem for oriented graphs is $\mathcal{NP}$-complete. This implies that the problem is $\mathcal{NP}$-complete even for semicomplete digraphs.

Tuza [724] studied this problem for semicomplete digraphs and posed the following conjecture:

Conjecture 6.7.12 *[724] Let s be a positive integer and suppose that $D = (V, A)$ is a semicomplete digraph such that for every $Y \subset V$, $|Y| < s$, the induced semicomplete digraph $D\langle V - Y\rangle$ is strong and has at least one ordinary arc. Then there exists a hamiltonian cycle in T which has at least s ordinary arcs.*

The following result shows that it is enough to prove that there is a cycle of length at least $s+1$ with this property.

Proposition 6.7.13 *[724] If a strong semicomplete digraph T has a cycle of length at least $s+1$ which contains at least s ordinary arcs, then T has a hamiltonian cycle with at least s ordinary arcs.*

Tuza has proved the existence of such a cycle for $s = 1, 2$, see [724]. It is easy to see that $s+1$ cannot be replaced by s in Proposition 6.7.13 (Exercise 6.43).

6.8 Arc-Disjoint Hamiltonian Paths and Cycles

From Euler's theorem (Theorem 1.6.3) one easily derives the following result attributed to Veblen in [115] (see also Exercise 6.44).

Theorem 6.8.1 *The arcs of a digraph can be partitioned into cycles if and only if, for each vertex x, we have $d^+(x) = d^-(x)$.* □

The proof of the following strengthening of Theorem 6.8.1 for regular digraphs by Kotzig is left as Exercise 6.46.

Theorem 6.8.2 *[503] If D is a regular digraph, then the arc set of D can be partitioned into cycle factors.* □

We now consider decompositions of the arc set of a digraph into hamiltonian cycles. Deciding whether such a decomposition exists for an arbitrary digraph is an extremely hard problem. Even for complete digraphs this is non-trivial. It is an old result due to Walecki (see [20]) that the edge set of the complete undirected graph K_n has a decomposition into hamiltonian cycles if and only if n is odd (if n is even then each vertex has odd degree and no decomposition can exist). Using this result we easily conclude that the arc set of $\overleftrightarrow{K}_n$ can be decomposed into hamiltonian cycles when n is odd. However for even n another approach is needed by the remark above.

It is easy to check that the arcs of $\overleftrightarrow{K}_4$ cannot be decomposed into hamiltonian cycles. Indeed, without loss of generality, the first cycle in such a decomposition is 12341 where the vertices of $\overleftrightarrow{K}_4$ are labeled 1,2,3,4. After removing these arcs one obtains a strong semicomplete digraph with a unique hamiltonian cycle 14321 and hence the desired decomposition cannot exist. With a little more effort one can also prove that the arc set of $\overleftrightarrow{K}_6$ cannot be decomposed into 5 hamiltonian cycles (Exercise 6.45). On the other hand Tillson proved that for all other values of n such a decomposition does indeed exist.

Theorem 6.8.3 (Tillson's decomposition theorem) *[719] The arcs of $\overleftrightarrow{K}_n$ can be decomposed into hamiltonian cycles if and only if $n \neq 4, 6$.* □

Theorem 6.8.3 will be used in Section 6.12. Answering a question of Alspach, Bermond and Sotteau, Ng [591] extended Theorem 6.8.3 to the following:

Theorem 6.8.4 *[591] The arcs of $\overleftrightarrow{K}_{r,r,\ldots,r}$ (s times) can be decomposed into hamiltonian cycles if and only if $(r, s) \neq (4, 1)$ and $(r, s) \neq (6, 1)$.* □

The following conjecture, due to Kelly (see [571]), is probably one of the best known conjectures in tournament theory:

Conjecture 6.8.5 (Kelly's conjecture) *The arcs of a regular tournament of order n can be partitioned into $(n-1)/2$ hamiltonian cycles.*

This conjecture was verified for $n \leq 9$ by Alspach [115, page 28]. Jackson [449] proved that every regular tournament of order at least 5 contains a hamiltonian cycle C and a hamiltonian path arc-disjoint from C. Zhang proved in [754] that there are always two arc-disjoint hamiltonian cycles for $n \geq 5$. A digraph D is **almost regular** if $\Delta^0(D) - \delta^0(D) \leq 1$. Thomassen [699] proved the following:

Theorem 6.8.6 *[699] Every regular or almost regular tournament of order n has at least $\lfloor\sqrt{n/1000}\rfloor$ arc-disjoint hamiltonian cycles.* □

This result was improved by Häggkvist to the following:

Theorem 6.8.7 *[387] There is a positive constant c (in fact $c \geq 2^{-18}$) such that every regular tournament of order n contains at least cn arc-disjoint hamiltonian cycles.* □

Thomassen [703] proved that the arcs of every regular tournament of order n can be covered by $12n$ hamiltonian cycles.

So far the Kelly conjecture remains unsettled as far as a published proof goes. Thus it remains a serious challenge to find a proof of this long standing and very interesting conjecture.

For further results on decompositions into hamiltonian cycles we refer the reader to the paper [20] by Alspach, Bermond and Sotteau and the paper [592] by Ng.

Let T be the tournament on $n = 4m+2$ vertices obtained from two regular tournaments T_1 and T_2, each on $2m+1$ vertices, by adding all arcs from the vertices of T_1 to T_2 (i.e. $V(T_1)\mapsto V(T_2)$ in T). Clearly T is not strong and so has no hamiltonian cycle. The minimum in-degree and minimum out-degree of T is m which is about $\frac{n}{4}$. Bollobás and Häggkvist [123] showed that if we increase the minimum in- and out-degree slightly, then, not only do we obtain many arc-disjoint hamiltonian cycles, we also obtain a very structured set of such cycles.

Theorem 6.8.8 *[123] For every $\epsilon > 0$ and every natural number k there is a natural number $n(\epsilon, k)$ with the following property. If T is a tournament of order $n > n(\epsilon, k)$ such that $\delta^0(T) \geq (\frac{1}{4} + \epsilon)n$, then T contains the kth power of a hamiltonian cycle.* $\square$

It is easy to prove that every tournament on n vertices with minimum in- and out degree at least $\frac{n}{4}$ is strongly connected (see Exercise 1.36).

We now turn our attention to other results concerning arc-disjoint hamiltonian paths and cycles in tournaments. Thomassen [699] completely characterized tournaments having at least two arc-disjoint hamiltonian paths. A tournament is **almost transitive** if it is obtained from a transitive tournament with acyclic ordering $u_1, u_2, \ldots, u_n$ (i.e. $u_i \rightarrow u_j$ for all $1 \leq i < j \leq n$) by reversing the arc u_1u_n. Let T be a non-strong tournament with the acyclic ordering $T_1, T_2, \ldots, T_k$ of its strong components. Two components T_i, T_{i+1} are called **consecutive** for $i = 1, 2, \ldots, k-1$.

Theorem 6.8.9 *[699] A tournament T fails to have two arc-disjoint hamiltonian paths if and only if T has a strong component which is an almost transitive tournament of odd order or T has two consecutive strong components of order 1.* $\square$

Deciding whether a given tournament T has a hamiltonian path P and a hamiltonian cycle C such that P and C are arc-disjoint seems to be a difficult problem. Thomassen found the following partial solution involving arc-3-cyclic tournaments:

Theorem 6.8.10 *[699] Let T be an arc-3-cyclic tournament of order at least 3. Then T has a hamiltonian path P and a hamiltonian cycle arc-disjoint from P, unless T is a 3-cycle or the tournament of order 5 obtained from a 3-cycle by adding two vertices x, y and the arc xy and letting y (respectively x) dominate (respectively, be dominated by) the vertices of the 3-cycle.* $\square$

It is easy to see that regular tournaments are arc-3-cyclic (Exercise 6.47). Hence Theorem 6.8.10 generalizes the result of Jackson above. But Theorem 6.8.10 goes much further since, as we mentioned in Section 6.6, almost all tournaments satisfy the assumption of the theorem (see [571]). The following conjecture in some sense generalizing Kelly's conjecture was proposed by Thomassen:

Conjecture 6.8.11 *[699] For any $\epsilon > 0$ almost all tournaments of order n have $\lfloor (0.5 - \epsilon)n \rfloor$ arc-disjoint hamiltonian cycles.*

Erdős (see [699]) raised the following problem:

Problem 6.8.12 *Do almost all tournaments have $\delta^0(T)$ arc-disjoint hamiltonian cycles?*

As we mentioned in the beginning of Section 6.7 there is no degree condition which guarantees that a strong tournament contains two arc-disjoint hamiltonian cycles. In fact one can easily show that even high arc-strong connectivity does not exclude the existence of one arc which is in all hamiltonian cycles (see Exercise 6.35). Thomassen posed the following conjecture.

Conjecture 6.8.13 *[699] For each integer $k \geq 2$ there exists an integer $\alpha(k)$ such that every $\alpha(k)$-strong tournament has k arc-disjoint hamiltonian cycles.*

Thomassen [699] showed by an example that $\alpha(2) > 2$ and conjectured that $\alpha(2) = 3$. His example also shows that α is not bounded by any linear function.

6.9 Oriented Hamiltonian Paths and Cycles

Since every tournament has a hamiltonian directed path, it is natural to ask whether every tournament contains every orientation of a hamiltonian undirected path. This is not true, as one can see from the examples in Figure 6.5.

Figure 6.5 The unique tournaments with no anti-directed hamiltonian path.

A path is **anti-directed** if the orientation of each arc on the path is opposite to that of its predecessor. The reader can easily verify that none of the three tournaments in Figure 6.5 contains an anti-directed hamiltonian path (Exercise 6.48). Grünbaum [340] proved that, except for the three tournaments of Figure 6.5, every tournament contains an anti-directed hamiltonian path. Rosenfeld [644] strengthened this to the following statement:

Theorem 6.9.1 *[644] In a tournament on at least 9 vertices, every vertex is the origin of an anti-directed hamiltonian path.* □

Rosenfeld conjectured that there exists a natural number N such that every tournament on at least N vertices contains every orientation of a hamiltonian undirected path. Grünbaum's examples show that we must have $N \geq 8$. Rosenfeld's conjecture has been studied extensively and many partial results were obtained until it was proved by Thomason [694] (see also Theorem 6.9.3). We will mention one of these partial results here (see also the papers [21] by Alspach and Rosenfeld and [683] by Straight).

Forcade found the following beautiful result which generalizes Redei's theorem for tournaments whose number of vertices is a power of two.

Theorem 6.9.2 *[244] If T is a tournament on $n = 2^r$ vertices for some r, then for every orientation P of a path on n vertices, T contains an odd number of occurrences of P.* □

Thomason [694] proved Rosenfeld's conjecture by showing that N is less than 2^{128}. He also conjectured that $N = 8$ should be the right number. This was confirmed very recently by Havet and Thomassé [408].

Theorem 6.9.3 (Havet-Thomassé theorem) *[408] Every tournament on at least 8 vertices contains every orientation of a hamiltonian path.* □

The proof of Theorem 6.9.3 in [408] is very long (involving a lot of cases), but it uses a very nice partial result which we shall describe below. First we need some new notation. Let $P = u_1u_2 \ldots u_n$ be an oriented path. The vertex u_1 (u_n) is the **origin** (**terminus**) of P. An **interval** of P is a maximal subpath $P' = P[u_i, u_j]$[7] such that P' is a directed path (i.e. either a (u_i, u_j)-path or a (u_j, u_i)-path). See an illustration in Figure 6.6. The intervals are labeled $I_1, I_2, \ldots, I_{t(P)}$ starting from u_1. The **length** $\ell_i(P)$ of the ith interval is the number of arcs in the directed subpath corresponding to I_i. If the first interval of P is directed out of u_1, then P is an **out-path**, otherwise P is an **in-path**. Now we can describe any oriented path P by a signed sequence $sgn(P)(\ell_1, \ell_2, \ldots, \ell_{t(P)})$, where $sgn(P)$ is '+' is P is an out-path and otherwise $sgn(P)$ is '$-$'. We also use the notation $*P$ to denote the subpath $P[u_2, u_n]$.

[7] We use the same notation here as for directed paths, i.e. $P[u_i, u_j] = u_iu_{i+1} \ldots u_j$ when $i \leq j$.

1 2 3 4 5 6 7 8 9 10 11 12

Figure 6.6 An oriented path with intervals $[1,3], [3,6], [6,7], [7,8], [8,10], [10,11], [11,12]$.

For every set $X \subseteq V$ in a tournament $T = (V, A)$, we define the sets $R^+(X)$ ($R^-(X)$) to be those vertices that can be reached from (can reach) the set X by a directed path. By definition $X \subseteq R^+(X) \cap R^-(X)$. A vertex u is an **out-generator** (**in-generator**) of T if $R^+(u) = V$ ($R^-(u) = V$). Recall that by Theorem 1.4.5, every tournament T has at least one out-generator and at least one in-generator. In fact, by Proposition 4.10.2, a vertex is an out-generator (in-generator) if and only if it is the initial (terminal) vertex of at least one hamiltonian path in T.

The next result, due to Havet and Thomassé, deals with oriented paths covering all but one vertex in a tournament. It plays an important role in the proof of Theorem 6.9.3 in [408].

Theorem 6.9.4 *[408] Let $T = (V, A)$ be a tournament on $n+1$ vertices. Then*

(1) For every out-path P on n vertices and every choice of distinct vertices x, y such that $|R^+(\{x,y\})| \geq \ell_1(P) + 1$, either x or y is an origin of (a copy of) P in T.
(2) For every in-path P on n vertices and every choice of distinct vertices x, y such that $|R^-(\{x,y\})| \geq \ell_1(P) + 1$, either x or y is an origin of (a copy of) P in T.

The following is an easy corollary of Theorem 6.9.4. We state it now since we shall use it in the inductive proof below.

Corollary 6.9.5 *[694] Every tournament T on n vertices contains every oriented path P on $n-1$ vertices. Moreover, every subset of $\ell_1(P)+1$ vertices contains an origin of P. In particular, there are at least two distinct origins of P in T.* $\square$

Proof of Theorem 6.9.4: (We follow the proof in [408]). The proof is by induction on n and clearly holds for $n = 1$. Now suppose that the theorem holds for all tournaments on at most n vertices. It suffices to prove (1) since (2) can be reduced to (1) by considering the converses of T and P.

Let $P = u_1 u_2 \dots u_n$ be given and let x, y be distinct vertices such that $|R^+(\{x,y\})| \geq \ell_1(P) + 1$. We may assume that $x \rightarrow y$ and hence $R^+(x) = R^+(\{x,y\})$. We consider two cases.

Case 1 $\boldsymbol{\ell_1(P) \geq 2}$: If $|N^+(x)| \geq 2$, let $z \in N^+(x)$ be an out-generator of $T\langle R^+(x) - x\rangle$ and let $t \in N^+(x)$ be distinct from z. By the definition of z we have that $|R^+_{T-x}(\{t,z\})| = |R^+(x)| - 1 > \ell_1(*P)$. Note that $*P$ is an out-path, since $\ell_1(P) > 1$. By the induction hypothesis, either z or t is the origin of $*P$ in $T-x$, implying that x is an origin of P in T.

Thus we may assume that $N^+(x) = \{y\}$. Since $|R^+(\{x,y\})| \geq \ell_1(P)+1 \geq 3$ we see that $N^+(y) \neq \emptyset$. Let q be an out-generator of $T\langle N^+(y)\rangle$. Then q is also an out-generator of $T\langle R^+(\{x,y\}) - y\rangle$, $q{\rightarrow}x$ and $|R^+_{T-y}(\{x,q\})| = |R^+(\{x,y\})| - 1 > \ell_1(*P)$. By induction, either x or q is the origin of $*P$ in $T-y$ and since x has no out-neighbour in $T-y$ it must be q that is the origin. Now we see that y is the origin of P in T.

Case 2 $\boldsymbol{\ell_1(P) = 1}$: We consider first the subcase when $|N^+(x)| \geq 2$. Let $X := R^-_{T-x}(N^+(x))$ and consider the partition $(X, Y, \{x\})$ of V, where $Y = V - X - x$. By the definition of these sets we have $Y{\mapsto}x$, $X{\mapsto}Y$ and $y \in X$. If $|X| \geq \ell_2(P)+1$, then we claim that x is an origin of P in T; indeed, let $p \in N^+(x)$ be an in-generator of $T\langle X\rangle$ and take $u \in N^+(x) - p$. By the induction hypothesis, either p or u is an origin of $*P$ in $T-x$ and hence x is an origin of P in T.

So we may assume that $|X| \leq \ell_2(P)$. Note that $\ell_2(P) \leq n-2$ holds always (remember we count arcs). Hence $|Y| > 1$, since T has $n+1$ vertices. Let s be an in-generator of $T\langle Y\rangle$. Since $d^+(x) > 1$ and $X{\mapsto}Y$ we have $R^-_{T-y}(s) = V - y$. Let $w \in Y - s$ be arbitrary. By the induction hypothesis either w or s is an origin of $*P$ in $T-y$ and hence y is an origin of P in T.

Now consider the case when $N^+(x) = \{y\}$. Suppose first that $|N^-_{T-x}(y)| \geq n-2$. By induction, Theorem 6.9.4 and hence Corollary 6.9.5 holds for $T - \{x,y\}$. Thus some vertex in $N^-_T(y)$ is an origin of $**P$. Hence x is an origin of P in T (using $x{\rightarrow}y$ and an arc into y from the origin of $**P$ in $T-\{x,y\}$). So we may assume that $|N^+(y)| \geq 2$. Let $U = R^-_{T-y}(N^+(y))$ and $W = V - U - \{x,y\}$. Then $W{\mapsto}\{x,y\}$ and $U{\mapsto}W \cup \{x\}$. If $|U| \geq \ell_2(P)+1$, then by the same proof as we used above (beginning of Case 2), we get that y is an origin of P. So suppose $|U| \leq \ell_2(P)$. This implies in particular that $\ell_2(P) \geq |N^+(y)| \geq 2$.

If $|W| \geq 2$ then we let $w \in W$ be an in-generator of $T - \{x,y\}$ and take $w' \in W - w$ arbitrary. By induction either w or w' is an origin of the in-path $**P$ (recall that $\ell_2(P) \geq 2$ and hence $**P$ is an in-path). Thus using the arc xy and an arc into y from the origin of $**P$ in W we see that x is the origin of P. Finally consider the case when $|W| = 1$ (note that $|W| = n-1-|U| \geq 1$, since $|U| \leq \ell_2(P) \leq n-2$). Then $|U| = n-2$ and $\ell_2(P) = n-2$ (since we assumed above that $\ell_2(P) \geq |U|$). Thus $*P$ is a directed in-path. Using that y is an in-generator of $T-x$, we get that x is an origin of P. This completes the proof of the theorem. $\square$

If the path in Theorem 6.9.4 has $n+1$ vertices instead of n, then the statement is no longer true. However, the exceptions (to the $n+1, n+1$ version of Theorem 6.9.4) can be characterized [408] and based on this characterization

Havet and Thomassé were able to prove that the tournaments in Figure 6.5 are indeed the only tournaments that do not contain every orientation of a hamiltonian path.

In [408] Havet and Thomassé also proved the following nice result which is of independent interest.

Proposition 6.9.6 *[408] Let P be an out-path on n_1 vertices and Q an in-path on n_2 vertices. Let $T = (V, A)$ be a tournament on $n = n_1 + n_2$ vertices. If $x \in V$ is the origin of a copy of P and of Q in T, then we may choose copies of P and Q such that $V(P) \cap V(Q) = \{x\}$ and x is the origin of both copies.*

How easy is it to find an occurrence of a prescribed orientation of a hamiltonian path P in a tournament? If P is a directed path, then this can be done in time[8] $O(n \log n)$ (see Section 1.9.1). Some patterns can be found faster; Bampis, Hell, Manoussakis and Rosenfeld [42] showed that one can find an anti-directed hamiltonian path in $O(n)$ time. This is the best possible as shown in [415]. The following somewhat surprising result by Hell and Rosenfeld shows that finding distinct patterns requires quite different complexities:

Theorem 6.9.7 *[415] For every $0 \leq \alpha \leq 1$ there exists an orientation P of a path on n vertices so that every algorithm which checks for an occurrence of P in a tournament T with n vertices must make $\Omega(n \log^{\alpha} n)$ references to the adjacency matrix of T in the worst case.* □

Based on Theorem 6.9.3 Havet proved the following result:

Theorem 6.9.8 *[405] There is an $O(n^2)$ algorithm that takes as input a tournament on $n \geq 8$ vertices and an oriented path P on at most n vertices and returns an occurrence of P in T.* □

It is not known whether there are orientations of paths that in the worst case need $\Omega(n^{1+\epsilon})$ references (for some $\epsilon > 0$) to the adjacency matrix to be found in a tournament. By this we mean that in some cases one needs that many steps to either find the desired path or conclude that no such path exists.

Instead of considering orientations of hamiltonian paths in tournaments, one may just as well consider orientations of hamiltonian cycles in tournaments. However, one particular cycle, namely the directed hamiltonian cycle, can only be found in strong tournaments. Rosenfeld [645] conjectured that the directed hamiltonian cycle is the only orientation of a hamiltonian cycle that can be avoided by tournaments on arbitrarily many vertices. This conjecture was settled by Thomason who proved the following:

---

[8] We remind the reader that in measuring the complexity, we only count how many times we have to ask about the orientation of a given arc.

Theorem 6.9.9 *[694] Every tournament on $n \geq 2^{128}$ vertices contains every oriented cycle of length n except possibly the directed hamiltonian cycle.*

Thomason also conjectured that the correct value of the lower bound on n is 9. One easily obtains a tournament with 8 vertices having no anti-directed hamiltonian cycle by adding a new vertex v to the tournament on 7 vertices in Figure 6.5 and joining v arbitrarily to the other 7 vertices. Hence 9 would be best possible if true.

Using the methods developed in [408] along with a number of new ideas, Havet [406] proved the following result. Recall that every strong tournament has a hamiltonian cycle.

Theorem 6.9.10 *[406] Every tournament T on $n \geq 68$ vertices contains every oriented cycle of length n, except possibly the directed hamiltonian cycle.* □

Not surprisingly, if a digraph is almost complete, then it will contain all orientations of a hamiltonian undirected path. The following result is due to Heydemann, Sotteau and Thomassen:

Theorem 6.9.11 *[427] Let D be a digraph on n vertices and at least $(n-1)(n-2)+3$ arcs and let C be an arbitrary orientation of a cycle of length n. Then D contains a copy of C, except for the case when D is not strong and C is a directed hamiltonian cycle.* □

6.10 Covering All Vertices of a Digraph by Few Cycles

Now we discuss another analogue of the hamiltonian cycle problem, namely that of covering the vertices of a digraph with few cycles. In some cases we insist that these are disjoint and that there is a prescribed number of cycles, whereas in other cases we allow the cycles to intersect, but only in a prescribed pattern.

6.10.1 Cycle Factors with a Fixed Number of Cycles

Two cycles X, Y in a digraph $D = (V, A)$ are **complementary** if $V(X) \cap V(Y) = \emptyset$ and $V(X) \cup V(Y) = V$, that is, these cycles form a 2-cycle factor in D.

Since every strong tournament has a hamiltonian cycle, a tournament T contains a 2-cycle factor if and only if T can be partitioned into two strong subtournaments. Thomassen posed the following problem which generalizes the problem of the existence of a 2-cycle factor in a tournament.

Problem 6.10.1 (Thomassen) *[629] Is it true that for all natural numbers r, s, there exists a natural number $f(r, s)$ with the following property: except for finitely many exceptions for each r, s, every $f(r, s)$-strong tournament T can be partitioned into an r-strong tournament T_1 and an s-strong tournament T_2?*

Considering the case $r = s = 1$, Reid proved the following (see also Exercise 6.52):

Theorem 6.10.2 *[629] Every 2-strong tournament on at least 8 vertices has a 2-cycle factor consisting of a 3-cycle and an $(n-3)$-cycle.*

This was extended by Song to all pairs of cycle lengths $k, n-k$, where $k = 3, 4, \ldots, n-3$ [678]. It follows from these results that $f(1,1) = 2$. It is worth noticing that the problem of determining the analogue $f'(1,1)$ of $f(1,1)$ for semicomplete digraphs is open. Since every 3-strong semicomplete digraph contains a spanning 2-strong tournament (Proposition 7.14.5), we obtain that $2 \leq f'(1,1) \leq 3$ holds for semicomplete digraphs.

There are a number of results on 2-cycle factors in bipartite tournaments. One of these is the following due to Song:

Theorem 6.10.3 *[677] Let R be a bipartite tournament with $2k+1$ vertices in each partite set ($k \geq 4$). If every vertex of R has out-degree and in-degree at least k, then for any vertex x in R, R contains a 2-cycle factor $C \cup C'$ such that C includes x and the length of C is at most 6 unless R is isomorphic to $\vec{C}_4[\overline{K}_{k+1}, \overline{K}_{k+1}, \overline{K}_k, \overline{K}_k]$.* □

For further results on 2-cycle factors in semicomplete bipartite digraphs see e.g. the paper [757] by Zhang and Wang and [756] by Zhang, Manoussakis and Song.

It seems that the problem deciding the existence of a 2-cycle factor in semicomplete p-partite digraphs with $p \geq 3$ is quite difficult and we do not know any non-trivial partial results about that. The following conjecture has been proposed by Volkmann. For a semicomplete multipartite digraph D with p partite sets $V_1, V_2, \ldots, V_p$, the independence number $\alpha(D)$ is equal to the size of a largest set among the V_i's.

Conjecture 6.10.4 *[728] Let D be a p-partite tournament with partite sets $V_1, V_2, ..., V_p$ and let $\alpha = \alpha(D)$. If D is $(\alpha+1)$-strong, then D has a 2-cycle factor, unless D is a member of a finite family of multipartite tournaments.*

In fact Conjecture 6.10.4 is just one instance of the following metaconjecture due to Volkmann (private communication, 1997). Several results which hold for k-strong tournaments should also hold for every semicomplete multipartite digraph D provided that D is $(\alpha(D) + k - 1)$-strong. One instance where this is known to be true is for the hamiltonian cycle problem (see Theorem 5.7.25).

An obvious necessary condition for a digraph D to contain a 2-cycle factor is that the girth of D is at most $n/2$. The second power $D = \vec{C}^2_{2k+1}$ of an odd cycle has girth $k+1$ and D is a 2-strong locally semicomplete digraph. This shows that Theorem 6.10.2 cannot be extended to locally semicomplete digraphs. Confirming a conjecture by Bang-Jensen [47],Guo and Volkmann proved that powers of odd cycles are the only exceptions when $n \geq 8$.

Theorem 6.10.5 *[351] Let D be a 2-strong locally semicomplete digraph on $n \geq 8$ vertices. Then D has a 2-cycle factor such that both cycles have length at least 3 if and only if D is not the second power of an odd cycle.* □

Guo and Volkmann have shown that, although Theorem 6.10.2 cannot be extended to locally semicomplete digraphs, there is still enough structure to allow 2-cycle factors with many different lengths. We refer the reader to [352] for details.

The next conjecture by Bang-Jensen, Guo and Yeo goes further than Problem 6.10.1. It may be seen as a first step towards studying partitions into subtournaments containing prescribed vertices in highly connected tournaments.

Conjecture 6.10.6 *[58] For all natural numbers r, s there exists a natural number $g(r,s)$ such that the following is true with no more than finitely many exceptions for each choice of r, s: for every tournament T which is $g(r,s)$-strong and every choice of distinct vertices $x, y \in V(T)$, there exist vertex-disjoint subtournaments T_x, T_y of T such that $V(T) = V(T_x) \cup V(T_y)$, T_x is r-strong, T_y is s-strong and $x \in V(T_x)$, $y \in V(T_y)$.*

Note that it is easy to decide in polynomial time whether a tournament T contains two disjoint cycles C_x and C_y such that $x \in V(C_x)$ and $y \in V(C_y)$. This follows from the fact that, by Moon's theorem, every strongly connected tournament is vertex-pancyclic. Hence C_x and C_y exist if and only if T contains disjoint 3-cycles, one containing x and the other y. It follows from this that every 4-strong tournament contains cycles C_x, C_y as above. Bang-Jensen, Guo and Yeo proved that this already holds for 3-strong tournaments and an infinite family of 2-strong counter examples was given [58]. Hence $g(1,1) = 3$.

The existence of a 2-cycle factor such that each cycle contains a prescribed vertex and has a prescribed length in a bipartite graph has been studied in the papers [516, 733] by Little, Teo and Wang.

We now turn to cycle factors with more than two cycles. Bollobás (see [678]) posed the following problem:

Problem 6.10.7 *Let k be a positive integer. What is the least integer $g(k)$ so that all but a finite number of $g(k)$-strong tournaments contain a k-cycle factor?*

Chen, Gould and Li [147] answered this problem by proving that $g(k) \leq 3k^2 + k$. In relation to Problem 6.10.7 Song made the following much stronger conjecture:

Conjecture 6.10.8 *[678] For any k integers $n_1, n_2, ..., n_k$ with $n_i \geq 3$ for $i = 1, 2, ..., k$ and $\sum_{i=i}^{k} n_i = n$, all but a finite number of k-strong tournaments on n vertices contain a k-cycle factor such that the k cycles have the lengths $n_1, n_2, ..., n_k$, respectively.*

If, instead of tournaments, we consider digraphs which are almost complete, then, by the following result, due to Amar and Raspaud, we may almost completely specify the lengths of the cycles in a cycle factor.

Theorem 6.10.9 *[24] Let D be a strong digraph on n vertices and at least $(n-1)(n-2)+3$ arcs. For every partition $n = n_1 + n_2 + \ldots + n_k$ such that $n_i \geq 3$, $i = 1, 2, \ldots, k$, D contains a k-cycle factor $C_1 \cup C_2 \cup \ldots \cup C_k$ such that C_i has length n_i for $i = 1, 2, \ldots, k$ except in two cases:*

$n = 6$, $n_1 = n_2 = 3$ and $\alpha(D) = 3$, or
$n = 9$, $n_1 = n_2 = n_3 = 3$ and $\alpha(D) = 4$. □

6.10.2 The Effect of $\alpha(D)$ on Spanning Configurations of Paths and Cycles

Since semicomplete digraphs have a lot of structure, it is natural to believe that some of this structure is present in digraphs with small independence number, in particular for digraphs of independence number two.

Two cycles C, C' are **consistent** if they are either disjoint or their intersection is a subpath in both cycles. Chen and Manalastras proved the following:

Theorem 6.10.10 *[146] If D is strong and $\alpha(D) \leq 2$, then D is either hamiltonian or it has a pair of consistent cycles which is spanning.* □

Bondy [125] gave a short proof of this theorem based on Lemma 5.2.2.

In Chapter 7 we introduce the concept of an ear decomposition of a strong digraph. Using this concept we see that, if D has a pair of consistent cycles C, C' which are spanning and not disjoint, then these along with all remaining arcs of D (not on C, C') form an ear decomposition with precisely two non-trivial ears. Clearly the converse also holds.

Theorem 6.10.10 immediately implies the following result, which implies Theorem 5.2.4 in the case $\alpha(D) = 2$:

Corollary 6.10.11 *[146] If D is strong and $\alpha(D) \leq 2$, then D is traceable.* □

It is tempting to ask whether one can generalize Corollary 6.10.11 to the statement that every k-strong digraph D with $\alpha(D) \leq k+1$ is traceable. However, the example in Figure 6.7 by Bondy [125] shows that such a generalization is not possible. See Conjecture 12.6.2 for a weaker conjecture which may still be true.

Figure 6.7 A 2-strong digraph D with $\alpha(D) = 3$ and no hamiltonian path. The vertical edges correspond to directed 2-cycles.

Note that, if a digraph $D = (V, A)$ has a hamiltonian path, then $\mathrm{pc}(D - X) \leq |X| + 1$ for every $X \subset V$ (see also Proposition 1.4.6). In the digraph in Figure 6.7 we have $\mathrm{pc}(D - X) = 3 = |X| + 1$ when X consists of the two left vertices. Hence, the example in Figure 6.7 also shows that the condition above is not always sufficient to guarantee a hamiltonian path in a digraph.

Gallai posed the following conjecture. For $\alpha = 2$ the conjecture follows from Theorem 6.10.10.

Conjecture 6.10.12 *[296] Every strong digraph D has a spanning collection of $\alpha(D)$ not necessarily disjoint cycles.*

The **cyclomatic** number of an (un)directed graph $D = (V, A)$ is the parameter $|A| - |V| + c(D)$, where $c(D)$ denotes the number of connected components of $UG(D)$. A digraph is **cyclic** if every vertex belongs to a cycle.

The following conjecture, which Bondy [125] attributes to Chen and Manalastras [146], generalizes Gallai's conjecture above and Theorem 5.2.4:

Conjecture 6.10.13 *[125, 146] Every strong digraph D contains a cyclic spanning subdigraph with cyclomatic number at most $\alpha(D)$.*

The example below due to Favaron (see [125]) shows that one cannot hope to find, for every strong digraph D, a strong spanning subdigraph of D with cyclomatic number at most $\alpha(D)$. Let $r \geq 2$ and take r copies $T_1, T_2, \ldots, T_r$

of the strong tournament on four vertices. Let the vertices be labelled so that the unique hamiltonian cycle in the ith copy is $u_i x_i v_i y_i u_i$, $i = 1, 2, \ldots, r$. Let D_r be the digraph obtained from the disjoint union of $T_1, T_2, \ldots, T_r$ by adding the arcs $u_i u_{i+1}$ and $v_{i+1} v_i$ for all odd i, respectively, $u_{i+1} u_i$ and $v_i v_{i+1}$ for all even i, $1 \leq i \leq r$. Then D_r is strong, $\alpha(D) = r$ and it can be shown that D_r has no strong spanning subdigraph with cyclomatic number less than $2r-1$ (Exercise 6.53). Moreover, every cyclic spanning subdigraph of D with cyclomatic number r consists of r disjoint 4-cycles.

6.11 Minimum Strong Spanning Subdigraphs

We consider the following problem, which we call the MSSS problem (MSSS stands for Minimum Spanning Strong Subdigraph): given a strongly connected digraph D, find a strongly connected spanning subdigraph D' of D such that D' has as few arcs as possible. This problem, which generalizes the hamiltonian cycle problem and hence is $\mathcal{NP}$-hard, is of practical interest and has been studied extensively in the literature, see e.g. [5, 317, 434, 478, 479, 673]. We will address this problem again in Section 7.16, where we also discuss the related problem for higher connectivities.

Since the MSSS problem is $\mathcal{NP}$-hard, it is natural to study the problem under certain extra assumptions. In order to find classes of digraphs for which we can solve the MSSS problem in polynomial time, we have to consider classes of digraphs for which we can solve the hamiltonian cycle problem in polynomial time. This follows from the fact that the hamiltonian cycle problem can be solved in polynomial time if we can solve the MSSS problem in polynomial time.

6.11.1 A Lower Bound for General Digraphs

Recall that $\mathrm{pcc}(D)$, the path-cycle covering number of D, is the smallest (positive) number of paths in a k-path-cycle factor of D. Define, for every digraph D, the number $\mathrm{pcc}^*(D)$ by

$$\mathrm{pcc}^*(D) = \begin{cases} 0 & \text{if } D \text{ has a cycle factor} \\ \mathrm{pcc}(D) & \text{otherwise.} \end{cases}$$

Proposition 6.11.1 *For every strongly connected digraph $D = (V, A)$ of order n, every spanning strong subdigraph of D has at least $n + \mathrm{pcc}^*(D)$ arcs.*

Proof: Let D be strong and let D' be a spanning strong subdigraph with $n+k$ arcs. We may assume (by deleting some arcs if necessary) that no proper subdigraph of D' is spanning and strong. It is easy to prove, by induction on k, that D' can be decomposed into a cycle $P_0 = C$ and k arc-disjoint paths or cycles $P_1, P_2, \ldots, P_k$ with the following properties (here D_i denotes the digraph with vertices $\bigcup_{j=0}^{i} V(P_j)$ and arcs[9] $\bigcup_{j=0}^{i} A(P_j)$):

1. For each $i = 1, \ldots k$, if P_i is a cycle, then it has precisely one vertex in common with $V(D_{i-1})$. Otherwise the end-vertices of P_i are distinct vertices of $V(D_{i-1})$ and no other vertex of P_i belongs to $V(D_{i-1})$.
2. $\bigcup_{j=0}^{k} A(P_j) = A(D')$.

By the minimality assumption on D', each P_i has length at least two. It follows that D has a k-path-cycle factor consisting of C and k paths P'_i, $i = 1, 2, \ldots, k$, where P'_i is the path one obtains from P_i by removing the vertices it has in common with $V(D_{i-1})$ (defined above). It follows that $\mathrm{pcc}^*(D) \leq k$. $\square$

We prove in the next subsection that the inequality of Proposition 6.11.5 is in fact an equality for extended semicompletedigraphs. It was shown in [90] that this is also the case for semicomplete bipartite digraphs. The inequality of Proposition 6.11.5 is not always an equality for general semicomplete multipartite digraphs, as such digraphs can have a cycle factor and still not be hamiltonian (see Section 5.7).

Figure 6.8 A quasi-transitive digraph D with $\mathrm{pcc}^*(D) = 0$ and no hamiltonian cycle.

Even for quasi-transitive digraphs strict inequality may hold in Proposition 6.11.5. The quasi-transitive digraph D in Figure 6.8 has a cycle factor consisting of two 3-cycles and hence $\mathrm{pcc}^*(D) = 0$, but D is not hamiltonian and it is easy to see that the minimum spanning strong subdigraph has 7 arcs.

6.11.2 The MSSS Problem for Extended Semicomplete Digraphs

The next result by Bang-Jensen and Yeo shows that the inequality in Proposition 6.11.1 is actually an equality for digraphs that are extensions of a

[9] This coincides with the definition of an ear decomposition in Section 7.2.

semicomplete digraph. The main tool in the proof below is the characterization of the longest cycle in an extended semicomplete digraph given in Theorem 5.7.8.

Theorem 6.11.2 *[90] Let $D = (V, A)$ be a strong extended semicomplete digraph and let $\tilde{D} = (V, \tilde{A})$ be a minimum strong spanning subdigraph of D. Then $|\tilde{A}| = n + \mathrm{pcc}^*(D)$.*

Proof: (Sketch) Let $D = S[H_1, H_2, \ldots, H_s]$, $s = |V(S)|$, be a strong extended semicomplete digraph, where the decomposition is such that S is semicomplete. For each $i = 1, 2, \ldots, s$ we let m_i denote the maximum number of vertices from H_i that can be covered by any cycle subdigraph of D. Let C be a longest cycle of D. By Theorem 5.7.8, C contains precisely m_i vertices from H_i for each $i = 1, 2, \ldots, s$. If D is hamiltonian, then $\mathrm{pcc}^*(D) = 0$ and there is nothing to prove. Hence we may assume below that $\mathrm{pcc}^*(D) > 0$. By Corollary 5.7.19, the extended semicomplete digraph $D' = D - C$ is acyclic. Let $k = \alpha(D')$. By Lemma 5.3.3, D' has a path-factor $P_1 \cup P_2 \cup \ldots \cup P_k$ where P_1 is a longest path in D', P_2 is a longest path in $D' - P_1$ and so on.

Start by letting $H := (V(C), A(C))$. Since P_1 is a longest path in D', its initial (terminal) vertex x (y) has no arc entering (going out) in D'. Thus, since D is strong there exist arcs ux, yv such that u, v are vertices of H. Change H by adding the vertices of P and all arcs of P along with the arcs ux, yv to H. Now consider the path P_2 in $D' - P_1$. Using that P_2 is a longest path in $D' - P_1$, we again conclude that there must exist an arc from $V(H)$ to the initial vertex of P_2 and an arc from the terminal vertex of P_2 to H. Now it is easy to see how to continue and end up with a subdigraph H which is strong, spanning and has $n + k$ arcs.

It remains to prove that this is optimal. By the remark above $\mathrm{pcc}^*(D) > 0$, so by Proposition 6.11.1 it suffices to prove that $k = \mathrm{pcc}(D)$. Let $p = \mathrm{pcc}(D)$ and let $R_1, R_2, \ldots, R_p, \mathcal{Q}$ be an arbitrary p-path-cycle factor of D where $\mathcal{Q}$ consists of one or more cycles and R_i is a path for $i = 1, 2, \ldots, p$. If some R_i contains two vertices from the same H_i, then we can replace it with a new path R'_i and a cycle C_i (Exercise 6.54). Doing this for all the paths $R_1, R_2, \ldots, R_p$ until none of these contains two independent vertices we end up with a collection of paths $R'_1, R'_2, \ldots, R'_p$, where R'_i is the result of removing zero or more cycles from $D\langle R_i\rangle$[10]. Now consider the cycle subdigraph $\mathcal{Q}'$ we obtain by taking $\mathcal{Q}$ and all the cycles we extracted above. By the definition of m_i, $\mathcal{Q}'$ contains at most m_i vertices from H_i. Thus $\alpha(D - V(\mathcal{Q}')) \geq k$ and since no R'_i contains two independent vertices, it follows that $p \geq k$ must hold. □

Corollary 6.11.3 *[90] The minimum spanning strong subdigraph of a strong extended semicomplete digraph can be found in time $O(n^{\frac{5}{2}})$.*

Proof: Exercise 6.55. □

[10] Observe that by the definition of p, no R'_i is empty.

6.11.3 The MSSS Problem for Quasi-Transitive Digraphs

We first give a lower bound for the number of arcs in any minimum spanning strong subdigraph of an arbitrary given strong quasi-transitive digraph. This bound can be calculated in polynomial time using Gutin's algorithm for finding a hamiltonian cycle in a quasi-transitive digraph (Theorem 5.9.4) as well as the algorithm of Theorem 5.9.5. We prove that this lower bound is also attainable for quasi-transitive digraphs. The proof of this uses Theorem 5.7.8.

Definition 6.11.4 *Let D be a strong quasi-transitive digraph and define* $\mathrm{pc}^*(D)$ *by* $\mathrm{pc}^*(D) = 0$ *if D is hamiltonian and* $\mathrm{pc}^*(D) = \mathrm{pc}(D)$ *otherwise.*

Lemma 6.11.5 *For every strongly connected quasi-transitive digraph D, every spanning strong subdigraph of D has at least $n + \mathrm{pc}^*(D)$ arcs.*

Proof: Exercise 6.57. □

In fact Lemma 6.11.5 holds for arbitrary digraphs. This is not in contradiction with Theorem 6.11.2 since $\mathrm{pcc}^*(D) = \mathrm{pc}^*(D)$ for every strong extended semicomplete multipartite digraph by Theorems 5.7.2 and 5.7.5. Below we characterize the optimal solution to the MSSS problem for quasi-transitive digraphs and show that the problem is polynomially solvable.

Theorem 6.11.6 *[82] Every minimum spanning strong subdigraph of a quasi-transitive digraph has precisely $n + \mathrm{pc}^*(D)$ arcs. Furthermore, we can find a minimum spanning strong subdigraph in time $O(n^4)$.*

Proof: Let $D = S[W_1, W_2, \ldots, W_s]$, $s = |S| \geq 2$, be the decomposition of a strong quasi-transitive digraph D according to Theorem 4.8.5. Using the algorithm of Theorem 5.9.4 we can check whether D is hamiltonian and find a hamiltonian cycle if one exists. If D is hamiltonian, then any hamiltonian cycle is the optimal spanning strong subdigraph. Suppose below that D is not hamiltonian. Then in particular we have $\mathrm{pc}^*(D) = \mathrm{pc}(D)$ by Definition 6.11.4.

Let $D_0 = S[H_1, H_2, \ldots, H_s]$ be the extended semicomplete digraph one obtains by deleting all arcs inside each W_i (that is, $|V(H_i)| = |V(W_i)|$ and H_i is obtained from W_i by deleting all arcs).

For each $i = 1, 2, \ldots, s$, let m_i denote the maximum number of vertices which can be covered in H_i by any cycle subdigraph of D_0. According to Theorem 5.7.8 every longest cycle C in D_0 contains exactly m_i vertices from H_i, $i = 1, 2, \ldots, s$. By Theorem 5.7.8 we can find C in time $O(n^3)$. Let

$$k = \max\{\mathrm{pc}(W_i) - m_i : i = 1, 2, \ldots, s\}. \quad (6.4)$$

Note that by Theorem 5.9.3, $k \geq 1$ since D has no hamiltonian cycle. Let $m_i^* = \max\{\mathrm{pc}(W_i), m_i\}$, $i = 1, 2, \ldots, s$ and define the extended semicomplete subdigraph D^* of D by $D^* = S[H_1^*, H_2^*, \ldots, H_s^*]$, where H_i^* is an independent set containing m_i^* vertices for $i = 1, 2 \ldots, s$. Since vertices inside an independent set of D have the same in- and out-neighbours, we may think of C as a longest cycle in D^* (i.e. C contains precisely m_i vertices from H_i^*, $i = 1, 2, \ldots, s$). By Corollary 5.7.19, $D^* - C$ is acyclic and by Lemma 5.3.3, $D^* - C$ can be covered by k paths $P_1^*, P_2^*, \ldots, P_k^*$ such that P_i^* is a longest path in $D^* - (V(P^*) \cup \ldots \cup V(P_{i-1}^*))$ for $i = 1, 2, \ldots, k$.

It follows from the proof of Theorem 6.11.2 that we can glue P_1^* onto C and then P_2^* onto the resulting graph etc., until we obtain a spanning strong subdigraph D^{**} of D^* with $|V^*| + k$ arcs.

Now we obtain a spanning strong subdigraph of the quasi-transitive digraph D as follows. Since $m_i^* \geq \mathrm{pc}(W_i)$ for $i = 1, 2, \ldots, s$, each W_i contains a collection of $t_i = m_i^*$ paths $P_{i1}, P_{i2}, \ldots, P_{it_i}$ such that these paths cover all vertices of W_i. Such a collection of paths can easily be constructed from a given collection of $\mathrm{pc}(W_i)$ paths which cover $V(W_i)$. Let $x_{i1}, x_{i2}, \ldots, x_{it_i}$ be the vertex set of H_i^*, $i = 1, 2, \ldots, s$. Replace x_{ij} in D^{**} by the path P_{ij} for each $j = 1, 2, \ldots, t_i$, $i = 1, 2, \ldots, s$. We obtain a spanning strong subdigraph D' of D. The number of arcs in D' is

$$\begin{aligned} A(D') &= \sum_{i=1}^{s} (|W_i| - m_i^*) + (|V^*| + k) \\ &= (n - |V^*|) + (|V^*| + k) \\ &= n + k \end{aligned} \tag{6.5}$$

It remains to argue that D' is the smallest possible. By Lemma 6.11.5, it suffices to prove that $\mathrm{pc}^*(D) \geq k$.

Since this part is similar to the proof of Theorem 6.11.2 we only sketch how to prove it. Let $P_1, P_2, \ldots, P_r$ be an optimal path cover of D. Path-contract all subpaths that lie inside some W_i and let $P_1', \ldots, P_r'$ denote the resulting paths. Delete all arcs that still remain inside each W_i after this contraction. That way we obtain a path cover of an extended semicomplete digraph which we may consider as a subdigraph of D_0.

As in the proof of Theorem 6.11.2 we can continue replacing paths in the current collection by a cycle or a path until every path in the current collection contains at most one vertex from H_i. Let $P_1'', P_2'' \ldots, P_r''$ be the final collection after removing all such cycles. Using an argument analogous to the last part of the proof of Theorem 6.11.2, we now conclude that $r \geq k$ implying that the subdigraph D' is optimal. □

6.11.4 The MSSS Problem for Decomposable Digraphs

In fact the proof of Theorem 6.11.6 is valid for a much larger class of digraphs as we show below. For every natural number t, let Ψ_t be the class of all digraphs for which a minimum path-factor can be found in polynomial time

$O(n^t)$. For every natural number t, let Ω_t be the class of all digraphs of the form $D = S[H_1, H_2, \ldots, H_s]$, $s = |S| \geq 2$, where S is a strong semicomplete digraph and $H_i \in \Psi_t$, $i = 1, 2, \ldots, s$. By Theorem 5.9.5 the class Ω_4 contains all quasi-transitive digraphs.

The next result is an extension of Theorem 5.9.3 to a much larger class of digraphs.

Theorem 6.11.7 *Let t be a natural number and let D be a strong digraph from the class Ω_t with decomposition $D = S[W_1, W_2, \ldots, W_s]$, where $s = |S|$, $W_i \in \Psi_t$, $i = 1, 2, \ldots, s$ and S is a strong semicomplete digraph. Let $D_0 = S[H_1, H_2, \ldots, H_s]$ be the extended semicomplete digraph obtained by deleting all arcs inside each W_i (that is, $|V(H_i)| = |V(W_i)|$). Then D is hamiltonian if and only if D_0 has a cycle subdigraph which covers at least* $\mathrm{pc}(W_i)$ *vertices of H_i, $i = 1, 2, \ldots s$.*

Proof: Exercise 6.58. □

Gutin's approach to solving the hamiltonian cycle problem for quasi-transitive digraphs can be extended to a proof of the following result.

Theorem 6.11.8 *For every natural number t, the hamiltonian cycle problem is solvable in time $O(n^{\max\{3,t\}})$ for digraphs that belong to Ω_t.*

Proof: Exercise 6.59. □

Let $D = S[H_1, H_2, \ldots, H_s]$ be a digraph in Ω_t. To find a minimum strong spanning subdigraph in D, let D' be the extended semicomplete digraph obtained from D by deleting all arcs within each H_i for $i = 1, 2, \ldots, s$. By Theorem 5.7.7, we can find a longest cycle C in D'. Let $m_i = |V(H_i) \cap V(C)|$ for $i = 1, 2, \ldots, s$ and let

$$k = \max\{\mathrm{pc}(H_i) - m_i : \ i = 1, 2, \ldots, s\}.$$

Using a proof analogous to that of Theorem 6.11.6, we can show that the minimum strong spanning subdigraph of D contains $n + k$ arcs when $k \geq 1$ and is a hamiltonian cycle when $k \leq 0$. Combining this with Theorems 6.11.7 and 6.11.8 we obtain the following result:

Theorem 6.11.9 *For every natural number t, the MSSS problem is solvable in time $O(n^{\max\{3,t\}})$ for all digraphs in Ω_t.* □

We close this section with the following conjecture by Bang-Jensen and Yeo:

Conjecture 6.11.10 *[90] There exists a polynomial algorithm for the MSSS problem in the case of semicomplete multipartite digraphs.*

6.12 Application: Domination Number of TSP Heuristics

The (asymmetric) travelling salesman problem (TSP) is formulated in Section 1.9. Here, the word asymmetric simply refers to the fact that in a 2-cycle the costs of the two arcs may be different.

A **heuristic** for an optimization problem $\mathcal{R}$ is an algorithm which given an instance R of $\mathcal{R}$ finds some solution s to R for which there is generally no guarantee on the quality of s compared to an optimal solution s^* to R. So for the TSP problem a heuristic is any algorithm which returns some permutation of the vertices of the input complete graph $\overleftrightarrow{K}_n$. For more on heuristics see Section 12.8.

An equivalent of the following notion of the domination number of an algorithm was introduced by Glover and Punnen [320]. The **domination number**, $\mathrm{domn}(\mathcal{A}, n)$, of a heuristic $\mathcal{A}$ for the TSP is the maximum integer $d = d(n)$ such that, for every instance $\mathcal{I}$ of the TSP on n cities, $\mathcal{A}$ produces a tour T which is not worse than at least d tours in $\mathcal{I}$ including T itself. Clearly, every exact TSP algorithm has domination number $(n-1)!$. Thus, the domination number of an algorithm close to $(n-1)!$ may be taken as an indication that the algorithm is of high quality.

Glover and Punnen [320] asked whether there exists an algorithm $\mathcal{A}$ whose running time is polynomial in n and which has domination number $\mathrm{domn}(\mathcal{A}, n) \geq n!/p$ for some p being a constant or even polynomial in n. They conjectured that, unless $\mathcal{P} = \mathcal{NP}$, the answer to this question is negative. In [381], Gutin and Yeo proved that the answer to the Glover-Punnen question is, in fact, positive. They showed the existence of such an algorithm for $p = n-1$. The proof of the main result in [381] (see Theorem 6.12.1) uses Tillson's Theorem 6.8.3.

Using Theorem 6.12.1, Punnen and Kabadi [615] proved that several well-known and widely used TSP construction heuristics, such as various vertex insertion algorithms and Karp's cycle patching algorithm, have domination number at least $(n-2)!$.

In this section, we prove Theorem 6.12.1 and the Punnen-Kabadi result on vertex insertion algorithms, Theorem 6.12.2.

Let $(\overleftrightarrow{K}_n, c)$ denote a complete digraph on n vertices whose arcs are weighted according to a weight function c. The total cost of all Hamilton cycles in $(\overleftrightarrow{K}_n, c)$ is denoted by $\sigma(n, c)$. Denote the sum of the costs of all arcs in $(\overleftrightarrow{K}_n, c)$ by $c(\overleftrightarrow{K}_n)$. The **average cost of a Hamilton cycle** in $(\overleftrightarrow{K}_n, c)$ is denoted by $\tau(n, c)$. As every arc of $\overleftrightarrow{K}_n$ is contained in $(n-2)!$ Hamilton cycles, $\tau(n, c) = \sigma(n, c)/(n-1)! = (n-2)!c(\overleftrightarrow{K}_n)/(n-1)!$, hence, $\tau(n, c) = c(\overleftrightarrow{K}_n)/(n-1)$. This formula can also be shown using linearity of expectation (see [14]). Recall that by a tour we mean a Hamilton cycle in

$\overleftrightarrow{K}_n$. An **automorphism** of a digraph D is a bijection $\phi:\ V(D) \to V(D)$ such that $xy \in A(D)$ if and only if $\phi(x)\phi(y) \in A(D)$.

Theorem 6.12.1 *[381] Let H be a tour in $\overleftrightarrow{K}_n$ such that $c(H) \leq \tau(n,c)$. If $n \neq 6$, then there are at least $(n-2)!$ tours in $\overleftrightarrow{K}_n$ whose cost is at least $c(H)$.*

Proof: The result is trivial for $n = 2, 3$. If $n = 4$, the result follows from the simple fact that the most expensive tour T in $\overleftrightarrow{K}_n$ has cost $c(T) \geq c(H)$.

Assume that $n \geq 5$ and $n \neq 6$. Let $D_1 = \{C_1, ..., C_{n-1}\}$ be a decomposition of the arcs of $\overleftrightarrow{K}_n$ into tours (such a decomposition exists by Theorem 6.8.3). Given a tour T in $\overleftrightarrow{K}_n$, clearly there is an automorphism of $\overleftrightarrow{K}_n$ that maps C_1 into T. Therefore, if we consider D_1 together with the decompositions $(D_1, ..., D_{(n-1)!})$ of $\overleftrightarrow{K}_n$ obtained from D_1 using all automorphisms of $\overleftrightarrow{K}_n$ which map the vertex 1 into itself, we will have every tour of $\overleftrightarrow{K}_n$ in one of the D_i's. Moreover, every tour is in exactly $n-1$ of the decompositions $D_1, D_2, \ldots, D_{(n-1)!}$ (by mapping a tour C_i into a tour C_j $(i, j \in \{1, 2, ..., n-1\})$ we fix the automorphism).

Choose the most expensive tour in each of D_i and form a set $\mathcal{E}$ from all distinct tours obtained in this manner. Clearly, $|\mathcal{E}| \geq (n-2)!$. As $\sum_{i=1}^{n-1} c(C_i) = c(\overleftrightarrow{K}_n)$, every tour T of $\mathcal{E}$ has cost $c(T) \geq \tau(n,c)$. Therefore, $c(H) \leq c(T)$ for every $T \in \mathcal{E}$. $\square$

Vertex insertion algorithms for the TSP work as follows. First, we find some ordering $v_1, ..., v_n$ of vertices of $(\overleftrightarrow{K}_n, c)$. Then, we perform $n-1$ steps. On the first step we form the cycle $v_1v_2v_1$. On step k, $2 \leq k \leq n-1$, given the k-cycle $v_{\pi(1)}v_{\pi(2)}...v_{\pi(k)}v_{\pi(1)}$ from the previous step, we find j_0, which minimizes the expression

$$c(v_{\pi(j)}v_{k+1}) + c(v_{k+1}v_{\pi(j+1)}) - c(v_{\pi(j)}v_{\pi(j+1)}),$$

$1 \leq j \leq k$, and insert v_{k+1} between $v_{\pi(j_0)}$ and $v_{\pi(j_0+1)}$ forming a $(k+1)$-cycle.

The fastest such algorithm is the random insertion algorithm, in which the initial vertex ordering is random (see the paper [319] by Glover, Gutin, Yeo and Zverovich for computational experiments with this and other heuristics for the (asymmetric) TSP).

Now we can prove the Punnen-Kabadi result:

Theorem 6.12.2 *[615] Let H_n be a tour constructed by a vertex insertion algorithm $\mathcal{A}$ for the TSP on $(\overleftrightarrow{K}_n, c)$. Then $c(H_n) \leq \tau(n,c)$.*

Proof: We prove this result by induction on n. The theorem is trivially true for $n = 2$. Let $H_{n-1} = v_{\pi(1)}v_{\pi(2)}...v_{\pi(n-1)}v_{\pi(1)}$ be the cycle constructed in Step $n-2$ of the algorithm and assume that in Step $n-1$, it was decided to insert v_n between $v_{\pi(j_0)}$ and $v_{\pi(j_0+1)}$ in order to obtain H_n. Then, we have

$$
\begin{aligned}
c(H_n) &= c(H_{n-1}) + c(v_{\pi(j_0)}v_n) + c(v_n v_{\pi(j_0+1)}) - c(v_{\pi(j_0)}v_{\pi(j_0+1)}) \\
&\le c(H_{n-1}) + \frac{\sum_{i=1}^{n-1}[c(v_{\pi(i)}v_n) + c(v_n v_{\pi(i+1)}) - c(v_{\pi(i)}v_{\pi(i+1)})]}{n-1} \\
&= c(H_{n-1}) + \frac{c(V-v_n, v_n) + c(v_n, V-v_n) - c(H_{n-1})}{n-1} \\
&\le \frac{(n-2)\tau(n-1,c) + c(V-v_n, v_n) + c(v_n, V-v_n)}{n-1} \\
&= \frac{c(\overleftrightarrow{K}_n - v_n) + c(V-v_n, v_n) + c(v_n, V-v_n)}{n-1} \\
&= \frac{c(\overleftrightarrow{K}_n)}{n-1} \\
&= \tau(n,c),
\end{aligned}
$$

where $\tau(n-1,c)$ is the average cost of a tour in $\overleftrightarrow{K}_n - v_n$. □

Theorems 6.12.1 and 6.12.2 imply the following result by Punnen and Kabadi:

Theorem 6.12.3 *[615] For every vertex insertion algorithm $\mathcal{A}$ we have* $\mathrm{domn}(\mathcal{A}, n) \ge (n-2)!$. □

6.13 Exercises

6.1. **(−)** Prove that a strong semicomplete digraph D has a hamiltonian path starting at x for every $x \in V(D)$.

6.2. Prove that, if D is a strong semicomplete digraph with distinct vertices x, y such that $D - x$ and $D - y$ are strong but $D - \{x, y\}$ is not strong, then D has an (x, y)-hamiltonian path and a (y, x)-hamiltonian path.

6.3. **(−)** Prove that, from a complexity point of view, the hamiltonian path problem, the $[x, y]$-hamiltonian path problem and the (x, y)-hamiltonian path problem are all equivalent. That is, each of them can be reduced in polynomial time to each of the two others.

6.4. Derive Corollary 6.2.2 from Theorem 6.2.1.

6.5. Prove Lemma 6.2.3.

6.6. Prove the last claim in the proof of Corollary 6.2.7.

6.7. Derive Theorem 6.2.6 from Theorem 6.2.4.

6.8. **2-regular 2-strong locally semicomplete digraphs.** Prove that for every $n \geq 5$ there exists (up to isomorphism) precisely one 2-strong and 2-regular locally semicomplete digraph, namely the second power $\vec{C}_n^2$ of an n-cycle.

6.9. Prove that, if D is the second power of an even cycle, then D contains a unique hamiltonian cycle. Next, prove that D is not weakly hamiltonian-connected.

6.10. Prove Lemma 6.2.8.

6.11. Prove that if D is the second power $\vec{P}_{2k+1}^2$ of an odd path $P = u_1u_2\dots u_{2k+1}$, then there is no pair of disjoint (u_1, u_{2k})-, (u_2, u_{2k+1})-paths in D.

6.12. Prove Theorem 6.2.11.

6.13. Suppose $D = (V, A)$ is a non-strong locally semicomplete digraph with strong decomposition D_1, D_2, D_3, D_4 such that $D - x$ is connected for every $x \in V$. Let $u_i \in V(D_i)$ be specified for each $i = 1, 2, 3, 4$. Prove that D contains disjoint (u_1, u_3)-,(u_2, u_4)-paths P, Q so that $V = V(P) \cup V(Q)$.

6.14. $(+)$ Prove the following. Let T be a 2-strong semicomplete digraph and x, y vertices of T, such that $T - x$ and $T - y$ are both 2-strong, $x \not\to y$, and neither x nor y is contained in a 2-cycle. If $T - \{x, y\}$ is not 2-strong then T has an (x, y)-hamiltonian path.Hint: consider a minimal separator of the form $\{u, x, y\}$.

6.15. $(+)$ Prove Proposition 6.3.2.

6.16. $(-)$ **Hamiltonian cycles containing a prescribed arc in semicomplete digraphs**. Use Theorem 6.3.1 to show that every 3-strong semicomplete digraph $D = (V, A)$ has a cycle containing the arc a for any prescribed arc $a \in A$.

6.17. $(++)$ Prove Theorem 6.4.5.

6.18. Prove Lemma 6.4.3.

6.19. **Longest $[x, y]$-paths in tournaments**. Find a characterization for the length of a longest $[x, y]$-path in a tournament. Hint: use Theorem 6.2.1.

6.20. **Non-pancyclic digraphs satisfying Meyniel's condition**. Prove that if $m > (n+1)/2$, then the digraph $D_{n,m}$ described after Theorem 6.5.2 satisfies Meyniel's condition for hamiltonicity but has no m-cycle.

6.21. **Pancyclic digraphs satisfying Woodall's condition for hamiltonicity**. Prove that, if D satisfies the condition in Corollary 5.6.6, then either D is pancyclic, or n is even and $D = \overleftrightarrow{K}_{\frac{n}{2},\frac{n}{2}}$. Hint: use Theorem 6.5.2.

6.22. Prove the following result due to Overbeck-Larisch [598]. If a digraph $D = (V, A)$ satisfies $d(x) + d(y) \geq 2n + 1$ for every pair of non-adjacent vertices $x, y \in V$, then D is pancyclic. Hint: use Theorem 6.5.2.

6.23. $(-)$ Prove that every regular tournament is strong.

6.24. $(+)$ Prove Lemma 6.5.8. Hint: use a similar approach as that taken in the proof of Lemma 6.5.7.

6.25. $(+)$ **Vertex-pancyclic quasi-transitive digraphs.** Prove part (b) of Theorem 6.5.9. Hint: use a similar approach as taken in the proof of (a) to reduce the problem to one for extended semicomplete digraphs and then apply Theorem 6.5.6.

6.26. Prove Lemma 6.5.12. Hint: consider a shortest cycle through v (which by the assumption has length at most k).

6.27. [420] Prove the following: let $C = v_1v_2 \dots v_kv_1$ be a non-extendable cycle in a digraph $D = (V, A)$ on n vertices where $2 \leq k \leq n-1$ and let $u \in V - V(C)$. Then
(a) for every $1 \leq i \leq k$, D contains at most one of the arcs v_iu and uv_{i+1}.
(b) $|(u, V(C))| + |(V(C), u)| \leq k$,
(c) for every $1 \leq i \leq k$, $|(v_i, V - V(C))| + |(V - V(C), v_{i+1})| \leq n - k$, and
(d) if $v_{i-1}u, uv_{i+1} \in A$, then for $1 \leq h \leq i-2$ or $i+1 \leq h \leq k$, D contains at most one of the arcs v_hv_i and v_iv_{h+1} and hence $|(v_i, V(C) - v_i)| + |(V(C) - v_i, v_i)| \leq k$.

6.28. **Cycle extendable regular tournaments.** Characterize these.

6.29. **Cycle extendable locally semicomplete digraphs.** Characterize cycle extendable locally semicomplete digraphs.

6.30. **(+) Weakly cycle extendable digraphs.** Call a digraph D **weakly cycle extendable** if every cycle C which is not a longest cycle of D is contained in some larger cycle C', i.e. $V(C) \subset V(C')$. For each of the following classes characterize weakly cycle extendable digraphs:
- Extended semicomplete digraphs.
- Path-mergeable digraphs.
- In-semicomplete digraphs.

6.31. Prove Corollary 6.6.2.

6.32. Prove Corollary 6.6.3.

6.33. **(+)** A bipartite digraph $D = (V, A)$ on an even number n of vertices is **even (vertex-)pancyclic** if it has cycles of all lengths $4, 6, 8, \dots, n$ (through every vertex $v \in V$). Prove the following theorem due to Zhang [755]:

Theorem 6.13.1 *A bipartite tournament D is even vertex-pancyclic if and only if D is hamiltonian and is not isomorphic to $\vec{C}_4[\overline{K}_{\frac{n}{4}}, \overline{K}_{\frac{n}{4}}, \overline{K}_{\frac{n}{4}}, \overline{K}_{\frac{n}{4}}]$.*

6.34. Extend Theorem 6.13.1 to semicomplete bipartite digraphs (Gutin [367]).

6.35. For every $p \geq 1$, construct an infinite family $\mathcal{S}$ of strong tournaments which satisfy that $\delta^0(T) \geq p$ for each $T \in \mathcal{S}$ and there is some arc $a \in A(T)$ which belongs to every hamiltonian cycle of T. Extend your construction to work also for arbitrary high arc-strong connectivity.

6.36. Prove Proposition 6.7.2.

6.37. **(+) Hamiltonian cycles in almost acyclic digraphs.** Prove that for every fixed k there is a polynomial algorithm to decide whether there is a hamiltonian cycle in a given digraph D, which is obtained from an acyclic digraph $H = (V, A)$ by adding a set S of k new vertices and some arcs of the form st where $s \in S$ and $t \in V \cup S$. Hint: use the fact that the k-path problem is polynomial for acyclic digraphs (see Theorem 9.2.14).

6.38. Let D be constructed as in Exercise 6.37. Show that, if k is not fixed (that is, k is part of the input), then the problem above is $\mathcal{NP}$-complete.

6.39. Let T be a tournament, let $Y_1, Y_2, \ldots, Y_s$ ($s \geq 1$) be disjoint sets of vertices in T and let x and y be arbitrary distinct vertices in $V(T) - (Y_1 \cup Y_2 \cup \ldots \cup Y_s)$. Prove that, if there exist k disjoint (x, y)-paths in T, then there exist at least $k - \sum_{i=1}^{s} \lfloor |Y_i|/2 \rfloor$ disjoint (x, y)-paths in $T - \cup_{i=1}^{s} A(T\langle Y_i \rangle)$.

6.40. (+) Let $X_1, X_2, \ldots, X_p$ and D be defined as in Theorem 6.7.8. Prove that D is strong. Hint: first prove the following two claims and then combine them into a proof that D is strong:
(a) If $x \in X_i$ and $y \in X_j$ ($1 \leq i \neq j \leq l$), then there are $\lfloor |X_i|/2 \rfloor + \lfloor |X_j|/2 \rfloor + \lceil |X_l|/2 \rceil$ disjoint (x, y)-paths in $D_{i,j}$.
(b) If $x, y \in X_i$ ($x \neq y$), then there are $|X_i|$ disjoint (x, y)-paths in D_i. Furthermore there is an (x, y)-path in D (Bang-Jensen, Gutin and Yeo [71]).

6.41. (+) Prove that the digraph D in Theorem 6.7.8 has a cycle factor [71]. Hint: let D' be obtained from D by the vertex-splitting technique (Section 3.2). Form a network from D' by putting lower bound 1 on arcs of the kind $v_t v_s$, $v \in V(D)$ and zero elsewhere. Put capacity 1 on arcs of the kind $v_t v_s$ and ∞ on all other arcs. Now apply Theorem 3.8.2 and deduce the result from the structure one can derive using a presumed bad cut $(S, \overline{S})$.

6.42. (+) Prove that the digraph D in Theorem 6.7.8 is hamiltonian [71]. Hint: consider any irreducible factor. Apply Theorem 5.7.21 and conclude that the cycle factor is a hamiltonian cycle.

6.43. Show by an example that $s+1$ cannot be replaced by s in Proposition 6.7.13.

6.44. Show that Theorem 6.8.1 follows from Theorem 1.6.3.

6.45. Prove that the arcs of $\overleftrightarrow{K}_6$ cannot be decomposed into 5 hamiltonian cycles.

6.46. (−) Prove Theorem 6.8.2. Hint: use Exercise 3.70.

6.47. (−) Prove that every regular tournament is arc-3-cyclic. Show that this is not always true for regular semicomplete digraphs.

6.48. (−) Verify that none of the three tournaments in Figure 6.5 contain an anti-directed hamiltonian path.

6.49. Prove Theorem 6.8.9.

6.50. **Orientations of paths in strong tournaments.** Prove the following statement. Let T be a strong tournament on n vertices and P an out-path on $n-1$ vertices. Then
(a) every vertex of T except possibly one is an origin of P and
(b) if $\ell_1(P) \geq 2$, then every vertex of out-degree at least 2 is an origin of P.

6.51. **Orientations of paths in 2-strong tournaments.** Let T be a 2-strong tournament on n vertices and let P be an oriented path on $n-1$ vertices. Prove that every vertex of T is an origin of P.

6.52. Show that there is only one 2-strong tournament on 7 vertices which has no 2-cycle factor.

6.53. Let D_r be the digraph which is defined in the end of Subsection 6.10.2. Show that every strong spanning subdigraph of D_r has cyclomatic number at least $2r-1$. Next show that every cyclic spanning subdigraph of D_r with cyclomatic number r is an r-cycle factor in which all cycles are 4-cycles.

6.54. Prove that if a path P in an extended semicomplete digraph D contains two vertices from an independent set I of D, then there exists a path P' and a cycle C' in D with $V(P) = V(P') \cup V(C')$.

6.55. (+) Prove Corollary 6.11.3. Hint: the proof is algorithmic. Identify the subroutines needed to do the different steps. See also Exercise 3.59.

6.56. Show that the proof of Theorem 6.11.6 can be turned into an $O(n^4)$ algorithm for finding a minimum strong spanning subdigraph of a quasi-transitive digraph.

6.57. (+) Prove Lemma 6.11.5. Hint: consider the way we argued in the proof of Proposition 6.11.1.

6.58. (+) Prove Theorem 6.11.7. Hint: use the same approach as in the proof of Theorem 5.9.1.

6.59. (+) Prove Theorem 6.11.8. Hint: use the same approach as in the proof of Theorem 5.9.4.

7. Global Connectivity

The concept of connectivity is one of the most fundamental concepts in (directed) graph theory. There are numerous practical problems which can be formulated as connectivity problems for digraphs and hence a significant part of this theory is also important from a practical point of view. Results on connectivity are often quite difficult and a deep insight may be required before one can obtain results in the area. The purpose of this chapter is to convey some of that insight by illustrating several important topics as well as techniques that have been successful in solving global connectivity problems. Several of these problems, such as the connectivity augmentation problems in Sections 7.6 and 7.7, are of significant practical interest. Because of the very large number of important results on connectivity, we will devote this chapter as well as Chapters 8 and 9 to this area. This chapter will mainly deal with global connectivity aspects. That is, the directed multigraph in question is k-(arc)-strong for some $k \geq 0$, or we want to make it k-(arc)-strong by adding new arcs.

We will often consider directed multigraphs rather than directed graphs, since several results on arc-strong connectivity hold for this larger class and also it becomes easier to prove many results. However, when we consider vertex-strong connectivity, multiple arcs play no role and then we may assume that we are considering digraphs. Note that, unless we explicitly say otherwise, we will assume that we are working with a directed graph (i.e there are no multiple arcs).

After introducing some new terminology and an efficient way of representing a directed multigraph as a network we proceed to ear-decompositions of strong directed multigraphs. We show how to use this useful concept to obtain short proofs of several basic connectivity results. Then we state and prove Menger's theorem which is one of the most fundamental results in graph theory. Based on Menger's theorem, we describe various algorithms to determine the arc-strong and vertex-strong connectivity of a directed multigraph. In Section 7.5 we introduce the operation of splitting off a pair of arcs incident with a vertex. We prove Mader's splitting theorem which allows one to give inductive proofs for several important results on directed multigraphs. Using Mader's theorem we describe a solution due to Frank for the problem of finding a minimum set of new arcs to add to a directed multigraph such that the result is a k-arc-strong multigraph. In Section 7.7 we describe a solution by Frank and Jordán of the analogous problem for vertex-strong connectivity.

Another way of increasing the arc-strong or vertex-strong connectivity of a digraph is by reversing the orientation of certain arcs. In Section 7.9 we discuss this approach and describe an interesting result for semicomplete digraphs by Bang-Jensen and Jordán. In Section 7.10 we study the structure of directed multigraphs which are k-(arc)-strong but removing any arc destroys that property. We prove deep results by Mader on the structure of such directed multigraphs. Section 7.11 deals with digraphs which are k-strong but no vertex can be deleted without decreasing the vertex-strong connectivity. In Section 7.12 we briefly discuss directed multigraphs for which the degree of arc-strong connectivity is as large as possible, that is, equal to the minimum degree. In Section 7.13 we show that decomposable digraphs have an interesting connectivity structure.

In Section 7.14 we study an interesting problem due to Jackson and Thomassen concerning the existence of highly connected orientations of digraphs with high connectivity. We show that such orientations exist in the case of locally semicomplete digraphs and quasi-transitive digraphs. In Section 7.15 we give a proof due to Lovász of the Lucchesi-Younger theorem concerning arc-disjoint dicuts in directed multigraphs. Finally, in Section 7.16 we consider the problem of finding a small spanning subdigraph of a directed multigraph D with the same degree of arc-strong, respectively vertex strong, connectivity as D.

7.1 Additional Notation and Preliminaries

Let $D = (V, A)$ be a directed multigraph and let $X, Y \subseteq V$ be subsets of V. We denote by $d^+(X, Y)$ the number of arcs with tail in $X - Y$ and head in $Y - X$, i.e $d^+(X, Y) = |(X - Y, Y - X)_D|$. Furthermore we let $d(X, Y) = d^+(X, Y) + d^+(Y, X)$. Hence we have $d^+(X) = d^+(X, V - X)$ and $d^-(X) = d^+(V - X, X)$. An arc xy **leaves** a set X if $x \in X$ and $y \in V - X$. The sets X, Y are **intersecting** if each of the sets $X - Y, X \cap Y, Y - X$ is non-empty. If also $V - (X \cup Y) \neq \emptyset$, then X and Y are **crossing**.

Let $\mathcal{F}$ be a family of subsets of a set S. We call a set $A \in \mathcal{F}$ a **member** of $\mathcal{F}$. The family $\mathcal{F}$ is an **intersecting family** (a **crossing family**) if $A, B \in \mathcal{F}$ implies $A \cup B, A \cap B \in \mathcal{F}$ whenever A, B are intersecting (crossing) members of $\mathcal{F}$. A family $\mathcal{F}$ of subsets of a set S is **laminar** if it contains no two intersecting members. That is, if $A, B \in \mathcal{F}$ and $A \cap B \neq \emptyset$ then either $A \subseteq B$ or $B \subseteq A$ holds. A family of sets is **cross-free** if it contains no two crossing members.

For an arbitrary directed multigraph $D = (V, A)$ and vertices $x, y \in V$ we denote by $\lambda(x, y)$ ($\kappa(x, y)$) the maximum number of arc-disjoint (internally disjoint) (x, y)-paths in D. The numbers $\lambda(x, y), \kappa(x, y)$ are called the **local**

arc-strong connectivity, respectively, the **local vertex-strong connectivity** from x to y. Furthermore we let

$$\begin{aligned} \lambda'(D) &= \min_{x,y \in V} \lambda(x,y) \\ \kappa'(D) &= \min_{x,y \in V} \kappa(x,y). \end{aligned} \tag{7.1}$$

Analogously to the way we defined a cut with respect to an (s,t)-flow in Chapter 3 we define an $\boldsymbol{(s,t)}$**-cut** to be a set of arcs of the form $(U, \bar{U})$, where $\bar{U} = V - U$ and $s \in U, t \in \bar{U}$. Recall that an (s,t)-separator is a subset $X \subseteq V(D) - \{s,t\}$ with the property that $D - X$ has no (s,t)-path. We also say that X **separates** s from t. Thus a separator of D is a set of vertices S such that S is an (s,t)-separator for some pair $s,t \in V(D)$ (recall the definition of a separator from Subsection 1.5). A **minimum separator** of D is a minimum cardinality separator X of D.

The following simple observation plays a central role in many proofs of connectivity results.

Proposition 7.1.1 *Let $D = (V, A)$ be a directed multigraph and let X, Y be subsets of V. Then the following holds:*

$$\begin{aligned} d^+(X) + d^+(Y) &= d^+(X \cup Y) + d^+(X \cap Y) + d(X,Y) \\ d^-(X) + d^-(Y) &= d^-(X \cup Y) + d^-(X \cap Y) + d(X,Y). \end{aligned} \tag{7.2}$$

Furthermore, if $d^-(X \cap Y) = d^+(X \cap Y)$, then we also have

$$\begin{aligned} d^+(X) + d^+(Y) &= d^+(X - Y) + d^+(Y - X) + \epsilon \\ d^-(X) + d^-(Y) &= d^-(X - Y) + d^-(Y - X) + \epsilon, \end{aligned} \tag{7.3}$$

where $\epsilon = d(X \cap Y, V - (X \cup Y))$.

Proof: Each of these equalities can easily be proved by considering the contribution of the different kinds of arcs that are counted on at least one side of the equality. For example Figure 7.1 shows the possible edges contributing to at least one side of the first equality. □

A set function f on a groundset S is **submodular** if $f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$ for all $X, Y \subseteq S$. The next corollary which follows directly from Proposition 7.1.1 is very useful, as we shall see many times in this chapter.

Corollary 7.1.2 *For an arbitrary directed multigraph D, d^+_D, d^-_D are submodular functions on $V(D)$.* □

Recall that for a proper subset X of $V(D)$ we denote by $N^+(X)$ the set of out-neighbours of X. The next result shows that the functions $|N^-|, |N^+|$ are also submodular.

X Y

Figure 7.1 The various types of arcs contributing to the out-degrees of the sets $X, Y, X \cap Y$ and $X \cup Y$.

Proposition 7.1.3 *Let $D = (V, A)$ be a digraph and let X, Y be subsets of V. Then the following holds:*

$$|N^+(X)| + |N^+(Y)| \geq |N^+(X \cap Y)| + |N^+(X \cup Y)|$$
$$|N^-(X)| + |N^-(Y)| \geq |N^-(X \cap Y)| + |N^-(X \cup Y)|.$$

Proof: These inequalities can easily be checked by considering the contributions of the different kind of neighbours of the sets $X, Y, X \cap Y$ and $X \cup Y$ (Exercise 7.1). $\square$

7.1.1 The Network Representation of a Directed Multigraph

In many proofs and algorithms concerning directed multigraphs, it is convenient to think of a directed multigraph as a (flow) network. Here we will formalize this and prove an elementary result which will be applied in later sections.

Definition 7.1.4 *Let $D = (V, A)$ be a directed multigraph. The* **network representation** *of D, denoted $\mathcal{N}(D)$, is the following network: $\mathcal{N}(D) = (V, A', \ell \equiv 0, u)$ where A' contains the arc ij precisely when D contains at least one arc from i to j. For every arc $ij \in A'$ u_{ij} is equal to the number of arcs from i to j in D. See Figure 7.2.*

The next lemma shows a useful connection between arc-disjoint paths in D and flows in $\mathcal{N}(D)$.

Lemma 7.1.5 *Let $D = (V, A)$ be a directed multigraph and let s, t be distinct vertices of V. Then $\lambda(s, t)$ equals the value of a maximum (s, t)-flow in $\mathcal{N}(D)$.*

Proof: Let $P_1, \ldots, P_r$ be a collection of pairwise arc-disjoint (s, t)-paths in D. These paths may use different copies of an arc between the same two

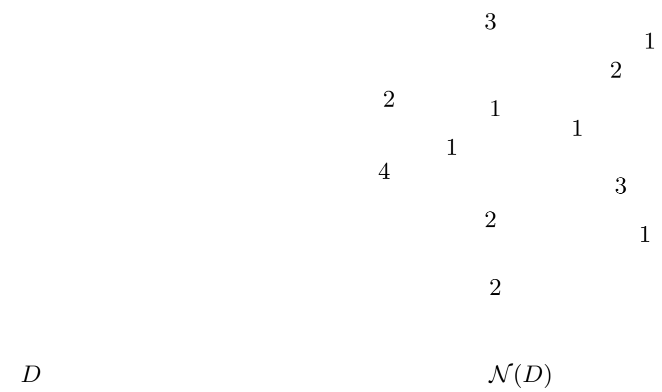


Figure 7.2 A directed multigraph D and its network representation $\mathcal{N}(D)$. Numbers on arcs indicate capacity in $\mathcal{N}(D)$.

vertices i and j, but, since the paths are arc-disjoint, in total they use no more than u_{ij} copies of the arc ij. Hence we can construct a feasible (s,t)-flow of value r in $\mathcal{N}(D)$ just by sending one unit of flow along each of the paths $P_1, \dots, P_r$. Conversely, if x is any integral (s,t)-flow of value k in $\mathcal{N}(D)$ (recall Theorem 3.5.5), then by Theorem 3.3.1, x can be decomposed into k (s,t)-path-flows $f(P_1), \dots, f(P_k)$ of value 1 (those that have a higher value $r > 1$ can be replaced by r (s,t)-path-flows of value 1 along the same path) and some cycle flows. By the capacity constraint on the arcs, at most u_{ij} of these path flows use the arc ij. Hence we can replace the arcs used by each $f(P_i)$ by arcs in D in such a way that we obtain k arc-disjoint (s,t)-paths in D. This completes the proof of the lemma. □

7.2 Ear Decompositions

In this section we study the structure of strongly connected digraphs by introducing the concept of an ear decomposition (see Figure 7.3) and derive a number of interesting results from this definition. Among other things, we reprove some of the results from Chapter 1.

Definition 7.2.1 *An* **ear decomposition** *of a directed multigraph $D = (V, A)$ is a sequence $\mathcal{E} = \{P_0, P_1, P_2, \dots, P_t\}$, where P_0 is a cycle*[1] *and each P_i is a path, or a cycle with the following properties:*

(a) P_i and P_j are arc-disjoint when $i \neq j$.

[1] Some authors take P_0 to be just a vertex, but it is easy to see that the two definitions are equivalent for strong directed multigraphs with at least one arc.

(b) *For each $i = 1, \ldots t$: If P_i is a cycle, then it has precisely one vertex in common with $V(D_{i-1})$. Otherwise the end-vertices of P_i are distinct vertices of $V(D_{i-1})$ and no other vertex of P_i belongs to $V(D_{i-1})$. Here D_i denotes the digraph with vertices $\bigcup_{j=0}^{i} V(P_j)$ and arcs $\bigcup_{j=0}^{i} A(P_j)$.*

(c) $\bigcup_{j=0}^{t} A(P_j) = A$.

Each P_i, $0 \leq i \leq t$ is called an **ear** *of $\mathcal{E}$. The* **number of ears** *in $\mathcal{E}$ is the number $t+1$. An ear P_i is* **trivial** *if $|A(P_i)| = 1$. All other ears are* **non-trivial**.

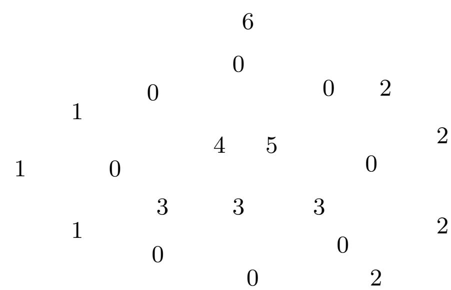


Figure 7.3 An ear decomposition $\mathcal{E} = \{P_0, P_1, \ldots, P_6\}$ of a digraph. The number on each arc indicates the number of the ear to which it belongs. The ears P_0, P_1, P_2, P_3 are non-trivial and the ears P_4, P_5, P_6 are trivial.

Theorem 7.2.2 *A directed multigraph is strong if and only if it has an ear decomposition. Furthermore, if D is strong, then for every vertex v, every cycle C containing v can be used as starting cycle P_0 for an ear decomposition of D.*

Proof: We may assume that $|V(D)| \geq 3$ since otherwise the claim is trivial. Suppose first that D has an ear decomposition $\mathcal{E} = \{P_0, P_1, P_2, \ldots, P_t\}$. Note that the digraph P_0 is strong. Now it is easy to prove, by induction on the number of ears in $\mathcal{E}$, that D is strong. If D_i is strong, then D_{i+1} is also strong since it is obtained by adding a path with two end-vertices x, y in D_i and all other vertices outside of $V(D_i)$.

Conversely, assume that D is strong and let v be an arbitrary vertex in $V(D)$. Since $|V(D)| \geq 3$ and D is strong, there is some cycle $C = u_1 u_2 \ldots u_r$, where $u_1 = u_r = v$, through v. Let $P_0 := C$, $i := 0$ and execute phase 1 and 2 below:

Phase 1:

1. If every vertex of $V(D)$ is in $V(D_i)$, then go to Phase 2.
2. Let $i := i + 1$ and let u be a vertex not in $V(D_{i-1})$ such that there is some arc xu from $V(D_{i-1})$ to u.

3. Let P_i be a shortest path from u to $V(D_{i-1})$.
4. Take xP_i as the next ear and repeat Phase 1.

Phase 2:

1. For each remaining arc vw of D which was not included in $A(D_i)$ (i is the counter above) do the following:
2. Let $i := i + 1$ and let $P_i = vw$ (that is, include all these arcs as trivial ears).

To see that the algorithm above finds an ear decomposition of D, it suffices to check that we can always find an arc xu and a path from u to $V(D_i)$ as claimed in Phase 1. This follows easily from the fact that D is strong. □

There are several interesting consequences of Theorem 7.2.2 and its proof.

Corollary 7.2.3 *Every ear-decomposition of a strong digraph on n vertices and m arcs has $m - n + 1$ ears.*

Proof: Exercise 7.2. □

Corollary 7.2.4 *Every strong directed multigraph D on n vertices contains a spanning strong subgraph with at most $2n - 2$ arcs. Furthermore, there are directed multigraphs for which every spanning strong subgraph has at least $2n - 2$ arcs.*

Proof: First observe that we can remove all trivial ears in any ear decomposition of D without destroying strong connectivity. Thus it suffices to estimate the number of arcs in the non-trivial ears. Let $\mathcal{E} = P_0, P_1, \ldots, P_r, P_{r+1}, \ldots, P_t$ be an ear decomposition of D where $P_0, P_1, \ldots, P_r$ are the non-trivial ears. Let P'_i be the path $P_i - V(D_{i-1})$. Since each P_i, $i = 1, 2, \ldots, r$ adds at least one new vertex, there can be no more than $n - |V(P_0)|$ of these. Each new ear P_i adds $|V(P'_i)| + 1$ new arcs and hence we can make the following estimate:

$$
\begin{aligned}
|A(D_r)| &= |V(P_0)| + \sum_{i=1}^{r}(|V(P'_i)| + 1) \\
&= |V(P_0)| + r + \sum_{i=1}^{r}(|V(P'_i)|) \\
&= n + r \\
&\leq n + n - |V(P_0)| \\
&\leq 2n - 2, \qquad (7.4)
\end{aligned}
$$

where equality only holds if $|V(P_0)| = 2$ and each P_i, $i = 1, 2, \ldots, r$, has length 2. To see that the estimate $2n-2$ is best possible, it suffices to consider the complete biorientation of a path on n vertices. □

Corollary 7.2.5 *There is a linear algorithm to find an ear decomposition of a strong directed multigraph D.*

Proof: This can be seen from the proof of Theorem 7.2.2. The proof itself is algorithmic and it is not too hard to see that if we use breadth first search (see Section 2.3.1) together with a suitable data structure to find the path from u to $V(D_{i-1})$, then we can obtain a linear algorithm. Details are left to the interested reader as Exercise 7.21. $\square$

Corollary 7.2.6 *It is an $\mathcal{NP}$-complete problem to decide whether a given digraph D has an ear decomposition with at most r non-trivial ears. It is $\mathcal{NP}$-complete to decide if a given digraph D has an ear decomposition with at most q arcs in the non-trivial ears.*

Proof: Note that in both cases the numbers r (respectively q) are assumed to be part of the input to the problem. A strong digraph D has an ear decomposition with only one non-trivial ear (respectively, precisely n arcs in the non-trivial ears) if and only if D has a Hamilton cycle. Hence both claims follow from Theorem 5.0.1. $\square$

The next two Corollaries were proved in Chapter 1, but we reprove them here to illustrate an application of ear-decompositions. Recall that a bridge of an undirected graph G is an edge e such that $G - e$ is not connected.

Corollary 7.2.7 *[637] A strong digraph D contains a spanning oriented subgraph which is strong if and only if $UG(D)$ has no bridge.*

Proof: If $UG(D)$ has a bridge, xy, then D contains the 2-cycle xyx, since D is strong. Observe that no matter which of these two arcs we delete we obtain a non-strong digraph. Suppose conversely that $UG(D)$ has no bridge. Consider again the proof of Theorem 7.2.2. If we can always choose the path from u to $V(D_{i-1})$ in such a way that it does not end in x, or contains at least one inner vertex, then it follows from the fact that we use shortest paths that no ear P_i, $i \geq 1$ contains a 2-cycle. In the remaining case, the only path from u to $V(D_{i-1})$ is the arc ux and hence the 2-cycle xux is a bridge in $UG(D)$. It remains to avoid using a 2-cycle as starting point (that is, as the cycle P_0). This can be done, unless all cycles in D are 2-cycles. If this is the case then $UG(D)$ is a tree and every edge of $UG(D)$ is a bridge, contradicting the assumption. $\square$

Corollary 7.2.8 *[120] A mixed graph M has a strong orientation if and only if M is strongly connected and has no bridge.*

Proof: This follows from Corollary 7.2.7, since we may associate with any mixed graph $M = (V, A, E)$ the directed graph D one obtains by replacing each edge in M by a 2-cycle. Clearly deleting an arc of a 2-cycle in D corresponds to orienting the corresponding edge in M. $\square$

Ear decompositions of undirected graphs can be similarly defined. These play an important role in many proofs on undirected graphs, in particular in Matching Theory; see e.g. the book by Lovász and Plummer [525].

7.3 Menger's Theorem

The following theorem, due to Menger [562], is one of the most fundamental results in graph theory.

Theorem 7.3.1 (Menger's theorem) *[562] Let D be a directed multigraph and let $u, v \in V(D)$ be a pair of distinct vertices. Then the following holds:*

(a) The maximum number of arc-disjoint (u, v)-paths equals the minimum number of arcs covering all (u, v)-paths and this minimum is attained for some (u, v)-cut $(X, \bar{X})$.
(b) If the arc uv is not in $A(D)$, then the maximum number of internally disjoint (u, v)-paths equals the minimum number of vertices in a (u, v)-separator.

Proof: First let us see that version (b) involving vertex disjoint paths can be easily derived from the arc-disjoint version (a). First recall that multiple arcs play no role in questions regarding (internally) vertex disjoint paths and hence we can assume that the directed multigraph in question is actually a digraph. Given a digraph $D = (V, A)$ and $u, v \in V$ construct the digraph D_{ST} by the vertex splitting procedure (see Section 3.2.4). Now it is easy to check that arc-disjoint (u_s, v_t)-paths in D_{ST} correspond to internally disjoint (u, v)-paths in D (if an (u_s, v_t)-path in D_{ST} contains the vertex x_t (x_s) for some $x \neq u, v$, then it must also contain x_s (x_t)). Furthermore, for any set of ℓ arcs that cover all (u_s, v_t)-paths in D_{ST}, there exists a set of ℓ arcs of the form $w_t^1 w_s^1, \ldots, w_t^\ell w_s^\ell$ with the same property and such a set corresponds to an (u, v)-separator $X = \{w^1, \ldots, w^\ell\}$ in D. Hence it suffices to prove (a).

Because of the similarity between Menger's theorem (in the form (a)) and the Max-flow Min-cut theorem (Theorem 3.5.3), it is not very surprising that we can prove Menger's theorem in version (a) using Theorem 3.5.3. We did part of the work already in Section 7.1.1 where we showed that $\lambda(u, v)$ equals the value of a maximum (u, v)-flow in $\mathcal{N}(D)$. Similarly it is easy to see that every (u, v)-cut $(X, \bar{X})$ in D corresponds to a (u, v)-cut $(X, \bar{X})$ in $\mathcal{N}(D)$ of capacity $|(X, \bar{X})|$ and conversely. Now (a) follows from Theorem 3.5.3. $\square$

As we shall see in Exercise 7.16, for networks where all capacities are integers, we can also derive the Max-flow Min-cut theorem from Menger's theorem.

In order to illustrate the use of submodularity in proofs concerning connectivity for digraphs we will give a second proof of Theorem 7.3.1 (a) due to Frank [260] (note that this proof requires no prerequisites other than Proposition 7.1.1):

Second proof of Menger's theorem part (a):

Clearly the maximum number of arc-disjoint (s, t)-paths can be no more than the minimum size of an (s, t)-cut.

The proof of the other direction is by induction on the number of arcs in D. Let k denote the size of a minimum (s, t)-cut. The base case is when D has precisely k arcs. Then these all go from s to t and thus D has k arc-disjoint (s,t)-paths. Hence we can proceed to the induction step. Call a vertex set U **tight** if $s \in U, t \notin U$ and $d^+(U) = k$. If some arc xy does not leave any tight set, then we can remove it without creating an (s,t)-cut of size $(k-1)$ and the result follows by induction. Hence we can assume that every arc in D leaves a tight set.

Claim: If X and Y are tight sets, then so are $X \cap Y$ and $X \cup Y$.

To see this we use the submodularity of d^+. First note that each of $X \cap Y$ and $X \cup Y$ contains s and none of them contains t. Hence, by our assumption, they both have degree at least k in D. Now using (7.2) we conclude

$$k + k = d^+(X) + d^+(Y) \geq d^+(X \cup Y) + d^+(X \cap Y) \geq k + k, \qquad (7.5)$$

by the remark above. It follows that each of $X \cup Y$ and $X \cap Y$ is tight and the claim is proved.

If every arc in D is of the from st, then we are done, so we may assume that D has an arc su where $u \neq t$. Let T be the union of all tight sets that do not contain u. Then $T \neq \emptyset$, since the arc su leaves a tight set. By the claim, T is also tight. Now consider the set $T \cup \{u\}$. If there is no arc from u to $V - T$, then $d^+(T \cup \{u\}) \leq k-1$, a contradiction since $T \cup \{u\}$ contains s but not t. Hence there must be some $v \in V - T - u$ such that $uv \in A(D)$. Now let D' be the digraph we obtain from D by replacing the two arcs su, uv by the arc sv. Suppose D' contains an (s,t)-cut of size less than k. That means that some set X containing s but not t has out-degree at most $k-1$ in D'. Since $d^+_D(X) \geq k$ it is easy to see that we must have $s, v \in X$ and $u \notin X$. Hence $d^+_D(X) = k$ and now we get a contradiction to the definition of T (since we know that $v \notin T$). Thus every (s,t)-cut in D' has size at least k. Since D' has fewer arcs than D it follows by induction that D' contains k arc-disjoint (s,t)-paths. At most one of these can use the new arc sv (in which case we can replace this arc by the two we deleted). Thus it follows that D also has k arc-disjoint (s,t)-paths. $\square$

Corollary 7.3.2 *Let $D = (V, A)$ be a directed multigraph. Then the following holds:*

(a) D is k-arc-strong if and only if it contains k-arc-disjoint (s,t)-paths for every choice of distinct vertices $s, t \in V$.
(b) D is k-strong if and only if $|V(D)| \geq k+1$ and D contains k internally vertex disjoint (s,t)-paths for every choice of distinct vertices $s, t \in V$.

Proof: Recall that, by definition, a directed multigraph $D = (V, A)$ is k-arc-strong if and only if $D - A'$ is strong for every $A' \subset A$ with $|A'| \leq k-1$. Now we see that (a) follows immediately from Theorem 7.3.1(a). To prove (b) we argue as follows: By definition (see Chapter 1) D is k-strong if and only if $|V(D)| \geq k+1$ and $D - X$ is strong for every $X \subset V$ such that $|X| \leq k-1$.

Suppose that D has at least $k+1$ vertices but is not k-strong. Then we can find a subset $X \subset V$ of size at most $k-1$ such that $D-X$ is not strong. Let $D_1, \dots, D_r$, $r \geq 2$ be any acyclic ordering of the strong components in $D-X$. Taking $s \in V(D_r)$ and $t \in V(D_1)$ it follows that there is no arc from s to t and that X is an (s,t)-separator of size less than k. Now it follows from Theorem 7.3.1(b) that D does not contain k internally vertex disjoint paths from s to t.

Suppose conversely that there exists $s, t \in V(D)$ such that there are no k internally disjoint (s,t)-paths in D. If there is no arc from s to t, then it follows from Theorem 7.3.1(b) that D contains an (s,t)-separator X of size less than k. Then $D-X$ is not strong and, by definition, D is not k-strong. Hence we may assume that there is an arc st in D. Let r be the number of arcs from s to t in D (i.e. $\mu(s,t)=r$). If $r \geq k$, then k of these arcs form the desired (s,t)-paths, so by our assumption on s,t we have $r<k$. Now consider the digraph D' obtained from D by removing all arcs from s to t. In D' there can be no $k-r$ internally disjoint (s,t)-paths (since otherwise these together with the r arcs from s to t would give a collection of k internally disjoint (s,t)-paths). Thus, by Theorem 7.3.1(b), there exists a set $X' \subset V$ of size less than $k-r$ which forms an (s,t)-separator in D'.

Let A, B denote a partition of $V-X'$ in such a way that $s \in B$, $t \in A$ and there is no arc from B to A in D'. Since $|V| \geq k+1$, at least one of the sets A, B contains more than one vertex. Without loss of generality we may assume that A contains a vertex v distinct from t. Now we see that $X' \cup \{t\}$ is an (s,v)-separator of size less than $k-r+1 \leq k$ in D and there is no arc from s to v in D. Applying Theorem 7.3.1(b) to this pair we conclude as above that D is not k-strong. □

Recall the numbers $\lambda'(D), \kappa'(D)$ which were defined in (7.1).

Corollary 7.3.3 *Let D be a directed multigraph. The number $\lambda'(D)$ equals the maximum number k for which D is k-arc-strong. The number $\kappa'(D)$ equals the maximum number k for which $k \leq |V|-1$ and D is k-strong. Hence we have $\lambda'(D)=\lambda(D)$ and $\kappa'(D)=\kappa(D)$.* □

7.4 Application: Determining Arc- and Vertex-Strong Connectivity

In applications it is often important to be able to calculate the degree of arc-strong or vertex-strong connectivity of a directed multigraph. We can reduce the problem of finding $\kappa_D(x,y)$ to that of finding the local arc-strong connectivity from x_s to y_t in the digraph D_{ST} which we obtain by applying the vertex splitting procedure to D (see the proof of Theorem 7.3.2). Thus it is sufficient to consider arc-strong connectivity. It follows from Menger's theorem and Lemma 7.1.5 that $\lambda(D)$ can be found using $O(n^2)$ flow calculations. Namely, determine $\lambda(x, y)$ for all choices of $x, y \in V(D)$. However, as we shall see below we can actually find $\lambda(D)$ with just $O(n)$ flow calculations. For a similar result see Exercise 7.7.

Proposition 7.4.1 *[654] For any directed multigraph $D = (V, A)$ with $V = \{v_1, v_2, \ldots, v_n\}$ the arc-strong connectivity of D satisfies*

$$\lambda(D) = \min\{\lambda(v_1, v_2), \ldots, \lambda(v_{n-1}, v_n), \lambda(v_n, v_1)\}.$$

Proof: Let $k = \lambda(D)$. By (7.1) and Corollary 7.3.3, $\lambda(D)$ is no more than the minimum of the numbers $\lambda(v_1, v_2), \ldots, \lambda(v_{n-1}, v_n), \lambda(v_n, v_1)$. Hence it suffices to prove that $k = \lambda(v_i, v_{i+1})$ for some $i = 1, 2, \ldots, n$ (where $v_{n+1} = v_1$). By Corollary 7.3.3 and Theorem 7.3.1, some $X \subset V$ has out-degree k. If there is an index $i \leq n - 1$ such that $v_i \in X$ and $v_{i+1} \in V - X$, then, by Menger's theorem, $\lambda(v_i, v_{i+1}) \leq k$ and the claim follows. If no such index exists, then we must have $X = \{v_r, v_{r+1}, \ldots, v_n\}$ for some $1 < r \leq n$. Now we get by Menger's theorem that $\lambda(v_n, v_1) \leq k$ and the proof is complete. □

Combining this with Lemma 7.1.5, we get the following result due to Schnorr [654]:

Corollary 7.4.2 *We can calculate the arc-strong connectivity of a directed multigraph by $O(n)$ maximum flow calculations in $\mathcal{N}(D)$.* □

If D has no multiple arcs, then its network representation $\mathcal{N}(D)$ has all capacities equal to 1 and it follows from Theorem 3.7.4 that we can find a maximum flow in $\mathcal{N}(D)$ in time $O(n^{\frac{2}{3}}m)$ and hence we can calculate $\lambda(D)$ in time $O(n^{\frac{5}{3}}m)$.

Esfahanian and Hakimi [224] showed that the bound, n, on the number of max-flow calculations that is needed can be improved by a factor of at least 2.

Note that, if we are only interested in deciding whether $\lambda(D) \geq k$ for some value of k which is not too big compared to m, then it may be better to use the simple labelling algorithm of Ford and Fulkerson (see Chapter 3). In that case it is sufficient to check for flows of value at least k, which can be done with k flow-augmenting paths and hence in time $O(km)$ per choice of source and terminal. Thus the overall complexity of finding $\lambda(D)$ is $O(knm)$ (see also the book by Even [229]). This can be improved slightly; see the paper [295] by Galil. For other connectivity algorithms based on flows, see e.g. [228, 232].

One may ask if there is a way of deciding whether a given directed multigraph D is k-(arc)-strong without using flows. Extending work by Linial, Lovász and Wigderson [515] (see also [523]), Cheriyan and Reif [150] gave

Monte-Carlo and Las Vegas[2] type algorithms for k-strong connectivity in digraphs. Both algorithms in [150] are based on a characterization of k-strong digraphs via certain embeddings in the Euclidean space $\mathcal{R}^{k-1}$. The algorithms are faster than the algorithms described above, but the price is the chance of an error (for the Monte Carlo algorithm), respectively only the expected running time can be given (for the Las Vegas Algorithm). We refer the reader to [150] for details.

The currently fastest algorithm to determine the arc-strong connectivity uses matroid intersection (see Section 12.7 for the definition of the matroid intersection problem) and is due to Gabow [287]. This algorithm finds the arc-strong connectivity of a digraph D in time $O(\lambda(D)m\log(n^2/m))$. It is based on Edmonds' branching theorem (Theorem 9.5.1). In Chapter 9 we discuss the relation between arc-strong connectivity and arc-disjoint branchings, which is used in Gabow's algorithm. Gabow's approach also works very efficiently for the case when we want to decide if $\lambda(D) \geq k$ for some number k.

The currently fastest algorithm to determine $\kappa(D)$ is due to Henzinger, Rao and Gabow [422]. This algorithm is based on flows and combines ideas from [228, 232, 295, 398]. The complexity of the algorithm is $O(\min\{\kappa(D)^3 + n, \kappa(D)n\}m)$.

For undirected graphs Ibaraki and Nagamochi [579] found a very elegant and effective way to calculate the edge-connectivity without using flow algorithms. We describe their method briefly below (see also [269, 580]).

A **maximum adjacency ordering** of an undirected graph $G = (V, E)$ is an ordering $v_1, v_2, \ldots, v_n$ of its vertices, satisfying the following property

$$d(v_{i+1}, V_i) \geq d(v_j, V_i) \text{ for } i = 1, 2, \ldots, n, i < j \leq n, \tag{7.6}$$

where $V_i = \{v_1, v_2, \ldots, v_i\}$ and $d(X, Y)$ denotes the number of edges with one end in $X - Y$ and the other in $Y - X$.

Theorem 7.4.3 *[579]*

(a) Given any undirected graph G on n vertices, one can find a maximum adjacency ordering of G starting at a prescribed vertex v_1 in time $O(n + m)$.
(b) For every maximum adjacency ordering $v_1, v_2, \ldots, v_n$ of G we have $\lambda(v_{n-1}, v_n) = d_G(v_n)$.

Corollary 7.4.4 *[579] There is an $O(nm + n^2)$ algorithm to determine the edge-connectivity of a graph with n vertices and m edges.*

[2] A Monte-Carlo algorithm always terminates, but may make an error with some small probability, whereas a Las Vegas algorithm may (with some small probability) never terminate, but if it does, then the answer it provides is correct; see the book [134] by Brassard and Bratley.

Proof: This is an easy consequence of (b) and the fact that for every choice of $x, y \in V(G)$:

$$\lambda(G) = \min\{\lambda(x,y), \lambda(G/\{x,y\})\}, \tag{7.7}$$

where $G/\{x,y\}$ is the graph we obtain from G by contracting the set $\{x,y\}$. The equality (7.7) follows from the fact that $\lambda(G)$ equals the size of a minimum cut $(X, V-X)$ in G. If this cut separates x,y, then $\lambda(G) = \lambda(x,y)$ by Menger's theorem, and otherwise X is still a cut in $G/\{x,y\}$, implying that $\lambda(G) = \lambda(G/\{x,y\})$ (contractions do not decrease edge-connectivity). Hence we can start from an arbitrary maximum adjacency ordering $v_1, v_2, \ldots, v_n$. This gives us $\lambda(v_{n-1}, v_n)$. Save this number, contract $\{v_{n-1}, v_n\}$ and continue with a maximum adjacency ordering of $G/\{v_{n-1}, v_n\}$. The edge-connectivity of G is the minimum of the numbers saved. We leave the remaining details to the interested reader (see also the paper [581] by Nagamochi and Ibaraki). $\square$

y x xy

D $D/\{x,y\}$

Figure 7.4 A digraph D with $\lambda(D) = 0$, $\lambda(x,y) = 2$ and $\lambda(D/\{x,y\}) = 1$.

It is an interesting open problem whether some similar kind of ordering can be used to find the arc-strong connectivity of a directed multigraph. Note that (7.7) does not hold for arbitrary directed multigraphs. To see this consider Figure 7.4.

7.5 The Splitting off Operation

In Frank's proof of Menger's theorem in Section 7.3, we saw how one could apply the idea of replacing two arcs incident to some vertex by one and thereby apply induction. In this section we shall see yet another indication that this type of operation can be very useful. We consider a directed multigraph D with a special vertex s. We always assume that

$$d^+_D(s) = d^-_D(s). \tag{7.8}$$

To emphasize that s is a special vertex we specify D as $D = (V+s, A)$ or $D = (V+s, E \cup F)$ where F is the set of arcs with one end-vertex in s. Furthermore we will assume that the local arc-strong connectivity between every pair x, y of vertices in V is at least k. By Menger's theorem this is equivalent to

$$d^+(U), d^-(U) \geq k \text{ for all } \emptyset \neq U \subset V. \tag{7.9}$$

Whenever a digraph $D = (V+s, A)$ satisfies (7.9) for some k we say that D is **k-arc-strong in V**.

We consider the operation of replacing a pair (us, sv) of arcs incident with s by one new arc uv. The operation of performing this replacement is called **splitting off** or just **splitting** the pair (us, sv) and the resulting directed multigraph is denoted by D_{uv}. The splitting of a pair (us, sv) is **admissible** if (7.9) holds in D_{uv}. If this is the case we will also say that the pair (us, sv) is an **admissible pair** (or an **admissible splitting**).

A set $\emptyset \neq X \subset V$ is **k-in-critical** (**k-out-critical**) if $d^-(X) = k$ ($d^+(X) = k$). When we do not want to specify whether X is k-in-critical or k-out-critical, we say that X is **k-critical**.

The following useful lemma is due to Frank:

Lemma 7.5.1 *[258] If X and Y are intersecting k-critical sets then one of the following holds:*

(a) $X \cup Y$ is k-critical,
(b) $Y - X$ is k-critical and $d(X \cap Y, V+s-(X \cup Y)) = 0$.

Proof: We consider three cases:
Case 1: $X \cup Y \neq V$ and X, Y are either both k-out-critical or both k-in-critical.

Assume that X, Y are both k-out-critical. It follows from (7.9) that $d^+(X \cup Y), d^+(X \cap Y) \geq k$. Using the submodularity of d^+_D, we obtain:

$$\begin{aligned} k + k &= d^+(X) + d^+(Y) \\ &\geq d^+(X \cup Y) + d^+(X \cap Y) \\ &\geq k + k, \end{aligned} \tag{7.10}$$

and from this we get that $X \cup Y$ is k-critical and hence (a) holds. The same conclusion is reached if X, Y are both k-in-critical.
Case 2: $X \cup Y = V$ and X, Y are either both k-out-critical or both k-in-critical.

We will assume that X, Y are both k-out-critical, the proof is analogous in the other case. Let $S = V+s-X$ and $T = V+s-Y$. Then $d^-(S) = d^-(T) = k$ and $S \cap T = \{s\}$. Since $S - T = Y - X$ and $T - S = X - Y$ we get from (7.9) that $d^-(S-T), d^-(T-S) \geq k$. Since $d^-(s) = d^+(s)$, we can apply (7.3) and obtain:

$$\begin{aligned} k + k &= d^-(S) + d^-(T) \\ &= d^-(S - T) + d^-(T - S) + d(S \cap T, V + s - (S \cup T)) \\ &\geq k + k + d(V - S, T), \end{aligned} \tag{7.11}$$

from which we see that $Y - X = S - T$ is k-in-critical and that $d(S \cap T, V + s - (S \cup T)) = 0$. Since $X \cap Y = V + s - (S \cup T)$ and $V + s - (X \cup Y) = \{s\} = S \cap T$ we also see that $d(X \cap Y, V + s - (X \cup Y)) = 0$. Thus (b) holds.

Case 3: One of $\boldsymbol{X, Y}$ is $\boldsymbol{k}$-in-critical and the other is $\boldsymbol{k}$-out-critical.

We consider the case when X is k-in-critical and Y is k-out-critical, the other case is analogous. Let $Z = V + s - X$. Then we have $d^+(Y) = d^+(Z) = k$, $Y \cap Z = Y - X$ and $Y \cup Z = V + s - (X - Y)$. Hence $d^+(Y \cap Z) = d^+(Y - X) \geq k$ and $d^+(Y \cup Z) = d^-(V + s - (Y \cup Z)) = d^-(X - Y) \geq k$. Now we can apply (7.2) and we get

$$\begin{aligned} k + k &= d^+(Y) + d^+(Z) \\ &= d^+(Y \cap Z) + d^+(Y \cup Z) + d(Y, Z) \\ &\geq k + k + d(Y, Z), \end{aligned} \tag{7.12}$$

implying that $d^+(Y - X) = d^+(Y \cap Z) = k$ and that $d(Y, Z) = 0$. Since $Z - Y = V + s - (X \cup Y)$ and $Y - Z = X \cap Y$, the last equality shows that $d(X \cap Y, V + s - (X \cup Y)) = 0$. Thus (b) holds. □

We are now ready to prove the following important result by Mader.

Theorem 7.5.2 (Mader's directed splitting theorem) *[537] Suppose that $D = (V + s, E \cup F)$ satisfies (7.9) and that $d^+(s) = d^-(s)$. Then for every arc sv there is an arc us such that the pair (us, sv) is an admissible splitting.*

Proof: The proof we give is due to Frank [258]. First note that a pair (us, sv) can be split off preserving (7.9) if and only if there is no k-critical set which contains both u and v. Hence if there is no k-critical set containing v, then we are done. If X and Y are intersecting k-critical sets containing v, then only alternative (i) can hold in Lemma 7.5.1, because the existence of the arc sv implies that $d(V + s - (X \cup Y), X \cap Y) \geq 1$. Hence the union T of all k-critical sets containing v is also k-critical. If we can find an in-neighbour u of s in $V - T$, then we are done, since by the choice of T, there is no k-critical set which contains u and v. So suppose that all in-neighbours of s are in T. If T is k-out-critical then

$$\begin{aligned} d^-(V - T) &= d^+(T) - d^+(T, s) + d^+(s, V - T) \\ &\leq k - (d^-(s) - d^+(s) + 1) \\ &= k - 1, \end{aligned}$$

since s has no in-neighbour in $V-T$ and sv is an arc from s to T (we also used $d^-(s)=d^+(s)$). This contradicts (7.9) so we cannot have that T is k-out-critical. But if T is k-in-critical, then

$$\begin{aligned} d^+(V-T) &= d^-(T+s) = d^-(T) - d^+(s,T) + d^+(V-T,s) \\ &\leq k-1+0 < k, \end{aligned}$$

a contradiction again. Hence we have shown that (us, sv) is an admissible pair and the proof is complete. □

s

Figure 7.5 A digraph $D=(V+s,A)$ which is 2-arc-strong in V and has no admissible splitting at s. Note that $d^-(s)=2\neq 1=d^+(s)$.

Note that the assumption that $d^-(s)=d^+(s)$ in Theorem 7.5.2 cannot be removed. Figure 7.5 shows an example of a digraph $D=(V+s,A)$ with no admissible splitting at s.

Corollary 7.5.3 *Suppose that $D=(V+s,E+F)$ satisfies (7.9) and that $d^+(s)=d^-(s)$. Then there exists a pairing $((u_1s,sv_1),\dots,(u_rs,sv_r))$, $r=d^-(s)$, of the arcs entering s with the arcs leaving s such that replacing all arcs incident with s by the arcs $u_1v_1,\dots,u_rv_r$ and then deleting s, we obtain a k-arc-strong directed multigraph D'.* □

See Figure 7.6 for an example of a complete splitting in a digraph.

Frank and Jackson showed that for eulerian directed multigraphs one can get a stronger result. Namely, it is possible to split off all arcs incident with the special vertex s in such a way that all local arc-strong connectivities within V are preserved.

Theorem 7.5.4 *[257, 451] Let $D=(V+s,A)$ be an eulerian directed multigraph. Then for every arc $us\in A$ there exists an arc $sv\in A$ such that $\lambda_{D_{uv}}(x,y)=\lambda_D(x,y)$ for all $x,y\in V$.* □

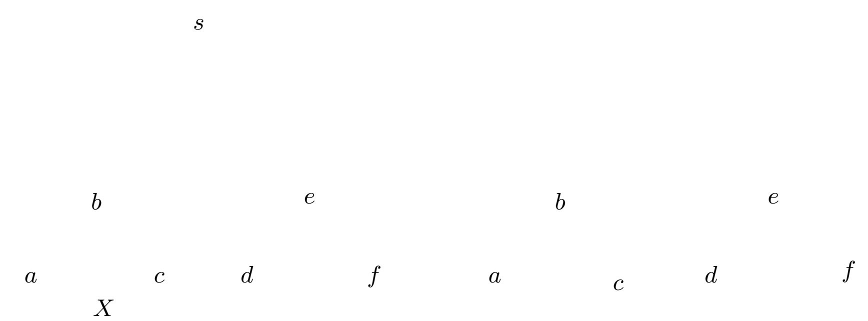


Figure 7.6 A digraph $D = (V + s, A)$ which is 2-arc-strong in V. A complete splitting of the arcs is shown in the right figure after removal of s. The set X shows that we cannot split off both of the pairs $(as, sb), (cs, sa)$, since that would leave X with out-degree one.

A similar result concerning local connectivity preserving splittings holds for general undirected graphs. This very powerful result was proved by Mader [536]. Such a similarity between eulerian digraphs and general undirected graphs with respect to certain properties seems to be quite common. To say it popularly: Eulerian digraphs often behave like undirected graphs. For another example of this phenomenon see Section 9.7.2.

Bang-Jensen, Frank and Jackson showed that it is possible to give a common generalization of Theorem 7.5.4 and Mader's directed splitting theorem (Theorem 7.5.2) to mixed graphs. Since the statement of this result is rather technical, we refer the interested reader to the paper [53].

It was pointed out by Enni in [218] that Theorem 7.5.4 cannot be extended to arbitrary digraphs, not even if one only wants to preserve the minimum of $\lambda(x, y)$ and $\lambda(y, x)$. For two other generalizations of Theorem 7.5.2 see the papers [684] by Su and [288] by Gabow and Jordán.

7.6 Increasing the Arc-Strong Connectivity Optimally

We will consider the following problem. Given a directed multigraph $D = (V, E)$ which is not k-arc-strong, find a minimum cardinality set of new arcs F to add to D such that the resulting directed multigraph $D' = (V, E \cup F)$ is k-arc-strong. This D' is called an **optimal augmentation** of D. We will present a solution to this problem due to Frank [258]. Frank solved the problem by supplying a min-max formula for the minimum number of new arcs as well as a polynomial algorithm to find such a minimum set of new arcs. First let us make the simple observation that such a set F indeed exists, since we may just add k parallel arcs in both directions between a fixed vertex $v \in V$ and all other vertices in V (it is easy to see that the resulting directed multigraph will be k-arc-strong).

Definition 7.6.1 *Let $D = (V, A)$ be a directed multigraph. Then $\gamma_k(D)$ is the smallest integer γ such that*

$$\sum_{X_i \in \mathcal{F}} (k - d^-(X_i)) \leq \gamma \text{ and}$$
$$\sum_{X_i \in \mathcal{F}} (k - d^+(X_i)) \leq \gamma,$$

for every subpartition $\mathcal{F} = \{X_1, \ldots, X_t\}$ of V.

We call $\gamma_k(D)$ the **subpartition lower bound for arc-strong connectivity**. By Menger's theorem, D is k-arc-strong if and only if $\gamma_k(D) \leq 0$. Indeed, if D is k-arc-strong, then $d^+(X), d^-(X) \geq k$ holds for all proper subsets of V and hence we see that $\gamma_k(D) \leq 0$. Conversely, if D is not k-arc-strong, then let X be a set with $d^-(X) < k$. Take $\mathcal{F} = \{X\}$, then we see that $\gamma_k(D) \geq k - d^-(X) > 0$.

Lemma 7.6.2 *[258] Let $D = (V, A)$ be a directed multigraph and let k be a positive integer such that $\gamma_k(D) > 0$. Then D can be extended to a new directed multigraph $D' = (V + s, A \cup F)$, where F consists of $\gamma_k(D)$ arcs whose head is s and $\gamma_k(D)$ arcs of whose tail is s such that (7.9) holds in D'.*

Proof: We will show that, starting from D, it is possible to add $\gamma_k(D)$ arcs from V to s so that the resulting graph satisfies

$$d^+(X) \geq k \quad \text{for all} \quad \emptyset \neq X \subset V. \tag{7.13}$$

Then it will follow analogously (by considering the converse of D) that it is also possible to add $\gamma_k(D)$ new arcs from s to V so that the resulting graph satisfies

$$d^-(X) \geq k \quad \text{for all} \quad \emptyset \neq X \subset V. \tag{7.14}$$

First add k parallel arcs from v to s for every $v \in V$. This will certainly make the resulting directed multigraph satisfy (7.13). Now delete as many new arcs as possible until removing any further arc would result in a digraph where (7.13) no longer holds (that is, every remaining new arc vs leaves a k-out-critical set). Let $\tilde{D}$ denote the current directed multigraph after this deletion phase and let S be the set of vertices v which have an arc to s in $\tilde{D}$. Let $\mathcal{F} = \{X_1, \ldots, X_r\}$ be a family of k-out-critical sets such that every $v \in S$ is contained in some member X_i of $\mathcal{F}$ and assume that $\mathcal{F}$ has as few members as possible with respect to this property. Clearly this choice implies that either $\mathcal{F}$ is a subpartition of V, or there is a pair of intersecting sets X_i, X_j in $\mathcal{F}$.

Case 1: $\mathcal{F}$ is a subpartition of V.

Then we have

$$\begin{aligned} kr &= \sum_{i=1}^{r} d^+_{\tilde{D}}(X_i) \\ &= \sum_{i=1}^{r} (d^+_D(X_i) + d^+_{\tilde{D}}(X_i, s)) \\ &= \sum_{i=1}^{r} d^+_D(X_i) + d^-_{\tilde{D}}(s), \end{aligned}$$

implying that $d^-_{\tilde{D}}(s) = \sum_{i=1}^{r}(k - d^+_D(X_i)) \leq \gamma_k(D)$, by the definition of $\gamma_k(D)$.

Case 2: Some pair $X_i, X_j \in \mathcal{F}$ is intersecting.

If X_i, X_j are crossing, then the submodularity of $d^+_{\tilde{D}}$ and (7.9) imply that $X_i \cup X_j$ is also k-out-critical and hence we could replace the two sets X_i, X_j by the set $X_i \cup X_j$ in $\mathcal{F}$, contradicting the choice of $\mathcal{F}$. Hence we must have $X_i \cup X_j = V$ and $\mathcal{F} = \{X_1, X_2\}$, where without loss of generality $i = 1, j = 2$. Let $X = V - X_1 = X_2 - X_1$ and $Y = V - X_2 = X_1 - X_2$. Then $d^-_D(X) = d^+_D(X_1)$ and $d^-_D(Y) = d^+_D(X_2)$ and hence we get

$$\begin{aligned} \gamma_k(D) &\geq (k - d^-_D(X)) + (k - d^-_D(Y)) \\ &= k - d^+_D(X_1) + k - d^+_D(X_2) \\ &\geq k - d^+_{\tilde{D}}(X_1) + k - d^+_{\tilde{D}}(X_2) + d^-_{\tilde{D}}(s) \\ &= d^-_{\tilde{D}}(s), \end{aligned}$$

since X_1, X_2 are k-out-critical in $\tilde{D}$. Thus $d^-_{\tilde{D}}(s) \leq \gamma_k(D)$ as claimed. □

Theorem 7.6.3 (Frank's arc-strong connectivity augmentation theorem) *[258] Let $D = (V, A)$ be a digraph and k a natural number such that $\gamma_k(D) > 0$. The minimum number of new arcs that must be added to D in order to give a k-arc-strong digraph $D' = (V, A \cup F)$ equals $\gamma_k(D)$.*

Proof: To see that we must use at least $\gamma_k(D)$ arcs, it suffices to observe that if X and Y are disjoint sets then no new arc can increase the out-degree (in-degree) of both sets. Hence a subpartition $\mathcal{F}$ realizing the value of γ_k in Definition 7.6.1 is a certificate that we must use at least $\gamma_k(D)$ new arcs.

To prove the other direction we use Mader's splitting theorem and Lemma 7.6.2. According to this lemma we can extend D to a new digraph $\tilde{D}$ by adding a new vertex s and $\gamma_k(D)$ arcs from V to s and from s to V. Note that we may not need $\gamma_k(D)$ arcs in both directions, but we will need it in one of the directions by our remark in the beginning of the proof. In the case where fewer arcs are needed, say from V to s we add arbitrary arcs from V to s so that the resulting number becomes $\gamma_k(D)$.

Now it follows from Corollary 7.5.3 that all arcs incident with s can be split off without violating (7.9). This means that, if we remove s, then the resulting graph D' is k-arc-strong. □

See Figure 7.7 for an example illustrating the theorem.

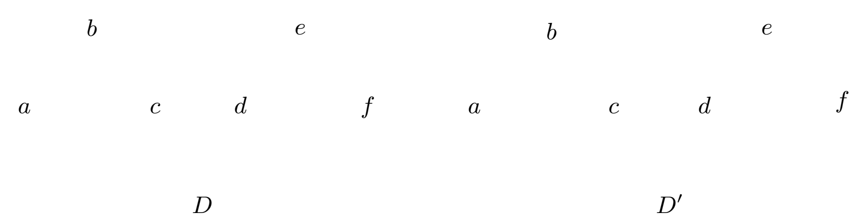


Figure 7.7 A digraph D with $\gamma_2(D) = 5$. The big circles indicate a subpartition which realizes $\gamma_2(D)$. The right part of the figure shows an optimal 2-arc-strong augmentation D' of D obtained by adding 5 new arcs. Compare this with Figure 7.6. Here the digraph in the right part is the same as the augmented digraph D'.

The reader may have noticed that in the proof of Lemma 7.6.2, we never used exactly how we obtained the minimal set of arcs from V to s so that (7.13) held. The proof is valid for every such set of arcs that is minimal with respect to deletion of arcs. This means in particular that we can use a greedy approach to find such a set of arcs starting from the configuration with k parallel arcs from every vertex $v \in V$ to s. This gives rise to the following algorithm, by Frank [258], for augmenting the arc-strong connectivity optimally to k for any digraph D which is not already k-arc-strong:

Frank's arc-strong connectivity augmentation algorithm
Input: A directed multigraph $D = (V, A)$ and a natural number k such that $\gamma_k(D) > 0$.
Output: A k-arc-strong optimal augmentation D^* of D.

1. Let $v_1, v_2 \dots, v_n$ be a fixed ordering of V and let s be a new vertex.
2. Add k parallel arcs from v_i to s and from s to v_i for each $i = 1, 2, \dots, n$.
3. Starting from $i := 1$, remove as many arcs from v_i to s as possible without violating (7.13); If $i < n$ then let $i := i + 1$ and repeat this step;
   Let γ^- denote the number of remaining arcs from V to s in the resulting digraph.
4. Starting from $i := 1$, remove as many arcs from s to v_i as possible without violating (7.14); If $i < n$ then $i := i + 1$ and repeat this step;

Let γ^+ denote the number of remaining arcs from s to V in the resulting digraph.
5. Let $\gamma = \max\{\gamma^-, \gamma^+\}$. If $\gamma^- < \gamma^+$, then add $\gamma^+ - \gamma^-$ arcs from v_1 to s; If $\gamma^+ < \gamma^-$, then add $\gamma^- - \gamma^+$ arcs from s to v_1.
6. Let D' denote the current digraph. In D' we have $d^-_{D'}(s) = d^+_{D'}(s)$ and (7.9) holds. Split off all arcs incident with s in D' by applying Theorem 7.5.2 γ times. Let D^* denote the resulting directed multigraph.
7. Return D^*.

Using flows this algorithm can be implemented as a polynomial algorithm for augmenting the arc-strong connectivity of a given digraph [258]. See Exercises 7.28 and 7.30.

Frank [258] pointed out that his algorithm also works for the so-called **vertex-weighted arc-strong connectivity augmentation problem**. Here there are weights $c(v)$ on the vertices and the cost of adding an arc from u to v is equal to $c(u) + c(v)$. The only change needed in the algorithm above is that now the ordering of the vertices should be so that $c(v_1) \leq c(v_2) \leq \ldots \leq c(v_n)$. The reason why this greedy approach works is outlined in [258] and comes from the fact that a certain polymatroidal structure is present [258, 274].

If instead we allow weights on the arcs and ask for a minimum weight (rather than just minimum cardinality) set of new arcs to add to D in order to obtain a k-arc-strong digraph D', then we have the **weighted arc-strong connectivity augmentation problem**.

Theorem 7.6.4 *The weighted arc-strong connectivity augmentation problem is $\mathcal{NP}$-hard.*

Proof: We show that the Hamilton cycle problem can be reduced to the weighted arc-strong connectivity augmentation problem in polynomial time. This will imply the claim by Theorem 5.0.1.

Let $D = (V, A)$ be a digraph on n vertices $V = \{1, 2, \ldots, n\}$. Define weights $c(ij)$ on the arcs of the complete digraph $\overleftrightarrow{K}_n$ with vertex set V as follows:

$$c(ij) = \begin{cases} 1 & \text{if } ij \in A \\ 2 & \text{if } ij \notin A. \end{cases} \tag{7.15}$$

Let $D_0 = (V, \emptyset)$ (that is, the digraph on V with no arcs). Since every vertex of a strong digraph is the tail of at least one arc, we need at least n arcs to make D_0 strong. Now it is easy to see that D_0 can be made strongly connected using arcs with total weight at most n if and only if D has a Hamilton cycle. Thus we have reduced the Hamilton cycle problem to the weighted arc-strong connectivity augmentation problem. Clearly our reduction can be carried out in polynomial time. $\square$

We complete this section with an interesting result by Cheng and Jordán. It implies that the so-called **successive augmentation property** holds for arc-strong connectivity.

Theorem 7.6.5 *[148] Let D be a directed multigraph with $\lambda(D) = \ell$. Then there exists an infinite sequence $D = D_0, D_1, D_2, \ldots$ of directed multigraphs such that, for every $i \geq 0$, D_{i+1} is a superdigraph of D_i, $V(D_i) = V(D)$ and D_i is an optimal $(\ell + i)$-arc-strong augmentation of D.* □

It is shown by an example in [148] that a similar property does not hold for the vertex-strong connectivity augmentation problem which we consider below.

7.7 Increasing the Vertex-Strong Connectivity Optimally

We now turn to the **vertex-strong connectivity augmentation problem**: given a digraph $D = (V, A)$ on at least $k + 1$ vertices, find a smallest set F of new arcs for which $D' = (V, A \cup F)$ is k-strong.

Note that when it comes to studying vertex-strong connectivity, multiple arcs play no role and hence we shall always consider digraphs (knowing that our results extend to directed multigraphs). In particular, in this section $d^+_D(v) = |N^+_D(v)|$ for any vertex v in a digraph D.

Let us first observe that, even if we do not allow multiple arcs, we cannot bound the number of arcs we need to add to make a digraph D k-strong by some function of $\gamma_k(D)$ (recall Definition 7.6.1). To see this, it suffices to note that there are k-arc-strong digraphs which are not k-strong and one can construct such digraphs where the number of new arcs one needs to add in order to obtain a k-strong superdigraph is arbitrarily high (see Exercise 7.31).

Suppose X is a set of vertices in a digraph D such that $N^+[X] \neq V$ and $|N^+(X)| < k$ (recall that $N^+[X] = X \cup N^+(X)$). Then it follows from Menger's theorem that D is not k-strong because the set $N^+(X)$ separates every vertex in X from every vertex in $V - N^+[X]$. Furthermore, in order to obtain a k-strong digraph by adding arcs to D we must add at least $k - |N^+(X)|$ new arcs with tail in X and head in $V - X$.

Similarly to the definition of $\gamma_k(D)$ in Definition 7.6.1 we can define $\gamma^*_k(D)$ as follows:

Definition 7.7.1 *Let $D = (V, A)$ be a directed graph. Then $\gamma^*_k(D)$ is the smallest integer γ such that*

$$\sum_{X \in \mathcal{F}^-} (k - |N^-(X)|) \leq \gamma \text{ and}$$
$$\sum_{X \in \mathcal{F}^+} (k - |N^+(X)|) \leq \gamma,$$

for every choice of subpartitions $\mathcal{F}^-, \mathcal{F}^+$ of V with the property that every $X \in \mathcal{F}^-$ satisfies $N^-[X] \neq V$ and every $X \in \mathcal{F}^+$ satisfies $N^+[X] \neq V$.

As with arc-strong connectivity it is not hard to see that $\gamma_k^*(D)$ is a lower bound for the number of new arcs we must add to D to obtain a k-strong digraph. This follows from the fact that the sets in $\mathcal{F}^-$ are disjoint and hence no new arc can increase the in-neighbourhoods (out-neighbourhoods) of two sets from $\mathcal{F}^-$ ($\mathcal{F}^+$). We call the number $\gamma_k^*(D)$ the **subpartition lower bound for vertex-strong connectivity**.

Let $a_k(D)$ denote the minimum number of new arcs that must be added to a digraph $D = (V, A)$ in order to obtain a k-strong digraph. It is easy to see that $a_k(D)$ is well-defined provided that D has at least $k+1$ vertices. We also call $a_k(D)$ the ***k*-strong augmentation number** of D.

7.7.1 One-Way Pairs

First we point out that for vertex-strong connectivity augmentation, the subpartition lower bound is no longer sufficient, that is, it may not be possible to make D k-strong by adding $\gamma_k^*(D)$ arcs. An example illustrating this is given in Figure 7.8(a). Here $k = 2$ and it is not difficult to check that $\gamma_k^*(D) = 2$. However, it is not possible to make D 2-strong by adding just two new arcs. In order to explain this, we need a few new definitions. Let X, Y be disjoint non-empty proper subsets of V. The ordered pair (X, Y) is a **one-way pair** in $D = (V, A)$ if D has no arc with tail in X and head in Y (that is, $Y \Rightarrow X$). This definition is due to Frank and Jordán [272]. For such a pair (X, Y) we refer to X (Y) as the **tail** (**head**) of the pair. Let $h(X, Y) = |V - X - Y|$. The **deficiency** of a one-way pair (X, Y) with respect to k-strong connectivity is

$$\eta_k(X, Y) = \max\{0, k - h(X, Y)\}. \tag{7.16}$$

For instance, if $N^+[X] \neq V$ then the pair $(X, V - N^+[X])$ is a one-way pair with deficiency $\eta_k(X, V - N^+[X]) = \max\{0, k - |N^+(X)|\}$. One-way pairs are closely related to k-strong connectivity.

Lemma 7.7.2 *[272] A digraph $D = (V, A)$ is k-strong if and only if we have $h(X, Y) \geq k$ for every one-way pair (X, Y) in D.*

Proof: Suppose first that D is k-strong. By Corollary 7.3.2, there are k internally disjoint (s, t)-paths for every choice of distinct vertices $s, t \in V$. Now let (X, Y) be a one-way pair and take $s \in X, t \in Y$. For every collection

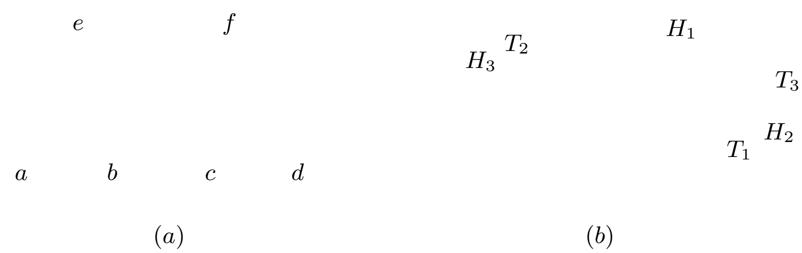


Figure 7.8 An example, due to Jordán [468, Figure 3.9.1], showing that the subpartition lower bound is not always attainable. The desired connectivity is $k = 2$ and the value $\gamma_2^*(D)$ is 2 and it is realized by the subpartitions $\{\{d\},\{e\}\},\{\{a\},\{f\}\}$, respectively (see (a)). Part (b) shows three pairwise independent one-way pairs $(T_1, H_1), (T_2, H_2), (T_3, H_3)$ (tails are indicated by boxes). This shows that $a_2(D) \geq 3$. In fact $a_2(D) = 3$, since adding the arcs af, ed, da will result in a 2-strong digraph.

of the k internally disjoint paths from s to t, each such path must use a vertex in $V - X - Y$ and hence $h(X,Y) \geq k$. Conversely, assume that $h(X,Y) \geq k$ for every one-way pair (X,Y). Let S be a minimal separator of D. By the definition of a separator, $V - S$ can be divided into two sets X, Y so that there is no arc from X to Y in $D - S$ (namely let s, t be separated by S and let X denote those vertices that can be reached from s in $D - S$ and $Y = V - X - S$). Thus (X,Y) is a one-way pair and $h(X,Y) = |S|$ showing that $|S| \geq k$ and hence D is k-strong. □

Two one-way pairs $(X,Y),(X',Y')$ are **independent** if either their heads or their tails are disjoint. Hence one-way pairs that contribute to the sums in Definition 7.7.1 are always independent since either all heads or all tails are disjoint for those pairs. As we saw in Figure 7.8, the sum of deficiencies over one way pairs for which either all tails are disjoint or all heads are disjoint does not always provide the right lower bound for the number of new arcs needed in order to make the digraph k-strong.

By Lemma 7.7.2, in order to obtain a k-strong superdigraph of D, we must add enough new arcs to eliminate all one-way pairs with $\eta_k(X,Y) > 0$ (we must add at least $\eta_k(X,Y)$ arcs from X to Y). Clearly, if $(X,Y),(X',Y')$ are independent one-way pairs, then no new edge can decrease both $\eta_k(X,Y)$ and $\eta_k(X',Y')$. This shows that, if $\mathcal{F}$ is any family of pairwise independent one-way pairs in D, then we must add at least

$$\eta_k(\mathcal{F}) = \sum_{(X,Y)\in\mathcal{F}} \eta_k(X,Y) \tag{7.17}$$

new arcs to D in order to obtain a k-strong digraph. We call the number $\eta_k(\mathcal{F})$ the **deficiency** of $\mathcal{F}$. Now consider Figure 7.8(b). Here we have indicated one-way pairs (T_i, H_i), $i = 1, 2, 3$. These are pairwise independent and have total deficiency 3. Thus it follows from our arguments above that we need at least 3 new arcs to make D k-strong. In fact 3 arcs are sufficient in this case as pointed out in the caption of the figure.

7.7.2 Optimal k-Strong Augmentation

The following theorem, due to Frank and Jordán, shows that the maximum deficiency over families of independent one-way pairs gives the right lower bound for the vertex-strong connectivity augmentation problem.

Theorem 7.7.3 (The Frank-Jordán vertex-strong connectivity augmentation theorem) *[272] For every digraph D on at least $k+1$ vertices we have*

$$a_k(D) = \max\{\eta_k(\mathcal{F}) : \mathcal{F} \text{ is a family of independent one-way pairs in } D\}. \tag{7.18}$$

In Section 7.8 we will show how to derive Theorem 7.7.3 from a more general result concerning a generalization of arc-connectivity augmentation.

Theorem 7.7.4 *[272] There exists a polynomial algorithm which, given a digraph $D = (V, A)$ and a natural number k, finds a minimum cardinality set F of new arcs to add to D so that the resulting graph is k-strong.* □

This algorithm relies on Theorem 7.7.3 and the ellipsoid method[3]and hence it is not a combinatorial algorithm. In [273] a combinatorial polynomial algorithm was found for fixed k. It is beyond the scope of this book to describe any of these algorithms here. The combinatorial algorithm in [273] relies on a detailed study of the structure of one-way pairs. We refer to the proof of Lemma 7.10.6 for an example of a proof that uses the structure of one-way pairs.

Although we may have $a_k(D) > \gamma_k^*(D)$ as we saw in Figure 7.8, Frank and Jordán proved (see below) that the difference cannot be arbitrary large. A family $\mathcal{F}$ of independent one-way pairs is **subpartition-type** if either all the tails in $\mathcal{F}$ are pairwise disjoint, or all the heads in $\mathcal{F}$ are pairwise disjoint. It is easy to see that if $\mathcal{F}$ is subpartition-type, then $\eta_k(\mathcal{F}) \leq \gamma_k^*(D)$.

Proposition 7.7.5 *[273] For any digraph $D = (V, A)$ and any target connectivity k there exists a family $\mathcal{F}$ of independent one-way pairs such that the deficiency, $\eta_k(\mathcal{F})$, of $\mathcal{F}$ equals $a_k(D)$ and $\mathcal{F}$ is either subpartition-type or the disjoint union of two families of subpartition-type. Thus $a_k(D) \leq 2\gamma_k^*(D)$.*

The next result shows that if we need to add many arcs to D (in terms of k) to make it k-strong, then the subpartition lower bound is attainable.

---

[3] For a thorough treatment of the ellipsoid method and its consequences for Combinatorial Optimization, see the book [339] by Grötschel, Lovász and Schrijver.

Proposition 7.7.6 *[273] If $\mathcal{F}$ is a family of independent one-way pairs and $\eta_k(\mathcal{F}) \geq 2k^2 - 1$, then $\mathcal{F}$ is subpartition-type. Hence if $a_k(D) \geq 2k^2 - 1$, then $\gamma_k^*(D) = a_k(D)$.*

Now let us consider the special case of the vertex-strong connectivity augmentation problem when we want to increase $\kappa(D)$ from k to $k+1$. The following result is due to Frank and Jordán:

Theorem 7.7.7 *[273] If $\kappa(D) = k$ and $a_{k+1}(D) \geq 2k+2$, then $a_{k+1}(D) = \gamma_{k+1}^*(D)$.*

Frank and Jordán also showed that when we augment the connectivity by just one, then we can restrict the structure of the set of new arcs.

Theorem 7.7.8 *[272] If $\kappa(D) = k$, then D can be optimally augmented to a $(k+1)$-strong digraph by adding disjoint cycles and paths. In particular if D is a k-strong and k-regular digraph, then there are disjoint cycles covering V whose addition to D gives a $(k+1)$-strong and $(k+1)$-regular digraph.*

It is instructive to compare this result with Theorem 7.10.7.

Recently, Frank has shown that the problem of augmenting the connectivity by one can be solved in polynomial time without using the ellipsoid method.

Theorem 7.7.9 *[266] There exists a combinatorial polynomial algorithm for increasing the vertex-strong connectivity of a digraph by one.*

7.7.3 Special Classes of Digraphs

For general digraphs one cannot say much about the structure of families of independent one-way pairs, but as we are going to see, there are (non-trivial) classes of digraphs for which nice structure can be found and hence a good estimate on the value of $a_k(D)$ can be given. The first result, due to Masuzawa, Hagihara and Tokura, deals with in-branchings.

Theorem 7.7.10 *[555] Let $B = (V, A)$ be an in-branching. Then $a_k(B)$ is given by*

$$a_k(B) = \sum_{v \in V} \max\{0, k - d^+(v)\}.$$

□

The proof of this result in [555] is long, but Frank and Jordán found a short proof based on Theorem 7.7.3, see [273].

For an arbitrary digraph we define ${\eta_k}^-, {\eta_k}^+$ by

$$\eta_k^-(D) = \sum_{v \in V} \max\{0, k - d^-(v)\}, \tag{7.19}$$

$$\eta_k^+(D) = \sum_{v \in V} \max\{0, k - d^+(v)\}. \tag{7.20}$$

Frank made the following conjecture, which would imply that we have $a_k(D) = \gamma_k(D)$ for every acyclic digraph D:

Conjecture 7.7.11 *[261] For any acyclic digraph D on at least $k+1$ vertices $a_k(D) = \max\{\eta_k{}^-(D), \eta_k{}^+(D)\}$.*

A partial result was obtained by Frank and Jordán in [273].

Lemma 7.7.12 *[273] Let $D = (V, A)$ be an acyclic digraph for which $a_k(D) = \gamma_k^*(D)$. Then $a_k(D) = \max\{\eta_k{}^-(D), \eta_k{}^+(D)\}$.*

Proof: Since $a_k(D) = \gamma_k^*(D)$ there exists some family $\mathcal{F}$ of independent one-way pairs with $\eta_k(\mathcal{F}) = a_k(D)$ such that all tails, or all heads, in $\mathcal{F}$ are pairwise disjoint. By considering the converse of D if necessary, we may assume that the tails $\{T_1, \ldots, T_t\}$ of $\mathcal{F}$ are pairwise disjoint.

Because D is acyclic, the subgraph induced by T_i is acyclic for each $i = 1, 2, \ldots, t$. Hence each T_i contains a vertex x_i of out-degree zero in $D\langle T_i \rangle$. Thus $N^+(x_i) \subseteq N^+(T_i)$ and hence $k - d^+(x_i) \geq k - |N^+(T_i)| \geq k - h(T_i, H_i)$ for each $i = 1, 2, \ldots, t$. Now we obtain

$$\begin{aligned} a_k(D) &\geq \eta_k{}^+(D) \\ &\geq \sum_{i=1}^{t} (k - d^+(x_i)) \\ &\geq \sum_{i=1}^{t} (k - h(T_i, H_i)) \\ &\geq a_k(D), \end{aligned}$$

showing that $a_k(D) = \eta_k{}^+(D)$. □

Bang-Jensen made the following conjecture at a meeting in Budapest in 1994:

Conjecture 7.7.13 *For every semicomplete digraph D on at least $k+1$ vertices*

$$a_k(D) \leq \frac{k(k+1)}{2}.$$

If true this would be the best possible since a transitive tournament T on $n \geq k+1$ vertices needs this many arcs. To see this it suffices to observe that, if $v_1, v_2, \ldots, v_n$ is the unique acyclic ordering of the vertices in T, then the first k vertices need $k, k-1, \ldots, 2, 1$ new arcs entering them in order to satisfy the condition that the in-degree is at least k. It is not difficult to check

(Exercise 7.20) that one can always make a transitive tournament k-strong by adding $\frac{k(k+1)}{2}$ new arcs. The following partial result follows from the work of Frank and Jordán [273]:

Proposition 7.7.14 *For every semicomplete digraph D on at least $k+1$ vertices we have $a_k(D) \leq k^2$.*

Proof: We prove this by showing that if D is an r-strong semicomplete digraph which has at least $r+2$ vertices, then we need at most $2r+1$ new arcs to make it $(r+1)$-strong. This will imply that we need at most k^2 arcs to make any semicomplete digraph k-strong.

Suppose first that D is not strongly connected. Since every semicomplete digraph has a Hamilton path (by Theorem 1.4.5), it follows that we can make D strong by adding one arc.

Suppose now that $r \geq 1$ and that there is some r-strong semicomplete digraph D for which we need at least $2r+2$ arcs to obtain an $(r+1)$-strong semicomplete digraph from D. Thus $a_{r+1}(D) \geq 2r+2$ and then we conclude from Theorem 7.7.7 that $a_{r+1}(D) = \gamma^*_{r+1}(D)$. Hence, by the definition of $\gamma^*_{r+1}(D)$, there exist $2r+2$ pairwise disjoint sets $X_1, X_2, \ldots, X_{2r+2}$, such that either each of these has $|N^+(X_i)| = r$ or each has $|N^-(X_i)| = r$. By considering the converse of D if necessary, we may assume that $|N^+(X_i)| = r$ for each X_i. Let X' be obtained by taking one vertex x_i from each X_i and let $D' = D\langle X' \rangle$. Since D' is semicomplete and has $2r+2$ vertices, it is easy to see that some x_i has at least $r+1$ out-neighbours in D'. However each of these contributes to $|N^+_D(X_i)|$, contradiction. $\square$

7.7.4 Splittings Preserving k-Strong Connectivity

In Section 7.5 we saw that, with respect to arc-strong connectivity, it is always possible to split off all arcs incident to a vertex v without decreasing the arc-strong connectivity of the resulting directed multigraph provided that $d^+(v) = d^-(v)$. To see that this does not extend to vertex-strong connectivity, consider the digraph D in Figure 7.8. If we add a new vertex s and arcs ds, es, sa, sf, then we obtain a 2-strong digraph D'. However, it follows from the fact that $a_2(D) = 3$ (as we argued previously, see Figure 7.8) that there cannot exist a complete splitting off at s in D' such that the resulting digraph (after removing s) is 2-strong.

Below we prove a splitting result for vertex-strong connectivity, due to Frank and Jordán. We do this to illustrate some of the proof techniques that can be used in this area. The reader will notice that they are different from the arc-strong connectivity proofs, although they do have common ingredients.

An arc a in a k-strong digraph D is ***k*-critical** if it cannot be deleted without destroying the property of D being k-strong. Note that if an arc is k-critical then it enters a set X with $|N^-_D(X)| = k$ and $|N^-_{D-a}(X)| = k-1$ and leaves a set Y with $|N^+_D(Y)| = k$ and $|N^+_{D-a}(Y)| = k-1$.

A subset $U \subset V$ in a k-strong digraph $D = (V, A)$ is **out-tight** (**in-tight**) if $|V - U| \geq k + 1$ and $|N_D^+(U)| = k$ ($|N_D^-(U)| = k$).

Lemma 7.7.15 *[84] Let $D = (V, A)$ be a k-strong digraph and let $e = xy$ be a k-critical arc in D. Then there exists a unique minimal out-tight set K in $D - e$ and a unique minimal in-tight set B in $D - e$. There is no arc from K to B in $D - e$, and in addition, $(D - e) + f$ is k-strong for any arc $f = uv$ with $u \in K$ and $v \in B$.*

Proof: Since e is k-critical, $\kappa(D - e) = k - 1$. Suppose that there exist two different minimal out-tight sets K_1 and K_2 in $D - e$. Let $H_1 = V - K_1 - N_{D-e}^+(K_1)$ and $H_2 = V - K_2 - N_{D-e}^+(K_2)$. Then (K_1, H_1) and (K_2, H_2) are one-way pairs in $D - e$ with $h_{D-e}(K_i, H_i) = k - 1$, $i = 1, 2$. Since we can make $D - e$ k-strong by adding the arc e, these one-way pairs cannot be independent. This implies that $x \in K_1 \cap K_2$ and $y \in H_1 \cap H_2$. Thus in $D - e$ we have $N_{D-e}^-(y) \subseteq V - K_1 \cup K_2$. Hence, by Menger's theorem, $|V - (K_1 \cup K_2)| \geq k$ and $|N^+(K_1 \cap K_2)|, |N^+(K_1 \cup K_2)| \geq k - 1$, since $D - e$ is $(k - 1)$-strong. Thus, using Proposition 7.1.3 and the fact that $D - e$ is $(k - 1)$-strong,

$$\begin{aligned} k - 1 + k - 1 &= |N_{D-e}^+(K_1)| + |N_{D-e}^+(K_2)| \\ &\geq |N_{D-e}^+(K_1 \cap K_2)| + |N_{D-e}^+(K_1 \cup K_2)| \\ &\geq k - 1 + k - 1. \end{aligned}$$

This gives $|N^+(K_1 \cap K_2)| = k - 1$, contradicting the minimality of K_1. The uniqueness of B follows similarly.

To see the second statement, observe that for any out-tight set L and the unique minimal out-tight set K we have $K \subseteq L$ and $B \subseteq (V - L - N^+(L))$. (In particular, $K \cap B = \emptyset$.) Hence, adding any arc from K to B will eliminate all one-way pairs (X, Y) with $h(X, Y) = k - 1$. $\square$

The following splitting result for vertex-strong connectivity is due to Frank and Jordán:

Theorem 7.7.16 *[271] Let $D = (V + s, A \cup F)$ be a k-strong digraph for which $|N^+(s)| = |N^-(s)| = d \geq 2k - 1$ holds and every arc e incident with s is k-critical. Then the arcs incident to s can be split off completely such that the resulting digraph D' obtained by deleting s is k-strong.* $\square$

Proof: If $k = 1$, then $d_D^-(s) = d_D^+(s)$, since D has no multiple arcs, and the claim follows from Theorem 7.5.2. Hence we may assume that $k \geq 2$.

Let $N^-(s) = \{u_1, \ldots, u_d\}$ and $N^+(s) = \{v_1, \ldots, v_d\}$. Since each arc incident with s is k-critical, it follows from Lemma 7.7.15 that there exist unique out-tight sets $O_1, O_2, \ldots O_d$ and unique in-tight sets $I_1, I_2, \ldots, I_d$ such that $u_i \in O_i$ and O_i is the unique minimal out-tight set in $D - u_is$, respectively, $v_i \in I_i$ and I_i is the unique minimal in-tight set in $D - sv_i$, for $i = 1, 2, \ldots, d$. We claim that $O_i \cap O_j = \emptyset$ for $1 \leq i < j \leq d$ and $I_i \cap I_j = \emptyset$ for $1 \leq i < j \leq d$.

Suppose this is not true. Then without loss of generality $O_i \cap O_j \neq \emptyset$ for some $i \neq j$. Note that $u_r s$ is the only arc from O_r to s in D for $r = 1, 2, \ldots, d$, since O_r has only $k-1$ out-neighbours in $D - u_r s$. Hence it follows that $u_i \in O_i - O_j$ and $u_j \in O_j - O_i$ and $O_i \cap O_j$ has no arc to s. Since $|N^-(s)| \geq 2k - 1 > k$ (because $k \geq 2$), we have $|V - (O_i \cup O_j)| \geq k - 1$. This and Menger's theorem imply that $|N^+_{D-s}(O_i \cap O_j)|, |N^+_{D-s}(O_i \cup O_j)| \geq k - 1$. However, applying Proposition 7.1.3 to O_i, O_j in $D - s$ (which is $(k-1)$-strong) we conclude

$$\begin{aligned}(k-1) + (k-1) &= |N^+_{D-s}(O_i)| + |N^+_{D-s}(O_j)| \\ &\geq |N^+_{D-s}(O_i \cap O_j)| + |N^+_{D-s}(O_i \cup O_j)| \\ &\geq (k-1) + (k-1).\end{aligned} \tag{7.21}$$

It follows from (7.21) that $|N^+_{D-s}(O_i \cap O_j)| = k - 1$ and since $O_i \cap O_j$ has no arc to s we get the contradiction $|N^+_D(O_i \cap O_j)| = k - 1$. Thus we have shown that $O_1, O_2, \ldots O_d$ are disjoint and similarly $I_1, I_2, \ldots, I_d$ are all disjoint.

This implies that $\gamma^*_k(D - s) \geq d = 2k - 1$ and hence, by Theorem 7.7.7 $a_k(D - s) = \gamma^*_k(D - s)$. Since D is k-strong it follows that $\gamma^*_k(D - s)$ cannot be greater than d, since the d arcs to and from s eliminate all sets with fewer than k neighbours. Thus $a_k(D - s) = d$. It remains to prove that we can make $D - s$ k-strong by adding a set of d arcs which form a pairing of $\{u_1, \ldots, u_d\}$ with $\{v_1, \ldots, v_d\}$.

Let F be any optimal augmenting set consisting of d arcs so that adding these arcs to $D - s$ results in a k-strong digraph D^*. Then F must contain exactly one arc whose tail is in O_i and exactly one arc whose head is in I_i, $i = 1, 2, \ldots, d$, since $O_1, O_2, \ldots O_d$ are disjoint and $I_1, I_2, \ldots, I_d$ are disjoint. This gives a pairing $(O_1, I_{\pi(1)}), \ldots, (O_d, I_{\pi(d)})$, where π is a permutation of $\{1, 2, \ldots, d\}$. Note that the set $\tilde{O}_i = V - (O_i \cup N^+_{D-s}(O_i))$ is in-tight in $D - s$. Let e_i be the unique arc in F which has tail in O_i and head in $I_{\pi(i)}$. Then e_i must have its head in $\tilde{O}_i$ (because after adding F, $\tilde{O}_i$ has an in-neighbour in O_i). Then the minimality of $I_{\pi(i)}$ and Proposition 7.1.3 implies that $I_{\pi(i)} \subseteq \tilde{O}_i$.

Clearly the arc e_i is k-critical in D^*, since it is the only arc from F which leaves O_i. Thus, by Lemma 7.7.15, there is a unique minimal out-tight set O containing the tail of e_i and a unique minimal in-tight set I containing the head of e_i in $D^* - e_i$. We claim that $O = O_i$ and $I = I_{\pi(i)}$. Clearly O_i is out-tight in $D^* - e_i$, so $O \subseteq O_i$. If we do not have equality, then this would contradict the minimality of O_i in $D - u_i s$ (here we used that s has precisely one in-neighbour in O_i). Now it follows from Lemma 7.7.15 that we can replace the arc e_i by any arc from O_i to $I_{\pi(i)}$, in particular, the arc $u_i v_{\pi(i)}$, and still have an optimal augmenting set F'. This shows that we can replace the arcs in F one by one, until we get the optimal augmenting set $F^* = \{u_1 v_{\pi(1)}, \ldots, u_d v_{\pi(D)}\}$ and the proof is complete. $\square$

For further results on splittings that preserve vertex-strong connectivity the reader is referred to the papers [271, 272] by Frank and Jordán, the paper [467] by Jordán and Jordán's PhD thesis [468].

7.8 A Generalization of Arc-Strong Connectivity

Below we show how to reduce the vertex-strong connectivity augmentation to a generalization of the arc-strong connectivity augmentation problem.

Let $D = (V, A)$ be a directed multigraph with two specified (not necessarily disjoint) subsets S, T of vertices. We say that D is **k-(S,T)-arc-strong** if there are k arc-disjoint (s,t)-paths in D for every choice of $s \in S, t \in T$. Thus if $S = T = V$ this corresponds to D being k-arc-strong.

Recall that in the proof of Menger's theorem (Theorem 7.3.1) we reduced local vertex-strong connectivity to local arc-strong connectivity via the vertex-splitting technique (recall Figure 3.4). It follows from the proof of Theorem 7.3.1 that a digraph $D = (V, A)$ is k-strong if and only if D_{ST} is k-(S,T)-arc-strong, where $S = \{v_s : v \in V\}, T = \{v_t : v \in V\}$. Two subsets X, Y are **(S,T)-independent** if $X \cap Y \cap T = \emptyset$, or $S \subset X \cup Y$. A family $\mathcal{F}$ of subsets of V is **(S,T)-independent** if the sets in $\mathcal{F}$ are pairwise (S,T)-independent. A set $X \subset V$ is **essential** if $X \cap T \neq \emptyset$ and $S - X \neq \emptyset$.

Frank and Jordán [272] characterized the size of a minimum cardinality set of new arcs to add to a digraph $D = (V, A)$ with specified subsets $S, T \subseteq V$ in order to make the resulting digraph k-(S,T)-arc-strong.

Theorem 7.8.1 *[272] Let $D = (V, A)$ be a digraph with a pair of non-empty (not necessarily distinct) subsets $S, T \subseteq V$. Then D can be made k-(S,T)-arc-strong by adding at most γ arcs with tails in S and heads in T if and only if*

$$\sum_{Z \in \mathcal{H}} (k - d^-(Z)) \leq \gamma \tag{7.22}$$

holds for every (S,T)-independent family $\mathcal{H}$ of essential subsets of V. □

To see that we really need to consider deficiencies of (S,T)-independent families (and not just a kind of subpartition lower bound), consider the digraph with four vertices $\{s_1, s_2, t_1, t_2\}$ and no arcs. If we take $k = 1$, then it is easy to see that, if we can add arcs from S to T only, we need four arcs s_it_j, $i, j = 1, 2$ to obtain a 1-(S,T)-arc-strong digraph. The only (S,T)-independent family with four members is $\{\{s_i, t_j\} : i, j = 1, 2\}$.

So far no combinatorial polynomial algorithm is known for the (S,T)-arc-strong connectivity augmentation problem for general k. For $k = 1$ Enni described such an algorithm in [219].

Theorem 7.8.1 is not only a generalization of the arc-strong connectivity augmentation result in Theorem 7.6.3 (and hence implies Theorem 7.6.3 as can be verified by solving Exercise 7.39). Theorem 7.8.1 also implies Theorem 7.7.3 as we shall see below.

Proof of Theorem 7.7.3 [273]:

Let $D = (V, A)$ be a digraph with $\kappa(D) < k$ which we want to make k-strong. We first construct the digraph $D' = (S \cup T, A')$ by the vertex splitting procedure (splitting each v into v_s, v_t, see Figure 3.4). By the remark in the beginning of this section D' is l-(S,T)-arc-strong if and only if D is l-strong. Let $\gamma_{k,S,T}(D')$ denote the **k-(S,T)-arc-strong connectivity augmentation number of** D, that is, the minimum number of new arcs with tails in S and heads in T, one has to add to D_{ST} in order to make it k-(S,T)-arc-strong. Furthermore let $\eta_k(D)$ ($\eta_{k,S,T}(D')$) denote the maximum deficiency, with respect to k, over all independent families of one-way pairs in D (respectively, (S,T)-independent families of essential sets in D').

From the construction of D' and the proof of Theorem 7.3.1, it follows easily that, if F is a new set of arcs all with tails in S and heads in T such that adding F to D' makes the resulting digraph k-(S,T)-arc-strong, then the corresponding set of arcs added to D will result in a k-strong digraph. Hence we have

$$a_k(D) \leq \gamma_{k,S,T}(D'). \tag{7.23}$$

Below we will demonstrate that $\eta_k(D) \geq \eta_{k,S,T}(D')$. We show that there is some family $\mathcal{F}'$ of (S,T)-independent essential sets with deficiency $\eta_{k,S,T}(\mathcal{F}') = \eta_{k,S,T}(D')$ from which we can construct an independent family $\mathcal{F}$ of one-way pairs in D with $\eta_k(\mathcal{F}) = \eta_{k,S,T}(\mathcal{F}')$. For this choose $\mathcal{F}' = \{Z_1, \ldots, Z_r\}$ with deficiency $\eta_{k,S,T}(D')$ to satisfy the following property:

$$|\mathcal{F}'| \text{ is minimal and with respect to this} \sum_{j=1}^{r}(|T - Z_j| + |S \cap Z_j|) \text{ is maximal.} \tag{7.24}$$

Claim A: For every $Z_j \in \mathcal{F}'$ there is no arc from $S - Z_j$ to $T \cap Z_j$.

Proof of Claim A: Suppose there is some j with $1 \leq j \leq r$ for which there is an arc st from $S - Z_j$ to $T \cap Z_j$. If $|T \cap Z_j| \geq 2$, then replacing Z_j by $Z_j - t$ we obtain a new (S,T)-independent family $\mathcal{F}''$ of essential sets and since $d^+_{D'}(t) = 1$ it follows that the deficiency of $\mathcal{F}''$ is at least that of $\mathcal{F}'$. But now $\mathcal{F}''$ contradicts the choice of $\mathcal{F}'$ so as to satisfy (7.24). Hence $T \cap Z_j = \{t\}$. Since $v_t \in T$ dominates $v_s \in S$ for each $v \in V$, we have $|S - Z_j| \geq 2$, (as otherwise $d^-(Z_j) \geq |T| - 1 = |V| - 1 \geq k$ and we could have deleted Z_j from $\mathcal{F}'$ without decreasing the deficiency, contradicting (7.24)). Now replace Z_j by $Z_j \cup \{s\}$ in $\mathcal{F}'$. The new family $\mathcal{F}^*$ still consists of essential sets and has at least the same deficiency. This contradiction to (7.24) completes the proof of the claim. $\square$

Note that by Claim A,

$$d^-_{D'}(Z_j) = |\{v_t : v_t \notin Z_j \text{ and } v_s \in Z_j\}|. \tag{7.25}$$

Claim B: For every $Z_j \in \mathcal{F}'$ there is some $v \in V$ so that both v_s and v_t belong to Z_j.

Proof Claim B: Suppose some Z_j does not satisfy this property. Choose $v_s \notin Z_j$ so that $v_t \in Z_j$. If $S - Z_j \neq \{v_s\}$, then replace Z_j by $Z_j \cup \{s\}$ in $\mathcal{F}'$. The new family $\mathcal{F}^*$ still consists of essential sets and has at least the same deficiency. This contradicts (7.24). Hence we may assume that $S - Z_j = \{v_s\}$. By the assumption that Z_j does not contain any pair v_s, v_t, we get that $T \cap Z_j = \{v_t\}$ and as above we see that Z_j can be deleted from $\mathcal{F}'$, contradicting (7.24). □

Now we can finish the proof of Theorem 7.7.3: Let

$$X_j = \{v \in V : v_s \notin Z_j\}, Y_j = \{v \in V : v_s \in Z_j \text{ and } v_t \in Z_j\}, 1 \leq j \leq r.$$

It follows from the fact that each Z_j is essential and Claim B that $X_j, Y_j \neq \emptyset$. Furthermore, by Claim A, (X_j, Y_j) is a one-way pair and, by (7.25), it has deficiency $k - d^-_{D'}(Z_j)$. Let

$$\mathcal{F} = \{(X_1, Y_1), \ldots, (X_r, Y_r)\}.$$

Since $\mathcal{F}'$ is (S,T)-independent, $\mathcal{F}$ consists of independent one-way pairs and by the remark above, the deficiency of $\mathcal{F}$ equals $\eta_{k,S,T}(D')$. This shows that $\eta_k(D) \geq \eta_{k,S,T}(D')$. Combining this with (7.23), we get

$$\eta_{k,S,T}(D') \leq \eta_k(D) \leq a_k(D) \leq \gamma_{k,S,T}(D').$$

By Theorem 7.8.1 equality holds everywhere and Theorem 7.7.3 follows. □

7.9 Arc Reversals and Vertex-Strong Connectivity

Suppose now that we want to increase the vertex-strong connectivity of a digraph by re-orienting arcs rather than adding new ones. This gives rise to the following problem.

Problem 7.9.1 *Given natural number k and a digraph $D = (V, A)$ on at least $k+1$ vertices, find a minimum set $F \subset A$ of arcs in D such that the digraph D' obtained from D by reversing every arc in F is k-strong.*

If such a subset exists, then we let $r_k(D) = |F|$, where F is a minimum cardinality subset of A, whose reversal makes the resulting digraph k-strong. Otherwise we let $r_k(D) = \infty$.

For arbitrary digraphs it is not clear how we can decide whether such a reversal even exists, let alone find an optimal one (unless we try all possibilities which clearly requires exponential time). Indeed, this seems to be a very difficult problem (see also Conjecture 8.6.7). Clearly, if $r_k(D) < \infty$, then we have

$$a_k(D) \le r_k(D), \tag{7.26}$$

since, instead of reversing in D we may add exactly those new arcs we would obtain by reversing and keep the original ones.

We will now show that for semicomplete digraphs D, the function $r_k(D)$ behaves nicely.

Lemma 7.9.2 *[84] If a semicomplete digraph D has at least $2k+1$ vertices then $r_k(D)$ is finite and is bounded by a function depending only on k.*

Proof: To see this it suffices to use the following two simple observations, the proofs of which are left to the reader as Exercises 7.26 and 7.36.

(a) If D is a k-strong digraph and D' is obtained from D by adding a new vertex x and arcs from x to every vertex in a set X of k distinct vertices of D and arcs from every vertex of a set Y of k distinct vertices of D to x, then D' is also k-strong.
(b) If T is a semicomplete digraph on at least $4k-1$ vertices, then T contains a vertex with in-degree and out-degree at least k.

By observations (a) and (b), for every semicomplete digraph T, $r_k(T) \le r_k(T')$ for some subgraph T' of T with $|V(T')| \le 4k-2$. Continue removing vertices as long as we can find a vertex of in- and out-degree at least k, or the current graph has $2k+1$ vertices. When this process stops we have $2k+1 \le |V(T')| \le 4k-2$ in the current semicomplete digraph T'. Then we can make T' k-strong by reversing some arcs and add back each of the removed vertices in the reverse order of the deletion. This provides a simple upper bound for $r_k(T)$ (and hence for $a_k(T)$) as a function of k: we need to reverse at most $\frac{(4k-2)(4k-3)}{4}$ arcs. $\square$

Note that the process above may not lead to an optimal reversal for the original semicomplete digraph(in terms of the number of arcs to reverse), not even if we reverse optimally in T' (see also Exercise 7.40). Bang-Jensen and Jordán showed that, somewhat surprisingly, as soon as the number of vertices in the given semicomplete digraph D is sufficiently high (depending only on k), the minimum number of arcs in D we need to reverse in order to achieve a k-strong semicomplete digraph equals the minimum number of new arcs we need to add to D to obtain a k-strong semicomplete digraph.

Theorem 7.9.3 *[84] If D is a semicomplete digraph on at least $3k-1$ vertices for some $k \ge 2$ then $a_k(D) = r_k(D)$.*

The idea is to show that $r_k(D) \le a_k(D)$, by demonstrating that a certain optimal augmenting set F of D has the property that, if we reverse the existing (opposite) arcs of F in D, then we obtain a k-strong semicomplete digraph. As we point out later, even for semicomplete digraphs, it is by no means the case that just an arbitrary optimal augmenting set will have this property. It was shown in [84] that $3k-1$ is the best possible for semicomplete digraphs. However, in the case when D is tournament, the question as to whether or not the bound is best possible was left open and the following conjecture was implicitly formulated.

Conjecture 7.9.4 *[84] For every tournament D on at least $2k+1$ vertices, we have $a_k(D) = r_k(D)$.*

One may argue that perhaps if we restrict ourselves to only adding arcs between adjacent vertices, then we could have $a_k(D) = r_k(D)$ for arbitrary digraphs D, provided both numbers are finite and the number of vertices in D is large enough. This is not true, however, as can be seen from the following example:

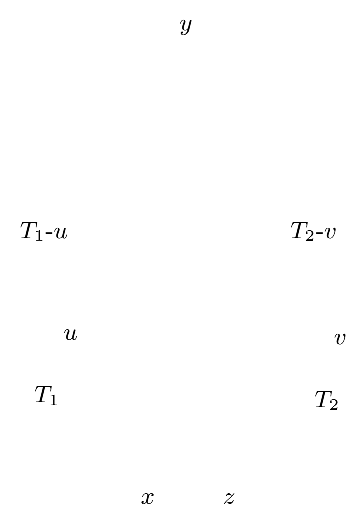


Figure 7.9 A digraph with $a_2(D) = 1$ and $r_2(D) = 2$. The digraphs T_1 and T_2 are 2-strong. Fat arcs between sets of vertices indicate that all arcs between these sets are present and have the direction shown.

Let T_1 and T_2 be disjoint 2-strong digraphs, let $u \in V(T_1)$, $v \in V(T_2)$ be fixed vertices and let D be the digraph obtained from T_1 and T_2 by adding new vertices x, y, z and the following arcs (see Figure 7.9):

$$\{r{\rightarrow}y : r \in V(T_1)\} \cup \{y{\rightarrow}s : s \in V(T_2)\} \cup \{s{\rightarrow}r : s \in V(T_2), r \in V(T_1)\} \cup$$
$$\{r{\rightarrow}x : r \in V(T_1) - u\} \cup \{s{\rightarrow}z : s \in V(T_2) - v\} \cup \{x{\rightarrow}u, z{\rightarrow}v, z{\rightarrow}x\}$$

It is not difficult to see that $a_2(D) = 1$ and that any arc whose addition to D results in a 2-strong digraph has tail x and head in $T_2 \cup z$. On the other hand it is also easy to see that $r_2(D) \geq 2$ (Exercise 7.34). This example can be modified to work for any $k \geq 1$ (Exercise 7.35).

If we add arcs to the digraph D described above without increasing the number of out-neighbours of x and of z, we can construct a semicomplete digraph D' of any given size for which xz is an optimal augmentation but reversing xz does not make D' 2-strong.This – and similar constructions for higher connectivity – show that even for semicomplete digraphs we cannot reverse along an arbitrary optimal augmenting set for $k \geq 2$.

The following conjecture which is stronger than Conjecture 7.7.13 was made by Bang-Jensen at a meeting in Budapest in 1994. Again the transitive tournament on $n \geq 2k+1$ vertices shows that the bound would be best possible if true.

Conjecture 7.9.5 *For every tournament T on n vertices and every positive integer k such that $n \geq 2k+1$ we have $r_k(T) \leq \frac{k(k+1)}{2}$.*

The problem of determining the optimal number of arcs to be reversed to make an arbitrary digraph k-arc-strong was shown by Frank to be polynomially solvable in [254]. We will return to this in Section 8.8.4, where we shall see how to solve this problem using submodular flows.

We complete this section with the following useful observation, which we use in Section 7.14.

Lemma 7.9.6 *[44, 344] Let $D = (V, A)$ be a k-strong digraph and let xy be an arc of D. If D has at least $(k+1)$-internally disjoint (x, y)-paths each of length at least 2, then the digraph D' obtained from D by replacing the arc xy by the arc yx (or just deleting xy if $yx \in A$) is k-strong. Furthermore, if D' is not $(k+1)$-strong, then every minimum separating set S' of D' is also separating in D.*

Proof: Suppose that D' is not $(k+1)$-strong. Let S' be a minimum separator of D'. Then $|S'| \leq k$ and there is some pair a, b of vertices separated by S' in D'. It follows from the assumption on $\kappa(x, y)$ that either $S' \cap \{x, y\} \neq \emptyset$, or S' does not separate x, y. From this we get that $\{a, b\} \neq \{x, y\}$ and that a, b are also separated by S' in D. This shows that every minimum separating set of D' is also separating in D. Since D is k-strong we have $|S'| = k$ and hence D' is k-strong. $\square$

7.10 Minimally k-(Arc)-Strong Directed Multigraphs

A directed multigraph $D = (V, A)$ is **minimally k-(arc)-strong** if D is k-(arc)-strong, but for every arc $e \in A$, $D - e$ is not k-(arc)-strong. From an application point of view it is very important to be able to identify a small subgraph of a k-(arc)-strong directed multigraph which is spanning and still k-(arc)-strong. The reason for this could be as follows. If many arcs of D are redundant, then it may make sense to discard these. If one is writing an algorithm for finding a certain structure that is based on k-(arc)-strong connectivity, then working with the smaller subgraph could speed up the algorithm, especially if k is relatively small compared to n.

Note however that, if we are given a k-(arc)-strong directed multigraph $D = (V, A)$ and ask for the smallest number of arcs in a spanning k-(arc)-strong subgraph of D, then this is an $\mathcal{NP}$-hard problem. Indeed, a strong digraph D on n vertices has a strong spanning subgraph on n arcs if and only if D has a hamiltonian cycle. Hence, we must settle for finding spanning subgraphs with relatively few arcs. Since every k-arc-strong directed multigraph on n vertices has at least kn arcs, the proof of Theorem 7.10.1 together with Exercise 9.27 implies that there is a polynomial algorithm to find a spanning k-arc-strong subgraph with no more than twice the optimum number of arcs. We discuss this topic in more detail in Section 7.16.

7.10.1 Minimally k-Arc-Strong Directed Multigraphs

We present some important degree results by Mader [535]. Combining these results with Theorem 7.5.2 we obtain a construction method (also due to Mader) to generate all k-arc-strong directed multigraphs. We start with a result by Dalmazzo which gives an upper bound on the number of arcs in any minimally k-arc-strong directed multigraph of order n.

Theorem 7.10.1 *[172] A minimally k-arc-strong directed multigraph has at most $2k(n-1)$ arcs and this is the best possible.*

Proof: Let $D = (V, A)$ be k-arc-strong and let s be a fixed vertex of V. By Theorem 7.3.2 $d^+(U), d^-(U) \geq k$ for every $\emptyset \neq U \subset V$. Hence, by Edmonds' branching theorem (Theorem 9.5.1), D contains k-arc-disjoint in-branchings $F^-_{s,1}, \ldots, F^-_{s,k}$ rooted at s and k arc-disjoint out-branchings $F^+_{s,1}, \ldots, F^+_{s,k}$ rooted at s. Let $A' = A(F^-_{s,1}) \cup \ldots \cup A(F^-_{s,k}) \cup A(F^+_{s,1}) \cup \ldots \cup A(F^+_{s,k})$ and let $D' = (V, A')$. Then D' is k-arc-strong and has at most $2k(n-1)$ arcs. Thus if D is minimally k-arc-strong, then $A = A'$. To see that this bound cannot be sharpened it suffices to consider the directed multigraph obtained from a tree T (as an undirected graph) and replacing each edge uv of T by k arcs from u to v and k arcs from v to u. $\square$

It it easy to see that, if $D = (V, A)$ is minimally k-arc-strong, then every arc uv leaves a k-out-critical set[4] and enters a k-in-critical set. Applying (7.2) we obtain Lemma 7.10.2 below which implies that every arc uv leaves

[4] Recall that this means that there is some $X \subset V$ such that $u \in X, v \in V - X$ and $d^+(X) = k$.

precisely one minimal k-out-critical set X_u and enters precisely one minimal k-in-critical set Y_u. Here minimal means with respect to inclusion.

Lemma 7.10.2 *If X, Y are crossing k-in-critical sets in D, then $X \cap Y$ and $X \cup Y$ are also k-in-critical sets and $d(X, Y) = 0$.*

Proof: Suppose X, Y are crossing and k-in-critical. Using (7.2) we get

$$\begin{aligned} k + k &= d^-(X) + d^-(Y) \\ &= d^-(X \cup Y) + d^-(X \cap Y) + d(X, Y) \\ &\geq k + k, \end{aligned}$$

implying that $X \cap Y$ and $X \cap Y$ are both k-in-critical and $d(X, Y) = 0$. □

Intuitively, Lemma 7.10.2 implies that minimally k-arc-strong directed multigraphs have vertices of small in-degree and vertices small out-degree. The next result by Mader shows that this is indeed the case. In fact, a much stronger statement holds.

Theorem 7.10.3 *[535] Every minimally k-arc-strong directed multigraph has at least two vertices x, y with $d^+(x) = d^-(x) = d^+(y) = d^-(y) = k$.*

Proof: We give a proof due to Frank [260]. Let $\mathcal{R}$ be a family of k-in-critical sets with the property that

$$\text{every arc in } D \text{ enters at least one member of } \mathcal{R}. \tag{7.27}$$

By our remark above such a family exists since D is minimally k-arc-strong.

Our first goal is to make $\mathcal{R}$ cross-free (that is, we want to replace $\mathcal{R}$ by a new family $\mathcal{R}^*$ of k-in-critical sets such that $\mathcal{R}^*$ still satisfies (7.27) and no two members of $\mathcal{R}^*$ are crossing). To do this we apply the so-called **uncrossing technique** which is quite useful in several proofs. If there are crossing members X, Y in $\mathcal{R}$, then by Lemma 7.10.2, $X \cap Y$ and $X \cup Y$ are k-in-critical and $d(X, Y) = 0$. Hence every arc entering X or Y also enters $X \cup Y$, or $X \cap Y$. Thus we can replace the sets X, Y by $X \cap Y, X \cup Y$ in $\mathcal{R}$ (we only add sets if they are not already there). Since $|X \cap Y|^2 + |X \cup Y|^2 > |X|^2 + |Y|^2$ and the number of sets in R does not increase, we will end up with a family $\mathcal{R}$ which is cross-free. Note that we could have obtained such a family directly by choosing the members in $\mathcal{R}$ as the unique minimal k-in-critical sets entered by the arcs of A. However, this choice would make the proof more complicated, since we lose the freedom of just working with a cross-free family satisfying (7.27). We shall use this freedom in Case 2 below. Assume below that

$$\mathcal{R} \text{ is cross-free.} \tag{7.28}$$

Now the trick is to consider an arbitrary fixed vertex s and show that $V - s$ contains a vertex with in-degree and out-degree k. This will imply the theorem.

Let s be fixed and define the families $\mathcal{S}$ and $\mathcal{U}$ as follows

$$\mathcal{S} = \{X \in \mathcal{R} : s \notin X\}, \;\; \mathcal{U} = \{V - X : s \in X \in \mathcal{R}\}. \tag{7.29}$$

Let $\mathcal{L} = \mathcal{L}(\mathcal{R}) = \mathcal{S} \cup \mathcal{U}$.

Claim A: The family $\mathcal{L}$ is laminar.

Proof of Claim A: We must show that no two members of $\mathcal{L}$ are intersecting. Suppose $X, Y \in \mathcal{L}$ are intersecting. Then X and Y cannot both be from $\mathcal{S}$ since then they are crossing and this contradicts (7.28). Similarly X and Y cannot both be from $\mathcal{U}$, since then $V - X, V - Y$ are crossing members of $\mathcal{R}$, a contradiction again. Finally, if $X \in \mathcal{S}$ and $Y \in \mathcal{U}$, then X and $V - Y$ are crossing members of $\mathcal{R}$, contradicting (7.28). This proves that $\mathcal{L}$ is laminar. □

By the choice of $\mathcal{S}$ and $\mathcal{U}$ we have the following property:

$$\text{Every arc either enters a member of } \mathcal{S} \text{ or leaves a member of } \mathcal{U} \text{ (or both).} \tag{7.30}$$

Suppose $\mathcal{R}$ is chosen such that (7.27) and (7.28) hold and furthermore

$$\sum_{X \in \mathcal{L}} |X| \text{ is minimal.} \tag{7.31}$$

To complete the proof of the theorem we consider two cases.

Case 1 Every member of $\mathcal{L}$ has size one:

Let $X = \{x \in V - s : \{x\} \in \mathcal{S}\}$ and $Y = \{y \in V - s : \{y\} \in \mathcal{U}\}$. Then X cannot be empty, since every arc leaving s enters X. Similarly Y is non-empty. Now if $X \cap Y = \emptyset$, then there can be no arc leaving X, by the definition of X and (7.30). However $d^+(X) \geq k$, since D is k-arc-strong and hence we have shown that $X \cap Y \neq \emptyset$. Let t be any element in $X \cap Y$, then we have $d^+(t) = d^-(t) = k$.

Case 2 Some member Z of $\mathcal{L}$ has size at least two:

Choose Z such that $|Z|$ is minimal among all members of $\mathcal{L}$ of size at least two.

Note that, if we consider the converse D^* of D and let $\mathcal{R}^* = \{V - X : X \in \mathcal{R}\}$ and then define $\mathcal{S}^*, \mathcal{U}^*$ as we defined $\mathcal{S}$ and $\mathcal{U}$ from $\mathcal{R}$, then $\mathcal{S}^* = \mathcal{U}$ and $\mathcal{U}^* = \mathcal{S}$. Furthermore, the corresponding family $\mathcal{L}^*$ satisfies (7.30) and (7.31). This shows that we may assume without loss of generality that $Z \in \mathcal{S}$. We claim that

$$\text{the directed multigraph } D\langle Z\rangle \text{ is strongly connected.} \tag{7.32}$$

Suppose this is not the case and let Z_1, Z_2 be a partition of Z with the property that there are no arcs from Z_2 to Z_1. Then we have $k \leq d^-(Z_1) \leq$

$d^-(Z) = k$, implying that Z_1 is k-in-critical and that every arc that enters Z also enters Z_1. Let $\mathcal{R}' = \mathcal{R} - \{Z\} + \{Z_1\}$, $\mathcal{S}' = \mathcal{S} - \{Z\} + \{Z_1\}$ and let $\mathcal{L}' = \mathcal{S}' \cup \mathcal{U}$. Then $\mathcal{L}'$ still satisfies (7.30) and

$$\sum_{X \in \mathcal{L}'} |X| < \sum_{X \in \mathcal{L}} |X|.$$

However, this contradicts the choice of $\mathcal{R}$. Thus we have shown that $D\langle Z\rangle$ is strongly connected. This establishes (7.32).

We return to the proof of the theorem. Let

$$A = \{z \in Z : \{z\} \in \mathcal{S}\}, B = \{z \in Z : \{z\} \in \mathcal{U}\}.$$

If $A \cap B \neq \emptyset$ then any vertex $t \in A \cap B$ has $d^+(t) = d^-(t)$ and we are done. Suppose $A \cap B = \emptyset$. Then we claim that

$$A = \emptyset. \tag{7.33}$$

Suppose $A \neq \emptyset$. By the choice of $\mathcal{R}$ so that $\mathcal{L}$ satisfies (7.31), we cannot leave out any set without violating (7.30). Hence we cannot have $A = Z$, because then we could leave out Z without violating (7.30). Now (7.32) implies that there is an arc uv from A to $Z - A$. Since $\mathcal{L}$ satisfies (7.30), the arc uv either enters some member of $\mathcal{S}$ or leaves a member of $\mathcal{U}$. If it enters a member M of $\mathcal{S}$, then by the definition of A, M cannot be of size one. On the other hand, by the fact that $\mathcal{L}$ is laminar and the minimality of Z, M also cannot have size at least two. Hence uv must leave a member W of $\mathcal{U}$. Since we have assumed $A \cap B = \emptyset$, this must be a set of size more than one. Using that $\mathcal{L}$ is laminar it follows that $W \subset Z$, contradicting the choice of Z. Hence we must have $A = \emptyset$ and (7.33) is established. Next we claim that

$$B = Z. \tag{7.34}$$

Since $A = \emptyset$ and Z is minimal among all members of $\mathcal{L}$ of size at least 2, every arc with both ends in Z must leave a member of B (using the same arguments as above). Hence $B \neq \emptyset$ and we must have $B = Z$, since otherwise (7.32) would imply the existence of an arc from $Z - B$ to B, contradicting what we just concluded.

Now we are ready to complete the proof of the theorem. Since $B = Z$, every vertex in Z has out-degree k. Thus we have

$$\begin{aligned} k|Z| &= \sum_{v \in Z} d^+(v) \\ &= d^+(Z) + |A(D\langle Z\rangle)| \\ &\geq k + |A(D\langle Z\rangle)| \end{aligned}$$

$$\begin{aligned}
&= k + (\sum_{v \in Z} d^-(v)) - d^-(Z) \\
&= \sum_{v \in Z} d^-(v) \\
&\geq k|Z|.
\end{aligned}$$

Hence equality holds everywhere, in particular, every vertex in Z has in- and out-degree k. $\square$

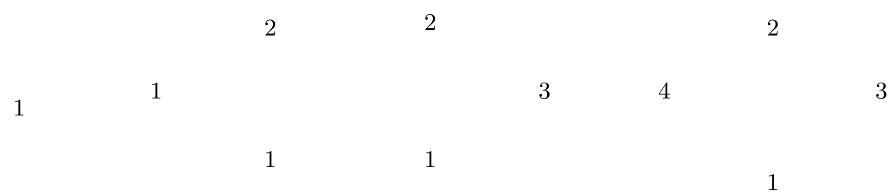


Figure 7.10 A construction of a 2-arc-strong directed multigraph starting from a single vertex.

Using Theorem 7.5.3 and Theorem 7.10.3 one can obtain the following complete characterization of k-arc-strong directed multigraphs, due to Mader [537].

Theorem 7.10.4 *[537] A directed multigraph D is k-arc-strong if and only if it can be obtained starting from a single vertex by applying the following two operations (in any order):*

Operation A: Add a new arc connecting existing vertices.
Operation B: Choose k distinct arcs $u_1v_1, \dots u_kv_k$ and replace these by $2k$ new arcs $u_1s, \dots, u_ks, sv_1, \dots, sv_k$, where s is a new vertex.

Proof: Clearly Operation A preserves the property of being k-arc-strong. To see that this also holds for Operation B we apply Menger's theorem. Suppose D is k-arc-strong and D' is obtained from D by one application of Operation B but D' is not k-arc-strong. Let $U \subset V(D')$ be some subset such that $d^+_{D'}(U) \leq k-1$. Then we must have $U \neq \{s\}$ and $U \neq V(D)$, since clearly s has in- and out-degree k in D'. Now it is easy to see that the corresponding set $U-s$ has out-degree less than k in D, a contradiction. From these observations it is easy to prove by induction on the number of vertices that every directed multigraph that can be constructed via operations A and B is k-arc-strong. Here we assume by definition that every directed pseudograph having just one vertex is k-arc-strong.

The other direction can be proved using induction on the number of arcs. If D is k-arc-strong and not minimally k-arc-strong, then we can remove an arc and apply induction. Otherwise it follows from Theorem 7.10.3 that D

contains a vertex s such that $d^+(s) = d^-(s) = k$. According to Theorem 7.5.3 this vertex and the $2k$ arcs incident with it can be replaced by k new arcs in such a way that the resulting directed multigraph D' is k-arc-strong. By induction D' can be constructed via operations A and B. Since we can go from D' back to D by using operation B once, D can be constructed using operations A and B. □

See Figure 7.10 for an illustration of the theorem.

7.10.2 Minimally k-Strong Digraphs

In this section $D = (V, A)$ is always a digraph (i.e. no multiple arcs) and hence we know that $d^+(v) = |N^+(v)|$ for each $v \in V$.

We saw in the last section that every minimally k-arc-strong directed multigraph has at least two vertices with in- and out-degree equal to k. Mader conjectures that this is also the case for vertex-strong connectivity in digraphs.

Conjecture 7.10.5 *[538] Every minimally k-strong digraph contains at least two vertices such that both have in- and out-degree k.*

This conjecture is still open and seems very difficult. For $k = 1$ the truth of Conjecture 7.10.5 follows from Theorem 7.10.3. Mader [541] has proved the conjecture for $k = 2$. For all other values of k the conjecture is open. Examples by Mader [535] show that one cannot replace two by three in the conjecture.

Recall (from Subsection 7.7.4) that an arc e of a k-strong digraph is k-critical if $D - e$ is not k-strong. By Lemma 7.7.2, for each k-critical arc uv we can associate sets T_{uv}, H_{uv} such that (T_{uv}, H_{uv}) is a one-way pair in $D - uv$ and $h(T_{uv}, H_{uv}) = k - 1$. This one-way pair may not be unique, but below we always assume that we have chosen a fixed one-way pair for each k-critical arc in D. Compare this with Lemma 7.7.15.

Lemma 7.10.6 *Let $D = (V, A)$ be a k-strong digraph. Then the following is true:*

(a) If D has two k-critical arcs ux, uy, such that $d^+(u) \geq k+1$, then $|T_{uy}| > |H_{ux}|$.
(b) If D has two k-critical arcs xu, yu, such that $d^-(u) \geq k+1$, then $|H_{xu}| > |T_{yu}|$.

Proof: Since (b) follows from (a) by considering the converse of D, it suffices to prove (a). Hence we assume that ux, uy are k-critical arcs of D and that $d^+(u) \geq k+1$. Let $(T_{ux}, H_{ux}), (T_{uy}, H_{uy})$ be the pairs associated with ux, uy above. Note that these are not one-way pairs in D, since there is a (unique) arc, namely ux (uy) which goes from T_{ux} (T_{uy}) to H_{ux} (H_{uy}). Let also $S_{ux} = V - (T_{ux} \cup H_{ux})$ and $S_{uy} = V - (T_{uy} \cup H_{uy})$. Then $|S_{ux}| = |S_{uy}| = k - 1$ and

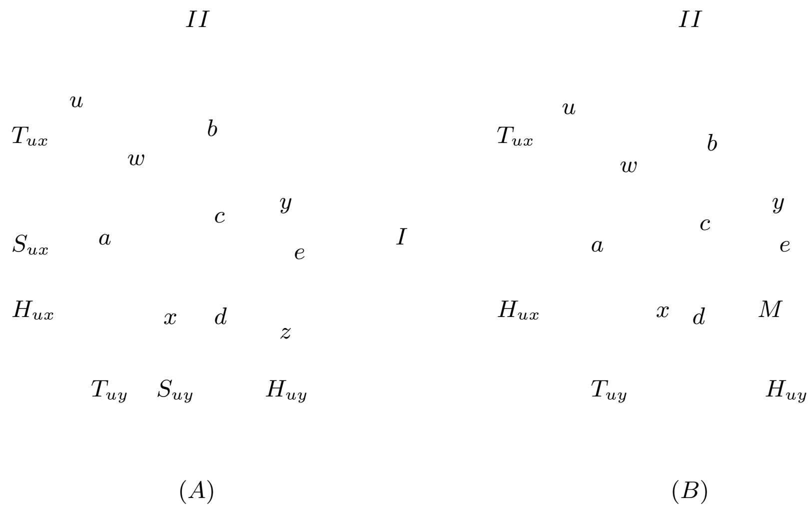


Figure 7.11 Illustration of the proof of Lemma 7.10.6. Part (A) illustrates the case when $H_{ux} \cap H_{uy} \neq \emptyset$. Part (B) illustrates the case when $H_{ux} \cap H_{uy} = \emptyset$. The first row of each 3×3 diagram corresponds to the set T_{ux}. The first column corresponds to T_{uy} and so on. The positions of x, y indicate that they can be in either of the two neighbouring cells. The numbers a, b, c, d, e denote the cardinality of the sets corresponding to their cell.

$x \in H_{ux} - H_{uy}, y \in H_{uy} - H_{ux}$. It will be useful to study Figure 7.11 while reading the proof.

Let a, b, c, d, e be defined as in Figure 7.11. Since each of the sets S_{ux}, S_{uy} has size $k-1$ we see that

$$a + b + 2c + d + e = 2k - 2. \tag{7.35}$$

We claim that $H_{ux} \cap H_{uy} = \emptyset$. Suppose this is not the case and let $z \in H_{ux} \cap H_{uy}$ be arbitrarily chosen. Now it follows from the fact that (T_{ux}, H_{ux}) is a one-way pair in $D - ux$ and (T_{uy}, H_{uy}) is a one-way pair in $D - uy$, that the set C_I, indicated by the line I in Figure 7.11, separates u from z in D. Hence $c + d + e \geq k$, since D is k-strong. Now (7.35) implies that the set C_{II}, indicated by the line II, has size at most $k-2$. Since $d^+(u) \geq k+1$ and u has precisely two arcs, namely ux, uy out of $T_{ux} \cap T_{uy}$ in $D - C_{II}$, we see that there is some out-neighbour w of u inside $T_{ux} \cap T_{uy}$. But now it is easy to see that $C_{II} \cup \{u\}$ separates w from z, contradicting that D is k-strong. Hence we have shown that $H_{ux} \cap H_{uy} = \emptyset$.

To complete the proof, we only need to show that $a \geq d$. Suppose this is not the case. Then in particular $d \geq 1$ and the size of the set C_{II} is at most $|S_{uy}| + a - d \leq k - 2$. Thus as above we can argue that u has an out-neighbour w inside $T_{ux} \cap T_{uy}$. Now $C_{II} \cup \{u\}$ separates w from x in D, a contradiction. □

An **anti-directed trail** is the digraph $\bar{T}$ one obtains from a closed undirected trail T of even length by fixing a traversal of T and orienting the edges so that every second vertex v has in-degree zero when we consider just the two arcs between v and its successor and predecessor on T. We denote the anti-directed trail $\bar{T}$ by $\bar{T} = v_1\bar{v}_1v_2\bar{v}_2 \ldots v_r\bar{v}_rv_1$, where $\bar{v}_i$ indicates that the vertex $\bar{v}_i$ is dominated by both its successor and its predecessor on the trail T. A vertex which dominates (is dominated by) both its successor and its predecessor on $\bar{T}$ is a **source** (**sink**) of $\bar{T}$. Note that if a vertex v is repeated on $\bar{T}$ then v may be both a source and a sink. An **anti-directed cycle** is an anti-directed trail in which no vertex occurs twice (that is, the underlying graph is just a cycle). See Figure 7.12 for an illustration of the definitions.

$\bar{v}_1$ v_1

$v_2 = \bar{v}_3$

v_3 $\bar{v}_2$

Figure 7.12 An anti-directed trail $v_1\bar{v}_1v_2\bar{v}_2v_3\bar{v}_3v_1$ on 6 vertices. The vertex $v_2 = \bar{v}_3$ is both a source and a sink of $\bar{T}$. Note that $\bar{T}$ contains no anti-directed cycle.

Now we can prove the following important result due to Mader:

Theorem 7.10.7 *[538] Let D be a k-strong digraph containing an anti-directed trail $\bar{T} = v_1\bar{v}_1v_2\bar{v}_2 \ldots v_r\bar{v}_rv_1$. Then at least one of the following holds:*

(a) Some arc $e \in A(\bar{T})$ is not k-critical in D.
(b) Some source v_i of $\bar{T}$ has out-degree k in D.
(c) Some sink $\bar{v}_j$ of $\bar{T}$ has in-degree k in D.

Proof: If (b) or (c) holds there is nothing to prove so suppose that $d^+(v_i) \geq k+1$ for each source and $d^-(\bar{v}_j) \geq k+1$ for each sink of $\bar{T}$. We shall prove that (a) holds.

Suppose to the contrary that every arc e on $\bar{T}$ is k-critical. Applying Lemma 7.10.6 (a) to the arcs $v_1\bar{v}_r, v_1\bar{v}_1$, we obtain $|T_{v_1\bar{v}_r}| > |H_{v_1\bar{v}_1}|$. Similarly, we get from Lemma 7.10.6 (b) that $|H_{v_1\bar{v}_1}| > |T_{v_2\bar{v}_1}|$. Repeating this argument around the trail we reach the following contradiction

$$|T_{v_1\bar{v}_r}| > |H_{v_1\bar{v}_1}| > |T_{v_2\bar{v}_1}| > |H_{v_2\bar{v}_2}| > \ldots > |H_{v_r\bar{v}_r}| > |T_{v_1\bar{v}_r}|.$$

Hence we have shown that (a) holds. □

The following is an easy consequence (see Exercise 7.48).

Corollary 7.10.8 *[538] Every minimally k-strong digraph contains a vertex x of in-degree k, or a vertex y of out-degree k.* □

Using Theorem 7.10.7, Mader proved the following much stronger statement.

Theorem 7.10.9 *[541] Every minimally k-strong digraph contains at least $k+1$ vertices of out-degree k and at least $k+1$ vertices of in-degree k.* □

Theorem 7.10.7 has many other nice consequences. Here is one for undirected graphs.

Corollary 7.10.10 *[533] Let C be a cycle of a k-connected undirected graph G. Then either C contains an edge e which can be removed without decreasing the connectivity of G, or some vertex $v \in V(C)$ has degree k in G.*

Proof: To see this, it suffices to consider the complete biorientation D of G and notice that $D - xy$ is k-strong if and only if $D - \{xy, yx\}$ is k-strong (Exercise 7.25) which happens if and only if $G - e$ is k-connected, where $e = xy$. Next, observe that in D, the cycle C either corresponds to one anti-directed trail C', obtained by alternating the orientation on the arcs taken twice around the cycle C, when $|C|$ is odd, or to two anti-directed cycles C', C'' when $|C|$ is even. Now the claim follows from Theorem 7.10.7. □

One reason why Corollary 7.10.10 is important is the following easy consequence concerning augmentations of undirected graphs, which was pointed out by Jordán.

Corollary 7.10.11 *[469] Let $G = (V, E)$ be an undirected graph which is k-connected, but not $(k+1)$-connected. Then every minimal set of edges F which augments the connectivity of G to $(k+1)$ induces a forest.* □

For directed graphs one obtains the following result, due to Jordán, on augmentations from k-strong to $(k+1)$-strong connectivity. Compare this with Theorem 7.7.8.

Corollary 7.10.12 *[467] Let $D = (V, A)$ be a directed graph which is k-strong, but not $(k+1)$-strong and let F be a minimal set of new arcs, whose addition to D gives a $(k+1)$-strong digraph. Then the digraph induced by the arcs in F contains no anti-directed trail.* □

One can also apply Theorem 7.10.7 to questions like: how many arcs can be deleted from a k-strong digraph, so that it still remains $(k-1)$-strong [540] (for undirected graphs see [122]). One easy consequence is the following.

Corollary 7.10.13 *[540] If $D = (V, A)$ is minimally k-strong and $D' = (V, A')$ is a spanning $(k-1)$-strong subgraph of D, then the difference $D_0 = (V, A - A')$ contains no anti-directed trail.*

Proof: Suppose $\bar{T} = v_1\bar{v}_1v_2\bar{v}_2\dots v_r\bar{v}_rv_1$ is an anti-directed trail in D_0. Since D is minimally k-strong, (a) cannot hold in Theorem 7.10.7. Suppose without loss of generality that (b) holds, then some source v_i has $d^+_D(v_i) = k$. However, since $d^+_{\bar{T}}(v_i) = 2$, this implies that $d^+_{D'}(v_i) = k-2$, contradicting the fact that D' is $(k-1)$-strong. □

Theorem 7.10.7 has many other important applications. We illustrate one such application in Section 7.16. We finish this section with a conjecture by Mader.

Conjecture 7.10.14 *[540] Every minimally k-strong digraph on n vertices contains at least $\frac{n-k}{k+1} + k$ vertices with out-degree equal to k.*

Mader has proved [540, page 437] that there are at least $\frac{1}{3}\sqrt{\frac{n}{k+1}}$ such vertices. For more on the topic see the very informative survey [540] by Mader.

7.11 Critically k-Strong Digraphs

In this section we always consider directed graphs (no multiple arcs). A vertex v of a digraph D is **critical** if $\kappa(D-v) < \kappa(D)$. The goal of this section is to illustrate some conditions under which we can always find a non-critical vertex in a digraph D. First observe that there can be no function $f(k)$ with the property that every k-strong digraph D with at least $f(k)$ vertices has a vertex v such that $D-v$ is still k-strong. This is not even the case for tournaments. To see this consider the example due to Thomassen (private communication, 1985) in Figure 7.13.

The example in Figure 7.13 can easily be generalized to arbitrary degrees of vertex-strong connectivity, by replacing each of the tournaments on seven vertices (right and left side of the figure) by the kth power of a $(2k+1)$-cycle and replacing the three long paths by k long paths starting at the top k vertices in the left copy and ending at the top k vertices in the right copy.

Below we discuss some results by Mader on sufficient conditions for a k-strong digraph to contain a non-critical vertex.

Definition 7.11.1 *Let D have $\kappa(D) = k$. A* **fragment** *in D is a subset $X \subset V$ with the property that either $|N^+(X)| = k$ and $X \cup N^+(X) \neq V$, or $|N^-(X)| = k$ and $X \cup N^-(X) \neq V$.*

Thus a fragment X corresponds to a one-way pair (X,Y) with $h(X,Y) = k$. Mader proved the following important result:

Theorem 7.11.2 *[539] Every critically k-strong digraph contains a fragment of size at most k.*

x y

Figure 7.13 A family $\mathcal{T}$ of 3-strong tournaments (the three paths from left to right can be arbitrary long). The big arc indicates that all arcs not explicitly shown go from right to left. It can be verified (Exercise 7.46) that each tournament in $\mathcal{T}$ is 3-strong and has the property that every vertex other than x, y is critical. Thus after removing at most two vertices we obtain a 3-strong tournament in which every vertex is critical.

This was conjectured by Hamidoune [394, Conjecture 4.8.3] who also conjectured the next two results, both of which are easy consequences of Theorem 7.11.2.

Corollary 7.11.3 *[539] Every critically k-strong digraph contains a vertex x with in-degree, or out-degree less than $2k$.*

Proof: Let $D = (V, A)$ be a critically k-strong digraph. By Theorem 7.11.2, D contains a fragment X with $|X| \le k$. By considering the converse of D if necessary, we may assume that $|N^+(X)| = k$. We prove that every vertex of X has out-degree at most $2k-1$. Let $x \in X$ be arbitrary. Note that every out-neighbour of x outside X contributes to $|N^+(X)|$, implying that there are at most k of these. Now the claim follows from the fact that $d^+_{D\langle X\rangle}(x) \le k - 1$. $\square$

We leave the proof of the next easy consequence as Exercise 7.41.

Corollary 7.11.4 *[539] Every critically k-strong oriented graph contains a vertex x with in-degree, or out-degree less than $\lfloor \frac{3k-1}{2} \rfloor$.* $\square$

7.12 Arc-Strong Connectivity and Minimum Degree

Let $D = (V, A)$ be a digraph and let $\delta(v) = \min\{d^+(v), d^-(v)\}$ for $v \in V$.

Obviously, the highest arc-strong connectivity a digraph can possibly have is $\delta^0(D)$. It is not easy to classify those digraphs for which the equality $\lambda(D) = \delta^0(D)$ actually holds. However, since we can calculate $\lambda(D)$ in polynomial time (see Subsection 7.4), it is easy to verify whether a given digraph D satisfies $\lambda(D) = \delta^0(D)$.

In this section we will give two sufficient conditions for this equality. The first result is due to Dankelmann and Volkmann.

Theorem 7.12.1 *[173] Let $D = (V, A)$ be a directed graph on n vertices without multiple arcs and let $v_1, v_2, \ldots, v_n$ be ordered so that $\delta(v_1) \geq \delta(v_2) \geq \ldots \geq \delta(v_n) = \delta^0(D)$. If $\delta^0(D) \geq \lfloor n/2 \rfloor$, or $\delta^0(D) < \lfloor n/2 \rfloor$ and there exists a k, $1 \leq k \leq \delta^0(D)$ such that*

$$\sum_{i=1}^{k}(\delta(v_i) + \delta(v_{n+i-\delta(D)-1})) \geq k(n-2) + 2\delta^0(D) - 1,$$

then $\lambda(D) = \delta^0(D)$.

Theorem 7.12.1 implies the following result by Xu which is a generalization of a result for undirected graphs in [329].

Corollary 7.12.2 *[742] Let D be a digraph on n vertices. If there are $\lfloor n/2 \rfloor$ disjoint pairs of vertices (v_i, w_i) with*

$$\delta(v_i) + \delta(w_i) \geq n \text{ for all } i = 1, 2, \ldots, \lfloor n/2 \rfloor,$$

then $\lambda(D) = \delta^0(D)$.

For further results on the relation between $\lambda(D)$ and $\delta^0(D)$ see [38, 173].

7.13 Connectivity Properties of Special Classes of Digraphs

In this section we describe a few results on the connectivity of various classes of digraphs introduced in Section 1.8 and Chapter 4. Some of these results will be used in other sections and chapters in this book.

The next lemma implies that almost all minimally k-strong decomposable digraphs are subdigraphs of extensions of digraphs.

Lemma 7.13.1 *[52] Let $D = F[S_1, S_2, \ldots, S_f]$ where F is a strong digraph on $f \geq 2$ vertices and each S_i is a digraph with n_i vertices and let $D_0 = F[\overline{K}_{n_1}, \overline{K}_{n_2}, \ldots, \overline{K}_{n_f}]$ be the digraph obtained from D by deleting every arc which lies inside some S_i (recall that $\overline{K}_{n_i}$ is the digraph on n_i vertices and no arcs). Let S be a minimal (with respect to inclusion) separating set of D_0. Then S is also a separating set of D, unless each of the following holds:*

(a) $S = V(S_1) \cup V(S_2) \ldots \cup V(S_f) \setminus V(S_i)$ for some $i \in \{1, 2, \ldots, f\}$, and
(b) $D\langle S_i \rangle$ is a strong digraph, and

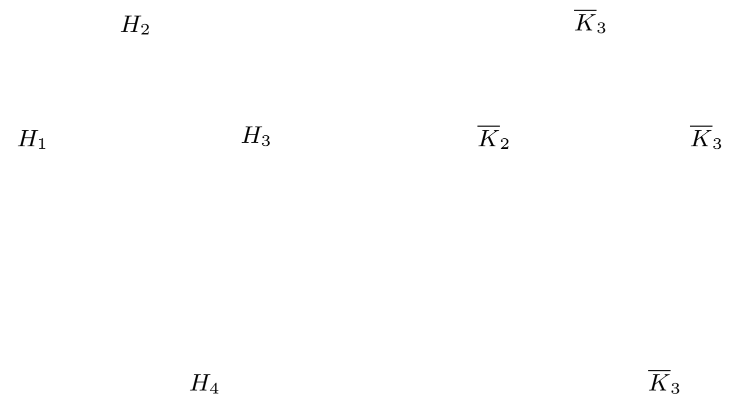


Figure 7.14 A 2-strong digraph D with decomposition $D = Q[H_1, H_2, H_3, H_4]$. Fat arcs indicate that all possible arcs are present and have the direction shown. The right figure shows the 2-strong digraph $D_0 = Q[\overline{K}_2, \overline{K}_3, \overline{K}_3, \overline{K}_3]$ obtained from D by deleting all arcs inside each H_i.

(c) $D = C_2[S, S_i]$.

In particular, if F has at least three vertices, then D is k-strong if and only if D_0 is k-strong.

Proof: Let S be a minimal separating set of D_0 and assume S is not separating in D. It is easy to see that, if x and y with $x, y \notin S$ belong to different S_i, then $D - S$ has an (x, y)-path if and only if $D_0 - S$ has such a path. Thus, since S is separating in D_0 but not in D, we must have $S = V(S_1) \cup V(S_2) \ldots \cup V(S_f) \setminus V(S_i)$ for some $i \in \{1, 2, \ldots, f\}$. Note that here we used the minimality of S to get that $S \cap S_j = \emptyset$ for some j. Now it follows trivially that $D\langle S_i \rangle$ must be a strong digraph, since $D - S$ is strong and the minimality of S implies that $D = C_2[S, S_i]$ (if some $S_j \subset S$ does not have arcs in both directions to S_i, then $S - S_j$ is also separating, contradicting the choice of S). $\square$

See Figure 7.14 for an example illustrating the lemma.

Combining Lemma 7.13.1 with Theorem 4.8.5 we obtain.

Corollary 7.13.2 *If D is a k-strong quasi-transitive digraph with decomposition $D = Q[W_1, \ldots, W_{|Q|}]$, then the digraph $D_0 = Q[\overline{K}_{|W_1|}, \ldots, \overline{K}_{|W_{|Q|}|}]$ (that is, the digraph obtained by deleting all arcs inside each W_i) is also k-strong.* $\square$

Another easy consequence of Lemma 7.13.1 is the following result by Bang-Jensen, Gutin and Yeo:

Lemma 7.13.3 *[70] Suppose that D is a digraph which can be decomposed as $D = F[S_1, S_2, ..., S_f]$, where $f = |V(F)| \geq 2$, and let $D_0 = D - \cup_{i=1}^{f}\{uv : u, v \in V(S_i)\}$. Then D is strong if and only if D_0 is strong.*

Here is a useful observation on locally semicomplete digraphs due to Bang-Jensen. The proof is left as Exercise 7.38.

Lemma 7.13.4 *[44] Let D be a strong locally semicomplete digraph and let S be a minimal (not necessarily minimum) separating set of D. Then $D - S$ is connected.* □

Lemma 7.13.5 *Let $D = (V, A)$ be a k-strong digraph and let D' be obtained from D by adding a new set of vertices X and joining each vertex of X to V in such a way that $|N^+_{D'}(v)|, |N^-_{D'}(v)| \geq k + 1$ for each $v \in X$. Then D' is k-strong. If D' is not also $(k+1)$-strong, then every minimum separating set of D' is also a minimum separating set of D.*

Proof: Suppose D' is not $(k+1)$-strong and let S' be a minimum separating set of D'. Then $|S'| \leq k$. Let $S = S' \cap V(D)$. Since every vertex of $X - S'$ has an in-neighbour and an out-neighbour in $V - S$ we get that $D - S$ is not strong and hence $S = S'$ must hold and S' is also separating in D. This implies that $|S'| = k$, D' is k-strong and every minimum separating set of D' is also a minimum separating set of D. □

7.14 Highly Connected Orientations of Digraphs

We saw in Corollary 7.2.7 that every strong digraph without a bridge has a strong orientation. In this section we investigate how much of the degree of arc-strong or vertex-strong connectivity of a digraph D comes from its 2-cycles. More precisely, suppose we must delete one arc of every 2-cycle (thus obtaining an orientation of D), can we always maintain a high arc-strong, respectively vertex-strong, connectivity if the starting digraph has high arc-strong, respectively vertex-strong, connectivity? It is not difficult to see that we may not be able to preserve the same degree of arc-strong, respectively vertex-strong, connectivity, not even if D is semicomplete. See Figure 7.15 for an example. So the question is whether there exist functions $f(k), g(k)$ with the property that every $f(k)$-strong (($g(k)$-arc-strong) digraph contains a spanning k-strong (k-arc-strong) subgraph without cycles of length 2.

Let us first consider arc-strong connectivity. Note that every k-arc-strong oriented graph D must have $UG(D)$ $2k$-edge-connected. In particular, if G is an undirected graph with edge-connectivity $\lambda(G) = 2k - 1$ and $\overleftrightarrow{G}$ is the complete biorientation of G, then D does not contain a spanning k-arc-strong subgraph. Hence the following result due to Jackson and Thomassen implies that $g(k) = 2k$ and this is the best possible by the remark above.

Figure 7.15 A 2-strong semicomplete digraph which has no 2-arc-strong spanning subtournament. Undirected edges correspond to directed 2-cycles.

Theorem 7.14.1 *[451, 708] Every $2k$-arc-strong digraph has a k-arc-strong orientation.* □

Since we may convert a digraph to a mixed graph by replacing each 2-cycle with an undirected edge, Theorem 7.14.1 follows from Theorem 8.9.1.

The vertex-strong connectivity case seems much harder. Jackson and Thomassen posed the following conjecture (see [708]):

Conjecture 7.14.2 *Every $2k$-strong digraph has a k-strong orientation.*

If true this would be the best possible (meaning that we cannot weaken the vertex-strong connectivity demand by one, without adding further requirements). To see this let G be the k'th power of an undirected cycle $C = v_1v_2 \dots v_{2r}v_1$ on $2r$, $r > k$ vertices. It is not difficult to prove that G is $2k$-connected and that the only separating sets of size $2k$ in G are those obtained by taking two sets of k consecutive vertices on C, each separated by at least one vertex on both sides. From this it follows that, if we add the diagonals $v_1v_{r+1}, v_2v_{r+2}, \dots, v_rv_{2r}$, then we obtain a $(2k+1)$-connected graph H. Now let D be the complete biorientation of H. Then $\kappa(D) = 2k+1$ and it is clear that D cannot have a $(k+1)$-strong orientation, since $UG(D)$ is not $2(k+1)$-edge-connected. See Figure 7.16 and Exercise 7.43.

Note that, if an oriented graph D is k-strong, then $UG(D)$ is k-connected and $2k$-edge-connected. However, the converse is not true, that is, it is not enough to require that D is k-strong and that $UG(D)$ is $2k$-edge-connected in order to guarantee that D has a k-strong orientation. The semicomplete digraph in Figure 7.15 shows this and the example can be generalized to an arbitrary odd number of vertices by taking the second power on an odd cycle C and orienting the original edges as in Figure 7.15. This shows that Conjecture 8.6.7 can neither be extended to mixed graphs, nor to digraphs. Another example, due to Alon and Ziegler [708, page 406]–showing that $UG(D)$ may be k-connected and $2k$-edge-connected and still D has no k-strong orientation–is obtained from the complete biorientation of the graph constructed by taking two large complete graphs G_1, G_2 sharing just one vertex v and adding $k-1$ independent edges with one end in $V(G_1) - v$ and the other in $V(G_2) - v$.

Figure 7.16 A 7-connected 7-regular graph obtained from the third power of a 10-cycle by adding longest diagonals

Very little progress has been made on Conjecture 7.14.2 and it is not even known if there is some function $f(k)$ so that every $f(k)$-strong digraph has a k-strong orientation. Below we shall describe some results on special classes of digraphs.

Using the structure theorem (Theorem 4.11.15) for locally semicomplete digraphs Guo proved that every $(2k-1)$-strong locally semicomplete digraph which is not semicomplete can be oriented as a k-strong local tournament.

This was improved by Huang [437] who proved that the following much stronger statement holds:

Theorem 7.14.3 *[437] Every k-strong locally semicomplete digraph which is not semicomplete can be oriented as a k-strong local tournament.* □

Bang-Jensen and Thomassen [44] proved that for semicomplete digraphs the function $f(k)$ indeed exists. The value of this function was later improved by Guo.

Theorem 7.14.4 *[344] For every natural number k, every $(3k-2)$-strong locally semicomplete digraph has an orientation as a k-strong local tournament digraph.*

We will not prove the bound $3k-2$ here, but instead give the proof by Bang-Jensen and Thomassen that $f(k) \leq 5k$ for semicomplete digraphs. That proof illustrates the main ideas and Guo's proof is a refinement of the proof we give. Note that by Theorem 7.14.3 it is enough to consider semicomplete digraphs.

We prove by induction on k that every $5k$-strong semicomplete digraph D contains a spanning k-strong tournament. The case $k = 1$ is easy, since by Theorem 1.5.1, every strong semicomplete digraph has a Hamilton cycle. Let C be a Hamilton cycle in D. For every 2-cycle of D delete an arbitrary arc of that 2-cycle, unless one of its arcs is used by C. In the latter case we delete one arc of the 2-cycle so as to preserve C. We obtain a spanning strong tournament T of D. Note that the case $k = 1$ also follows easily from Corollary 7.2.7.

Suppose we have proved the statement for all $r \leq k-1$, that is, every $5r$-strong semicomplete digraph contains a spanning r-strong tournament. Let D be a $5k$-strong semicomplete digraph and suppose D does not contain a spanning k-strong tournament. We derive a contradiction to this assumption. First observe that we must have $|V(D)| \geq 5k+2$ since otherwise D is the complete digraph on $5k+1$ vertices and this clearly contains a k-connected spanning tournament.

By induction D contains a $(k-1)$-strong spanning tournament. Let T be chosen among all $(k-1)$-strong spanning tournaments of D such that the following holds:

(i) The number s of separating sets of size $k-1$ in T is minimum over all $k-1$-strong spanning subtournaments of D.
(ii) T has a separating set S of size $k-1$ such that the number m of strong components of $T-S$ is minimum taken over all separating sets of size $k-1$ of T.

Let S be some separating set of T such that $T-S$ has precisely m strong components $T_1, \ldots, T_m$ (written in the unique acyclic order). Let $U = V(T_1) \cup \ldots \cup V(T_{m-1})$ and $W = V(T_m)$. Since D is $5k$-strong it follows easily from Menger's theorem (Corollary 7.3.2) that in D there are $5k$ internally disjoint paths from W to U (see Exercise 7.19). At most $k-1$ of these can pass through S. Thus in $D-S$ there are at least $4k+1$ arcs from W to U. Let $U' \subset U$ ($W' \subset W$) be those vertices v of U (W) for which some arc in D from W to U has v as its head (tail). Since $D-S$ has at least $4k+3$ vertices, either U or W has size at least $2k+2$. Using this and the fact that $D-S$ has $4k+1$-internally disjoint (w,u)-paths for every choice of $u \in U, w \in W$, we get from Corollary 7.3.2 that either $|U'| \geq 2k+1$ or $|W'| \geq 2k+1$. By considering the converse of D if necessary, we may assume $|U'| \geq 2k+1$.

The digraph $T\langle U' \rangle$ is a tournament on at least $2k+1$ vertices and hence it has a vertex x with at least k out-neighbours in U'. Let y be a vertex in W' such that yx is an arc of D (y exists since $x \in U'$). In T we have the arc xy (since every vertex in U dominates every vertex in W) and since x has out-degree at least k in $T\langle U' \rangle$, there are at least k (x,y)-paths of length 2 in T. Let T' be the spanning tournament in D that we obtain from T by replacing the arc xy by the arc yx. Applying Lemma 7.9.6 we get that T' has no more than s minimum separating sets. However, it is easy to see that $T'-S$ is either strong (if $x \in V(T_1)$), or it has fewer strong components than $T-S$ and hence we obtain a contradiction to the choice of T according to (i), (ii). □

It can be seen by inspecting Guo's proof in [344] that $(3k-2)$-strong connectivity is the best bound one can prove using his approach. However, at least for $k=2$ this is not sharp when we have more than $2k$ vertices:

Proposition 7.14.5 *[83] Every 3-strong semicomplete digraph on at least 5 vertices contains a spanning 2-strong tournament.* □

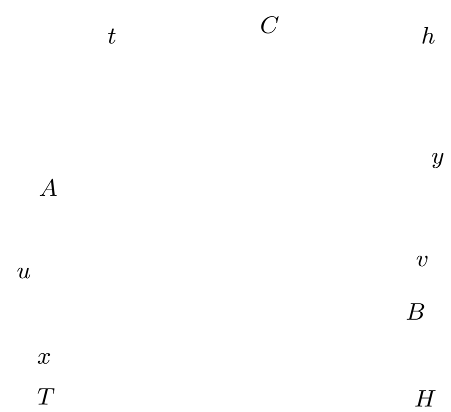


Figure 7.17 A k-strong semicomplete digraph D. All arcs between H and T go from H to T, except the 2-cycle xyx shown as an edge. All other arcs not shown are in 2-cycles. A, B, C represent arbitrary complete digraphs on at least one vertex each. The set C has $k-3$ vertices and hence k is defined as $|C|+3$. The one-way pair T, H (in $D-xy$) shows that we cannot delete xy and the one-way pair (T', H') with $T' = A \cup \{y, t\}, H' = B \cup \{h, x\}$ shows that we cannot delete yx.

It is perhaps worthwhile to notice that it does not seem easy to construct k-strong semicomplete digraphs with many vertices such that both arcs of some 2-cycle are critical with respect to k-strong connectivity (that is, deleting any of these arcs, the digraph is no longer k-strong). In order to obtain such a semicomplete digraph we must construct it so that we can prove that it is k-strong and that some 2-cycle xyx has the property that none of $D - \{xy\}$ and $D - \{yx\}$ is k-strong. Here the concept of one-way pairs and Lemma 7.7.2 is a useful tool. Suppose that none of $D - \{xy\}$ and $D - \{yx\}$ are k-strong. Then each of these must be $(k-1)$-strong and there must exist one-way pairs $(T_1, H_1), (T_2, H_2)$ in $D - xy$, respectively $D - yx$ with $h(T_1, H_1) = h(T_2, H_2) = k-1$ and $x \in T_1 \cap H_2, y \in T_2 \cap H_1$. Based on these findings one can construct a semicomplete digraph with the desired property. See Figure 7.17. We leave it to the reader to verify that D is indeed k-strong (Exercise 7.42).

Let us call a 2-cycle xyx in a semicomplete digraph D **critical** if we cannot delete any of the arcs xy, yx without decreasing the vertex-strong connectivity of D.

Problem 7.14.6 *Investigate the structure of the critical 2-cycles in semicomplete digraphs.*

As an illustration of the usefulness of the structural characterization of quasi-transitive digraphs in Theorem 4.8.5 we show how Theorem 7.14.4 implies the same statement for quasi-transitive digraphs.

Corollary 7.14.7 *For every natural number k, every $(3k-2)$-strong quasi-transitive digraph has an orientation as a k-strong quasi-transitive digraph.*

Proof: Let D be a $(3k-2)$-strong quasi-transitive digraph and let $D = Q[W_1, \ldots, W_q]$, $q = |Q|$, be a decomposition of D according to Theorem 4.8.5. By Corollary 7.13.2, the digraph D_0 obtained from D by deleting all arcs inside each W_i is also $(3k-2)$-strong. By Theorem 4.8.5, if Q contains a 2-cycle $q_iq_jq_i$, then each of W_i, W_j have size one. Now let H be a semicomplete digraph obtained from D_0 by adding an arbitrary arc between every pair of vertices inside each $V(W_i)$. Clearly H is (at least) $(3k-2)$-strong and hence, by Theorem 7.14.4, it contains a spanning k-connected tournament T (which is obtained from H by deleting one arc from every 2-cycle, that is, T is an orientation of H). By the way we constructed H, we have $T = Q'[T_1, \ldots, T_q]$ for some choice of tournaments $T_1, \ldots, T_{|Q|}$ on $|W_1|, \ldots, |W_q|$ vertices respectively. Here Q' is a spanning tournament in Q. Applying Corollary 7.13.2 to $T = Q'[T_1, \ldots, T_q]$, we get that the quasi-transitive digraph $D' = Q'[\overline{K}_{|W_1|}, \ldots, \overline{K}_{|W_q|}]$ is k-strong and by the remark above on 2-cycles in Q we see that D' is a spanning subgraph of D. It is easy to see that, if we delete an arc from every 2-cycle of a quasi-transitive digraph, then the result is a quasi-transitive digraph. Let W_i' be obtained from W_i by deleting one arc from every 2-cycle in W_i for $i = 1, 2, \ldots, W_q$. Now we see that $D'' = Q'[W_1', W_2', \ldots, W_q']$ is the desired k-strong orientation of D. □

Note that it also follows from the proof above that every $(3k-2)$-strong quasi-transitive digraph contains a spanning k-strong extended tournament.

7.15 Packing Cuts

In this section we consider directed multigraphs. Let $D = (V, A)$ be a directed multigraph which is connected, but not strongly connected. A **directed cut** (or just a **dicut**) in D is a set of arcs of the form $(X, V - X)$, where X is a non-empty proper subset of V such that there are no arcs from $V - X$ to X (i.e. $(X, V - X)$ is a one-way pair with $h(X, V - X) = 0$). Two directed cuts are arc-disjoint if they do not share an arc. Note that two dicuts $(X, V - X)$ and $(Y, V - Y)$ may be arc-disjoint but still $X \cap Y \neq \emptyset$. As an example consider a directed path $x_1x_2 \ldots x_k$. Here $\{(\{x_1, \ldots, x_i\}, \{x_{i+1}, \ldots, x_k\}) : 1 \leq i \leq k-1\}$ is a family of $k-1$ arc-disjoint cuts (each having precisely one arc). Clearly these cuts overlap considerably when we consider their vertex sets. For simplicity we will sometimes denote a dicut $(X, V - X)$ just by the set X.

A **dijoin** is a subset $A' \subset A$ which covers all dicuts. Define $\Omega(D)$ and $\tau(D)$ as follows

$$\begin{aligned} \Omega(D) &= \text{ the maximum number of arc-disjoint dicuts in } D. \\ \tau(D) &= \min\{|A'| : A' \text{ is a dijoin}\}. \end{aligned} \tag{7.36}$$

Suppose $D = (V, A)$ is connected but not strongly connected. Then it is clear that we can obtain a strong directed multigraph by contracting certain arcs. It is also clear that, if we contract an arc a which is not an arc of a dicut $(X, V - X)$, then in the resulting directed multigraph $D' = (V', A')$, the corresponding pair $(X', V - X')$ is still a dicut. On the other hand, if A' is a dijoin and we contract all arcs of A', then the resulting directed multigraph is strong. Let $\rho(D)$ denote the minimum number of arcs whose contraction in D leads to a strong directed multigraph. Then it follows from the discussion above that

$$\Omega(D) \leq \rho(D) \leq \tau(D). \tag{7.37}$$

Note that, if D is a directed (x, y)-path on r vertices, then $a_1(D) = 1$, since we may add a new arc yx and get a strong digraph. However, in order to obtain a strong directed multigraph by contracting arcs, we must contract $r-1$ arcs, showing that $\rho(D) = r - 1$. This proves that $\rho(D)$ and $a_1(D)$ may be arbitrarily far apart.

Let D be a directed multigraph. Recall that the operation of subdividing an arc consists of replacing the arc xy in question by the path xuy of length two, where u is a new vertex. If several arcs are subdivided, then all the new vertices (used to subdivide these arcs) are distinct.

Lemma 7.15.1 *Let $D = (V, A)$ be a directed multigraph and let D' be obtained from D by subdividing each arc once. If D has k arc-disjoint dicuts, then D' has $2k$ arc-disjoint dicuts.*

Proof: Let $D' = (V', A')$ be obtained from D by subdividing each arc once. Let $X_1, \dots, X_k$ be chosen such that the dicuts $(X_1, V - X_1), \dots, (X_k, V - X_k)$ are arc-disjoint in D. For each dicut $(X_i, V - X_i)$ we denote by X_i' the set we obtain in D' by taking the union of X_i and the new vertices that subdivide the arcs leaving X_i. Now it is easy to see that each of the dicuts $(X_1, V' - X_1), (X_1', V' - X_1'), \dots, (X_k, V' - X_k), (X_k', V' - X_k')$ are arc-disjoint. □

The next theorem, due to Lucchesi and Younger shows that in fact equality holds everywhere in (7.37).

Theorem 7.15.2 (the Lucchesi-Younger theorem) *[528] Let $D = (V, A)$ be a directed multigraph which is connected and either D has just one vertex, or it is not strongly connected. Then $\Omega(D) = \tau(D)$.*

Proof: We give a proof due to Lovász [521]. The proof is by induction on the number of arcs in A. If $A = \emptyset$, then D has precisely one vertex and there are no dicuts. Hence the statement of the theorem is vacuously true.

Now let $a \in A$ be an arbitrary arc. Contract a and consider the resulting directed multigraph D/a. Note that the dicuts of D/a are exactly those in D which do not contain the arc a. By induction, $\tau(D/a) = \Omega(D/a)$. Hence if $\Omega(D/a) \leq \Omega(D)-1$, then we can cover all dicuts in D by $\tau(D/a)+1 \leq \Omega(D)$ arcs and the theorem is proved. Hence we may assume that

$$\Omega(D/a) = \Omega(D) \text{ for every arc } a \in A. \tag{7.38}$$

By Lemma 7.15.1, if we subdivide all arcs in A, then the resulting digraph has at least $\Omega(D) + 1$ arc-disjoint dicuts (with equality only if $\Omega(D) = 1$). Hence, starting from D and subdividing arbitrary (not previously subdivided) arcs, we will get a sequence of directed multigraphs $D_0 = D, D_1, \ldots, D_h$, where $\Omega(D_i) = \Omega(D)$ for each $i \leq h-1$ and $\Omega(D_h) = \Omega(D) + 1$. Let f be the last arc we subdivided in this process and let $H = D_{h-1}$. Now H contains $\Omega(D) + 1$ dicuts $X_1, X_2, \ldots, X_{\Omega(D)+1}$ such that only two of them have an arc in common and that arc is f.

Observe that H/f arises from G/f by subdivision. Hence, by the assumption (7.38), $\Omega(H/f) = \Omega(D)$ and so H contains $\Omega(D)$ arc-disjoint dicuts $Y_1, Y_2, \ldots, Y_{\Omega(D)}$ none of which contains the arc f. This implies that $X_1, X_2, \ldots, X_{\Omega(D)+1}, Y_1, Y_2, \ldots, Y_{\Omega(D)}$ is a collection of $2\Omega(D) + 1$ dicuts in H such that no arc belongs to more than two of these. Thus the following lemma will give us a contradiction, implying that (7.38) cannot hold and hence the theorem follows.

Lemma 7.15.3 *If a digraph D contains at most k arc-disjoint dicuts, and $\mathcal{C}$ is any collection of dicuts in D such that no arc belongs to more than two dicuts in $\mathcal{C}$, then $|\mathcal{C}| \leq 2k$.*

Proof of Lemma 7.15.3: Call two dicuts $(X, V-X), (Y, V-Y)$ **crossing** if X and Y are crossing as sets. The first step is to uncross crossing dicuts in the family.

It follows from (7.2) that, if $(X, V-X), (Y, V-Y)$ are crossing dicuts, then each of $(X \cup Y, V-(X \cup Y)), (X \cap Y, V-(X \cap Y))$ are dicuts and $d(X,Y) = 0$. Furthermore, the dicuts $(X \cup Y, V-(X \cup Y))$ and $(X \cap Y, V-(X \cap Y))$ cover each arc of D the same number of times as the dicuts $(X, V-X)$, $(Y, V-Y)$ (here we used that $d(X,Y) = 0$). Let $\mathcal{C}' = \mathcal{C} - \{(X, V-X), (Y, V-Y)\} + \{(X \cup Y, V-(X \cup Y)), (X \cap Y, V-(X \cap Y))\}$. Then $\mathcal{C}'$ has the same property as $\mathcal{C}$ that no arc covers more than two dicuts in $\mathcal{C}$ and furthermore we have

$$\sum_{(X,V-X)\in\mathcal{C}} |X|^2 \leq \sum_{(Z,V-Z)\in\mathcal{C}'} |Z|^2, \tag{7.39}$$

because $|X \cup Y|^2 + |X \cap Y|^2 > |X|^2 + |Y|^2$ when X, Y cross. Hence, if we replace crossing dicuts pairwise as we did above, then we will eventually reach a new family $\mathcal{C}^*$ of size $|\mathcal{C}|$ such that the dicuts in $\mathcal{C}^*$ are pairwise non-crossing and no arc of D belongs to more than two dicuts in $\mathcal{C}^*$. Hence it suffices to prove that $\mathcal{C}^*$ contains at most $2k$ dicuts.

Let $\mathcal{C}^* = \{Z_1, Z_2, \ldots, Z_M\}$ and let $A_i = (Z_i, V - Z_i)$, $i = 1, 2, \ldots, M$ be the corresponding arc sets. Construct an undirected graph $G(\mathcal{C}^*) = (V, E)$ as follows: $V = \{v_1, v_2, \ldots, v_M\}$ and there is an edge between v_i and v_j if and only if $A_i \cap A_j \neq \emptyset$. Since D contains at most k arc-disjoint dicuts, it follows that $G(\mathcal{C}^*)$ has at most k independent vertices. Hence it suffices to show that $G(\mathcal{C}^*)$ is a bipartite graph since then we get $|\mathcal{C} = |\mathcal{C}^*| \leq 2k$.

Let $v'_1 v'_2 \ldots v'_s v'_1$ be an arbitrary cycle in $G(\mathcal{C}^*)$. Note that the arc sets of the corresponding dicuts $A'_1, \ldots, A'_s$ must be different, since if $(Z'_i, V - Z'_i) = (Z'_j, V - Z'_j)$ for some $1 \leq i < j \leq s$, then every arc in $(Z'_i, V - Z'_i)$ is covered twice (by $(Z'_i, V - Z'_i)$ and by $(Z'_j, V - Z'_j)$) and hence the vertices v'_i, v'_j each have degree one in $G(\mathcal{C}^*)$, contradicting the fact that they are on a cycle. Note also that if two dicuts $(X, V - X)$ and $(Y, V - Y)$ have $X \cup Y = V$, then they are arc-disjoint and hence are not adjacent in $G(\mathcal{C}^*)$.

X X Y

Y

(a)

Y X Y

X

(b)

Figure 7.18 Illustration of the definition of being to the right and left for cuts. In the two situations in part (a) (part (b)) the dicut $(X, V - X)$ is to the left (right) of the dicut $(Y, V - Y)$. In the right part of (a) we have $X \cup Y = V$.

Since $A'_i \cap A'_{i+1} \neq \emptyset$ for $i = 0, 1, \ldots, s-1$, where $A'_0 = A'_s$, it follows from our remarks above that we have either $Z'_i \subset Z'_{i+1}$ or $Z'_{i+1} \subset Z'_i$. We prove that the two possibilities occur alternatingly and hence s is even. Suppose not, then without loss of generality we have $Z'_0 \subset Z'_1 \subset Z'_2$. Let us say that a dicut A'_i is **to the left of** another dicut A'_j if either $Z'_i \subset Z'_j$, or $Z'_i \cup Z'_j = V$ (which is equivalent to $V - Z'_i \subset Z'_j$) and that A'_i is **to the right of** A'_j if $Z'_i \cap Z'_j = \emptyset$ (which is equivalent to $Z'_i \subset V - Z'_j$), or $Z'_j \subset Z'_i$ (which is equivalent to $V - Z'_i \subset V - Z'_j$). See Figure 7.18. Since $\mathcal{C}^*$ contains no crossing members, each $A'_i \neq A'_j$ is either to the right or to the left of A'_j. Since A'_2 is to the right of A'_1 and $A'_0 = A'_s$ is to the left of A'_1, it follows that there is some $2 \leq j \leq s-1$ such that A'_j is to the right of A'_1 and A'_{j+1} is to the left of A'_1. Suppose first that $Z'_j \cap Z'_1 = \emptyset$, then we cannot have $Z'_{j+1} \subset Z'_1$ as A'_{j+1} and A'_j have a common arc. So we must have $Z'_1 \cup Z'_{j+1} = V$, but then any arc a common to A'_j and A'_{j+1} enters Z'_1, contradicting that $d^-(Z'_1) = 0$.

Hence we must have $Z'_1 \subset Z'_j$. The fact that A'_j, A'_{j+1} have a common arc a (and hence either $Z'_j \subset Z'_{j+1}$ or $Z'_{j+1} \subset Z'_j$) implies that, by the choice of j, we have $Z'_{j+1} \subset Z'_1 \subset Z'_{j+1}$. But now the arc a belongs to three dicuts A'_1, A'_j and A'_{j+1}, a contradiction. This completes the proof of the lemma and, by the remark above, also the proof of the theorem. $\square$

Combining (7.37) and Theorem 7.15.2, we obtain:

Corollary 7.15.4 *Let D be a non-strong directed multigraph whose underlying graph is connected. Then $\rho(D) = \tau(D)$, that is, D can be made strongly connected by contracting $\tau(D)$ arcs.*

The proof of Theorem 7.15.2 is not constructive but using submodular flows one can find a minimum dijoin $A' \subseteq A$ of D in polynomial time. See Corollary 8.8.10.

7.16 Application: Small Certificates for k-(Arc)-Strong Connectivity

We complete the chapter with a topic that is, of practical interest and at the same time illustrates important applications of several of the concepts from the chapter.

Let $D = (V, A)$ be a directed multigraph which is k-(arc)-strong. What is the cost (measured in the number of arcs, or the sum of arc costs if these are present) of a minimum cost spanning subgraph $D' = (V, A')$ of D such that D' is k-(arc)-strong? A spanning k-(arc)-strong subgraph D' of D is called a **certificate** for k-(arc)-strong connectivity of D. Finding an optimal certificate (that is, one with the smallest cost) for k-(arc)-strong connectivity is a difficult problem, even when $k = 1$. Namely, if all costs are 1 (that is, we only count the number of arcs), then the optimal D' has n arcs if and only if D has a Hamilton cycle. Thus the problem is $\mathcal{NP}$-hard already when $k = 1$ and we have uniform costs. By the remark above, the Hamilton cycle problem is a special case of the problem of finding an optimal certificate for strong connectivity. This makes it interesting to consider classes of digraphs for which we know that the Hamilton cycle problem is polynomially solvable and to see what we can say about the complexity of finding the optimal certificate for vertex-strong connectivity. This was done in Section 6.11 for some classes of generalizations of tournaments.

In practical applications, e.g. to speed up algorithms, it is often important to work with a small certificate for k-(arc)-connectivity. This means that one is interested in finding polynomial algorithms which find a certificate D' for

k-(arc)-strong connectivity with the property that the cost of D' is not more than some constant (larger than 1) times the cost of the optimal certificate[5].

In this section we present some recent results by Cheriyan and Thurimella [151] which show that we can approximate the size of a smallest k-(arc)-strong spanning subgraph better, the higher k is.

7.16.1 Finding Small Certificates for Strong Connectivity

For $k = 1$, the 2-approximation algorithm sketched in the proof of Theorem 7.10.1 could be used since D is 1-strong if and only if it is 1-arc-strong. However, one can do better than this. When $k = 1$ the problem of finding a small certificate for strong connectivity is a special case[6] of a problem which is also called the problem of finding the **minimum equivalent subdigraph** of a directed multigraph. That is, given D, find a spanning subgraph D' with as few arcs as possible such that D' contains an (x, y)-path if and only if D does for every choice of $x, y \in V(D)$ (it is clear that D' will not contain multiple arcs and hence must be a digraph). This problem, which has many practical applications, has been considered several times in the literature, see e.g. [5, 317, 434, 478, 479, 554, 636, 673]. See also Section 4.3.

Now let D be a strongly connected digraph (recall that we may assume that D has no multiple arcs since multiple arcs will not be present in a minimally strong directed multigraph). Khuller, Raghavachari and Young [478] gave a 1.65-approximation algorithm for the size of a smallest strongly connected subgraph of any strongly connected digraph. The idea in the algorithm from [478] is to find a long cycle, contract it and continue recursively. The authors were able to show that this approach can be performed in such a way that one obtains a solution in polynomial time with no more than 1.65 times the size of an optimum solution. This was later improved to about 1.61 using results from [479].

Khuller, Raghavachari and Young also considered the restriction when the digraph in question has no cycle with more than r arcs. Then the problem is known under the name SCCS_r [478]. In [479] it is shown that if one only considers digraphs with no cycle longer than 3, then the optimal certificate can be found in polynomial time. The algorithm is based on the following result.

[5] Such an algorithm which is polynomial and finds a solution (for a minimization problem) whose cost is at most μ times the value of an optimal solution ($\mu \geq 1$) is called an **μ-approximation algorithm** for the problem.

[6] It is in fact the most important ingredient since once we know the best subgraph inside each strong component, we can contract each strong component to a vertex and consider the problem of finding a minimum equivalent subdigraph of an acyclic directed multigraph. That problem is solvable in polynomial time by Proposition 4.3.5.

Theorem 7.16.1 *[479] The SCCS$_3$ problem reduces in time $O(n^2)$ to the problem of finding a minimum edge-cover*[7] *in a bipartite graph.*

This gives an $O(n^2 + m\sqrt{n})$ time algorithm for the SCCS$_3$ problem, since the problem of finding a minimum edge cover in a bipartite graph is equivalent to the problem of finding a maximum matching in such a graph [497]. The latter problem can be solved in time $O(\sqrt{n}m)$ (see Theorem 3.11.1).

However, already the SCCS$_5$ problem is $\mathcal{NP}$-hard and the SCCS$_{17}$ is even MAX $\mathcal{SNP}$-hard, implying that there cannot exist a polynomial time approximation scheme for this problem, unless $\mathcal{P} = \mathcal{NP}$ [478]. Khuller, Raghavachari and Young posed the following problem concerning the weighted version. Here the goal is to find a spanning strong subdigraph of minimum weight.

Problem 7.16.2 *[478] Does there exist a μ-approximation algorithm for minimum weight strong connectivity certificates with $\mu < 2$?*

The existence of a polynomial algorithm with approximation guarantee 2 follows from the fact that finding a minimum cost in-branching (out-branching) with a given root can be done in polynomial time (see Section 9.10). Indeed, if F_r^- (F_r^+) is a minimum cost in-branching (out-branching) rooted at r, then $D' = (V, A(F_r^-) \cup A(F_r^+))$ is strong and clearly has cost at most twice the optimum. In Exercise 7.49 the reader is asked to show that the approximation guarantee of this approach cannot be lower than 2.

Once again we remind the reader that in Section 6.11 we showed that an optimal strong subdigraph of a digraph D can be found in polynomial time in case D belongs to one of several classes of generalizations of tournaments.

7.16.2 Finding k-Strong Certificates for $k > 1$

Cheriyan and Thurimella recently gave an approximation algorithm with a very good approximation guarantee by combining some fairly elementary results on subgraphs of (di)graphs with Mader's powerful result on anti-directed trails and k-critical arcs (Theorem 7.10.7). We start with the two subgraph results and then describe the simple algorithm from [151].

Proposition 7.16.3 *[151] Let $B = (V, E)$ be a bipartite graph with minimum degree k. Let $E' \subset E$ be a minimum cardinality subset of E with the property that $B' = (V, E')$ has minimum degree $k-1$. Then $|E'| \leq |E| - |V|/2$ and this bound is best possible.* □

Proposition 7.16.4 *There exists a polynomial algorithm $\mathcal{A}$ which, given a directed multigraph $D = (V, A)$ with minimum semi-degree $\delta(D) \geq r$, returns a minimum cardinality subset $A' \subseteq A$ such that the directed multigraph $D' = (V, A')$ has $\delta(D') \geq r$.*

[7] An **edge-cover** of an undirected graph $G = (V, E)$ is a set of edges $E' \subset E$ such that every $v \in V$ is incident with at least one edge from E'.

Proof: This (as well as the more general minimum arc cost version) can be solved using a minimum value (minimum cost) flow algorithm on a suitable network constructed from D (see Exercise 7.47). $\square$

Theorem 7.16.5 *[151] There exists a polynomial algorithm which, given a digraph $D = (V, A)$ which is k-strong, returns a spanning k-strong subgraph $D'' = (V, A'')$ of D such that $|A''| \leq (1 + \frac{1}{k})|A^*_{opt}|$, where A^*_{opt} denotes a minimum cardinality arc set $A^*_{opt} \subseteq A$ such that $D^* = (V, A^*_{opt})$ is k-strong.*

Proof: Let $\mathcal{B}$ be the following algorithm:

Input: A directed graph $D = (V, A)$ and a number k such that D is k-strong.
Output: A small certificate $\tilde{D} = (V, \tilde{A})$ for k-strong connectivity of D.

1. Use the algorithm $\mathcal{A}$ of Proposition 7.16.4 to find a minimum cardinality subset $A' \subset A$ such that the digraph $D' = (V, A')$ has $\delta(D') \geq k - 1$;
2. Let $\bar{A} = A - A'$;
3. Find a minimal (with respect to inclusion) subset $A'' \subset \bar{A}$ with the property that $\tilde{D} = (V, A' \cup A'')$ is k-strong;
4. Return $\tilde{D}$.

Clearly $\tilde{D} = (V, A' \cup A'')$ is k-strong, so we can concentrate on the approximation factor and the running time.

To see that the approximation factor is as claimed, let $D^* = (V, A^*_{opt})$ denote an arbitrary optimal certificate for k-strong connectivity of D. Clearly we have

$$|A'| \leq |A^*_{opt}|. \tag{7.40}$$

To bound the size of A'' we use Theorem 7.10.7. We claim that $D'' = (V, A'')$ has no anti-directed trail. Suppose it does and let T be an anti-directed trail in D''. Note that T is a subdigraph of $\tilde{D}$. Hence we can apply Theorem 7.10.7 to $\tilde{D}$. Now it follows from the fact that every arc of A'' is k-critical in $\tilde{D}$ that only (b) or (c) can hold in Theorem 7.10.7 when applied to $\tilde{D}$. However by the choice of A', neither (b), nor (c) can hold in $\tilde{D}$ since every source (sink) of T has out-degree (in-degree) at least $k + 1$ in D. Thus T cannot exist and D'' has no anti-directed trail. From this it follows, by considering the bipartite representation $BG(D'')$, that

$$|A''| \leq 2|V| - 1. \tag{7.41}$$

We leave the proof of this as Exercise 7.48 (recall the definition of $BG(D)$ in Chapter 1).

Combining (7.40) and (7.41), it is easy to see that the approximation guarantee of $\mathcal{B}$ is at least as good as $(1 + \frac{2}{k})$. However, using Proposition 7.16.3 we can do a little better. Let A^{**} be a minimum cardinality subset of A^*_{opt} so that the spanning subgraph $D^{**} = (V, A^{**})$ has $\delta(D^{**}) \geq k - 1$.

Consider $BG(D^*)$ and the edge sets E^*, E^{**} corresponding to A^*_{opt} and A^{**}. By Proposition 7.16.3

$$\begin{aligned}|A^{**}| = |E^{**}| &\leq |E^*| - |V(BG(D^*))|/2 \\ &= |A^*_{opt}| - |V|.\end{aligned} \tag{7.42}$$

By the choice of A' we have $|A'| \leq |A^{**}|$ and combining (7.41) and (7.42) gives

$$\begin{aligned}\frac{|\tilde{A}|}{|A^*_{opt}|} &\leq \frac{|A^*_{opt}| - |V| + (2|V| - 1)}{|A^*_{opt}|} \\ &\leq 1 + \frac{1}{k},\end{aligned} \tag{7.43}$$

since clearly $|A^*_{opt}| \geq k|V|$.

It remains to prove that $\mathcal{B}$ can actually be performed in polynomial time. Step 1 is performed by the polynomial algorithm $\mathcal{A}$ whose existence is proved in Exercise 7.47. Step 3 can be implemented by starting from D and deleting arcs of $\bar{A}$ one by one until every remaining arc from $\bar{A}$ is k-critical. Clearly this part can be done in polynomial time, using any algorithm for checking whether a digraph is k-strong. □

The authors claimed in [151] that the running time of the algorithm can be made $O(k|A|^2)$.

7.16.3 Certificates for k-Arc-Strong Connectivity

In Theorem 7.10.1 we saw that for k-arc-strong connectivity one can approximate the size (measured in number of arcs) of an optimal certificate for k-arc-strong connectivity within a factor of 2, using arc-disjoint in- and out-branchings. In Chapter 9 we shall see that one can even handle the case when there are costs on the arcs and still get a 2-approximation algorithm. Since D is 1-arc-strong if and only if it is strong, we covered the case $k = 1$ in the discussion above for vertex-strong connectivity.

Cheriyan and Thurimella showed that also for arc-strong connectivity one can approximate the size of an optimal certificate better the higher the arc-strong connectivity is.

Theorem 7.16.6 *[151] There exists a polynomial algorithm which given a digraph $D = (V, A)$ which is k-arc-strong returns a spanning k-arc-strong subgraph $D' = (V, A')$ of D such that $|A'| \leq (1 + 4/\sqrt{k})|A_{opt}|$, where $|A_{opt}|$ denotes the number of arcs in an optimal certificate for k-arc-strong connectivity. The running time of the algorithm is $O(k^3|V|^3 + |A|^{1.5}(\log(|V|)^2)$.*

Proof: The idea is similar to the vertex-strong connectivity case so we will only sketch the proof here. Let $D = (V, A)$ be k-arc-strong. First find, using the algorithm $\mathcal{A}$, a minimum cardinality subset $U \subset A$ such that $H = (V, U)$ has $\delta(H) \geq k$. Then find an inclusion-wise minimal subset $U' \subset (A - U)$ such that $\tilde{H} = (V, U \cup U')$ is k-arc-strong. As in the proof of Theorem 7.16.5, the key step is to estimate the size of U', since $|U|$ is clearly at most the size of an optimal solution.

To estimate $|U'|$ we use the following definition. An arc uv of a k-arc-strong digraph W is **special** if $W - uv$ is not k-arc-strong and furthermore $d^+_W(u), d^-_W(v) \geq k+1$. Clearly each arc in the set U' is special in the digraph $\tilde{H}$. Hence we can apply the following estimate.

Theorem 7.16.7 *[151] Let $k \geq 1$ be an integer and let $W = (V, A)$ be k-arc-strong. The number of special arcs in W is at most $4\sqrt{k}|V|$.* □

Combining this with the fact that $|A'|$ is no more than the size of an optimal certificate the theorem follows. For the complexity bound we refer to [151]. □

See also [152] for an expanded version of [151].

7.17 Exercises

7.1. **Submodularity of $|N^-|$ and $|N^+|$**. Prove Proposition 7.1.3.

7.2. **(−)** Prove Corollary 7.2.3.

7.3. **Complexity of converting between a directed multigraph and its network representation.** Show that given a directed multigraph D one can construct its network representation $\mathcal{N}(D)$ in polynomial time. Show that converting in the other direction cannot always be done in a time which is polynomial in the size of the network representation. Hint: recall that we assume that capacities are represented as binary numbers.

7.4. Prove that, if $D = (V, A)$ is an eulerian directed multigraph and X is a proper non-empty subset of V, then $d^+(X) = d^-(X)$.

7.5. Show that every k-regular tournament is k-arc-strong.

7.6. **(−)** Prove that every eulerian directed multigraph is strong.

7.7. Let D be a digraph, let s be a vertex of D and let k be a natural number. Suppose that $\min\{\lambda(s, v), \lambda(v, s)\} \geq k$ for every vertex $v \in V(D) - s$. Prove that $\lambda(D) \geq k$.

7.8. **(−) Vertex-strong connectivity of planar digraphs**. In a planar undirected graph G on n vertices and m edges we always have $m \leq 3n - 6$ by Euler's formula (see Corollary 4.14.3). Conclude that no planar digraph is 6-strong.

7.9. **(−)** Let D be a k-strong digraph and let a be an arbitrary arc of D. Prove that $D - a$ is $(k-1)$-strong.

7.10. **(−)** Let D be a k-strong digraph and let a be an arbitrary arc of D. Let D' be obtained from D by reversing a. Prove that D' is $(k-1)$-strong.

7.11. **Connectivity of powers of cycles**. Recall that the kth power of a cycle $C = v_1 \dots v_n v_1$ is the digraph with vertex set $\{v_1, \dots, v_n\}$ and arc set $\{v_i v_j : i+1 \le j \le i+k, i = 1, 2, \dots, n\}$. Prove that the kth power of a cycle on $n \ge k+1$ vertices is k-strong.

7.12. **(−)** For every natural number k describe a k-strong digraph D for which reversing any arc of D results in a digraph with vertex-strong connectivity less than k.

7.13. **(+) Finding k arc-disjoint (x,y)-paths of minimum total weight.** Let $D = (V, A, w)$ be a directed multigraph with weights on the arcs, let $x, y \in V$ be distinct vertices and let k be a natural number. Describe a polynomial algorithm which either finds a minimum weight collection of k arc-disjoint (x,y)-paths, or demonstrates that D does not have k arc-disjoint (x,y)-paths. Hint: use flows. Argue that you can find k internally disjoint (x,y)-paths of minimum total weight using a similar approach.

7.14. **(+) Minimum augmentations to ensure k arc-disjoint (s,t)-paths.** Let $D = (V, A, w)$ be a directed multigraph, let s, t be special vertices of D and let k be a natural number such that D does not have k arc-disjoint (s,t)-paths. Prove that it is possible to augment D optimally so that the new directed multigraph has k arc-disjoint (s,t)-paths and all new arcs go from s to t. Now consider the same problem when there are weights on the arcs. Devise an algorithm to find the cheapest set of new arcs whose addition to D gives a directed multigraph with k arc-disjoint (s,t)-paths. Hint: use min cost flows.

7.15. **(+) Minimum number of new edges to add so that the new digraph has k arc-disjoint out-branchings at s**. Show how to reduce this problem to the general k-arc-connectivity augmentation. Try to derive a min-max formula for the optimal number of new arcs.

7.16. **Equivalence of Menger's theorem and the Max Flow Min Cut theorem**. Prove that Menger's theorem implies the Max-flow Min-cut theorem for network in which all capacities are integer valued.

7.17. **Refining Menger's theorem**. Let D be a k-strong directed multigraph. Let $x_1, x_2, \dots, x_r, y_1, y_2, \dots, y_s$ be distinct vertices of D and let $a_1, a_2, \dots, a_r$, $b_1, b_2, \dots, b_s$ be natural numbers such that

$$\sum_{i=1}^{r} a_i = \sum_{j=1}^{s} b_j = k.$$

Prove that D contains k internally disjoint paths $P_1, P_2, \dots, P_k$ with the property that precisely a_i (b_j) of these start at x_i (end at y_j). Argue that the analogous statement concerning arc-disjoint paths is true if we replace vertex-strong connectivity by arc-strong connectivity.

7.18. **Refining Menger's theorem for undirected graphs**. Prove the analogous statement of Exercise 7.17 for undirected graphs.

7.19. **Menger's theorem for sets of vertices**. Let D be k-strong and let X, Y be distinct subsets of $V(D)$. Prove that D contains k internally disjoint paths which start in X and end in Y and have only their starting (ending) vertex in X (Y).

7.20. **Augmenting acyclic tournaments to k-strong connectivity**. Prove that an acyclic tournament on $n \geq k+1$ vertices can be made k-strong by adding $\frac{k(k+1)}{2}$ arcs. Hint: use Exercise 7.11.

7.21. **(+) Ear decomposition in linear time**. Supply the algorithmic details missing in the proof of Corollary 7.2.5. In particular, describe how to store the arcs in such a way that the ear decomposition can be found in linear time.

7.22. **(+) Strong orientations of mixed multigraphs in linear time**. Give an $O(n+m)$ algorithm for finding a strong orientation of a mixed multigraph or a proof that no such orientation exists (Chung, Garey and Tarjan [157]).

7.23. **(+) Cycle subdigraphs containing specified arcs**. Prove the following. Suppose D is k-strong (respectively, k-arc-strong) and $e_1, e_2, \ldots, e_k$ are arcs of D such that no two arcs have a common head or tail. Then D has a cycle subgraph (respectively, a collection of arc-disjoint cycles) $\mathcal{F} = \{C_1, \ldots, C_r\}$, $1 \leq r \leq k$ such that each arc e_i is an arc of precisely one of the cycles in $\mathcal{F}$. Hint: add two new vertices s, t, connect these appropriately to D and then apply Menger's theorem to s and t.

7.24. Prove the following: Every s-regular round digraph has strong vertex- and arc-connectivity equal to s (Ayoub and Frisch [34]).

7.25. **Connectivity of complete biorientations of undirected graphs.** Let G be a k-connected undirected graph for some $k \geq 1$ and let D be the complete biorientation of G. Prove that for every arc xy of D the digraph $D - xy$ is k-strong if and only if $D - \{xy, yx\}$ is k-strong.

7.26. **Obtaining new k-strong digraphs by adding vertices**. Let D be a k-strong digraph, let x be a new vertex and let D' be obtained from D and x by adding k arcs from x to distinct vertices of D and k arcs from distinct vertices of D to x. Prove that D' is k-strong.

7.27. **Obtaining new k-arc-strong directed multigraphs by adding new vertices**. Let D be a k-arc-strong directed multigraph, let x be a new vertex and let D' be obtained from D and x by adding k arcs from x to arbitrary vertices of D and k arcs from arbitrary vertices of D to x. Prove that D' is k-arc-strong.

7.28. **(+) Greedy deletion of arcs in Frank's algorithm**. Show how to implement Steps 2 and 3 of Frank's algorithm in Section 7.6 by using flows to find the maximum number of arcs that can be deleted for each vertex v_i (Frank [258]). Hint: let t be a vertex of $V - v_i$, identify s and t to one vertex t' and then calculate $\lambda(v_i, t')$ in the resulting directed multigraph. Do this for all $t \in V - v_i$ and let ρ be the smallest of the numbers calculated. Using Menger's theorem, show that we may delete precisely $\min\{\mu(v_i, s), \rho - k\}$ arcs from v_i to s without violating (7.9).

7.29. Perform Frank's algorithm on the digraph in Figure 7.19 when the goal is to obtain a 2-arc-strong directed multigraph.

Figure 7.19 A directed graph H.

7.30. **(+) Finding an admissible split**. Show that Step 5 of Frank's algorithm in Section 7.6 can be implemented using flows. That is, show how to decide if a given splitting (us, sv) is admissible, that is, it preserves k-arc-strong connectivity in V (Frank [258]). Hint: we need to decide if there is a set $U \subset V$ such that $u, v \in U$ and $d^+(U) = k$ or $d^-(U) = k$. This can be done using flows in a way similar to that outlined in the hint above.

7.31. (+) Let $D = \vec{C}_n[\overleftrightarrow{K}_k, I_1, \overleftrightarrow{K}_k, \ldots, \overleftrightarrow{K}_k, I_1]$, where I_1 denotes the digraph, that is, just an isolated vertex and n is an even number. Prove that $\gamma_k(D) = k$. Try to determine $a_k(D)$.

7.32. Let H be the digraph in Figure 7.19. Determine $a_2(H)$ and a set of $a_2(H)$ arcs whose addition to H results in a 2-strong digraph. Use one-way pairs to verify optimality.

7.33. Let D be a digraph with $\kappa(D) = k$ and suppose that $\gamma^*_{k+1}(D) = 2k + 1$. Prove that $a_{k+1}(D) = \gamma^*_{k+1}(D)$.

7.34. Let D be the digraph illustrated in Figure 7.9. Prove that $r_2(D) \geq 2$.

7.35. Generalize the example in Figure 7.9 to obtain a set of digraphs $\mathcal{D} = \{D_1, D_2, \ldots, \}$ such that $r_k(D_k) > a_k(D_k)$, $k = 1, 2, \ldots$.

7.36. **Vertices with high in- and out-degree in semicomplete digraphs.** Prove that every semicomplete digraph on at least $4k - 1$ vertices has a vertex x with $d^+(x), d^-(x) \geq k$. Show that this is the best possible.

7.37. **Minimal *k*-out-critical sets are strongly connected**. Prove that, if D is a directed multigraph and X is a minimal k-out-critical set, then the directed multigraph $D\langle X\rangle$ is strongly connected.

7.38. **Removing a minimal separating set from a locally semicomplete digraph**. Prove Lemma 7.13.4.

7.39. **Deriving Theorem 7.6.3 from Theorem 7.8.1**. Show that Theorem 7.6.3 follows from Theorem 7.8.1. Hint: use (7.22) and the two ways of being (S, T)-independent to derive Theorem 7.6.3.

7.40. Let T be the tournament on 7 vertices shown in Figure 7.20. Show that $r_2(T) = 1$ and that $r_2(T - v) = 3$.

7.41. Derive Corollary 7.11.4 from Theorem 7.11.2.

7.42. **Semicomplete digraphs with a *k*-critical 2-cycle**. Prove that the semicomplete digraph D in Figure 7.17 is k-strong, but that neither $D - xy$ nor $D - yx$ is k-strong.

7.43. **Constructing *k*-(strongly)-connected *k*-regular (di)graphs**. Prove that the rth power of an undirected cycle is $(2r)$-connected. Prove that, if n is even and G is obtained from an even cycle $v_1v_2\ldots v_{2k}v_1$ by taking the rth power

v

X Y

Figure 7.20 A strong tournament T on 7 vertices. The fat arcs indicate that all arcs between the sets indicated have the directions shown.

and then adding longest diagonals (v_1v_{k+1}, v_2v_{k+2} etc), then G is $(2r+1)$-connected. These graphs are due to Harary [399], see also the book [717, page 202-205] by Thulasiraman and Swamy.

7.44. **Bi-submodularity of the function $h(X,Y)$ on one-way pairs**. Let $D=(V,A)$ be a digraph. Recall that a pair (X,Y), where $X,Y \subset V$, is a one-way pair if there are no edges from X to Y and that $h(X,Y)$ is defined by $h(X,Y) := |V-(X \cup Y)|$. Prove that the function $h(X,Y)$ is **bi-submodular**, i.e. for every choice of one-way pairs $(X,Y),(X',Y')$ the following holds:

$$h(X,Y)+h(X',Y') \geq h(X \cup X', Y \cap Y') + h(X \cap X', Y \cup Y').$$

Hint: consider the contribution of a vertex $v \in V$ to each side of the inequality.

7.45. Let D be a digraph that is k-strong but not $(k+1)$-strong. Call a one-way pair (X,Y) *critical* if $h(X,Y)$=k. By Lemma 7.7.2 the family

$$\mathcal{F} = \{(X,Y) : (X,Y) \text{ is a critical one-way pair}\}$$

is non-empty. Prove that $\mathcal{F}$ is a crossing family of pairs of sets, i.e. if $(X,Y),(X',Y') \in \mathcal{F}$ satisfy $X \cap X' \neq \emptyset$ and $Y \cap Y' \neq \emptyset$, then $(X \cup X', Y \cap Y'),(X \cap X', Y \cup Y') \in \mathcal{F}$. Hint: use Exercise 7.44.

7.46. **Large 3-strong tournaments with every vertex critical**. Prove that every tournament in the class $\mathcal{T}$ from Figure 7.13 is 3-strong and that every vertex different from x,y is critical.

7.47. **Finding subgraphs with specified bounds on degrees**. Describe a polynomial algorithm which takes as input a digraph $D=(V,A)$ on n vertices and non-negative integers $a_1,a_2,\ldots,a_n$, $b_1,b_2,\ldots,b_n$ such that $d^+_D(v_i) \geq a_i$ and $d^-_D(v_i) \geq b_i$ for $i=1,2,\ldots,n$ and returns a minimum cardinality subset A' of A such that the digraph $D'=(V,A')$ satisfies that $d^+_{D'}(v_i) \geq a_i$ and $d^-_{D'}(v_i) \geq b_i$ for $i=1,2,\ldots,n$. Hint: use flows and use a similar network to that used in the proof of Theorem 3.11.5.

7.48. Prove that if a digraph $D=(V,A)$ contains no anti-directed trail, then $|A| \leq 2|V|-1$. Hint: consider the bipartite representation $BG(D)$ of D and show that this has no cycle.

7.49. (+) Show that for every p with $1 < p < 2$ there exists a weighted digraph $D = D(p)$ for which the weight of $D' = (V, A(F_r^-) \cup A(F_r^+))$, where F_r^- (F_r^+) is a minimum cost in-branching (out-branching) rooted at r in D is at least p times the weight of a minimum cost strong spanning subdigraph of D.

7.50. (−) Let D be a k-arc-strong semicomplete digraph on at least $2k+2$ vertices. Prove that there exists an arc a of D such that $D - a$ is k-arc-strong. Hint: prove that D cannot be minimally k-arc-strong.

7.51. (−) Describe a polynomial algorithm which given a directed multigraph D decides whether $\lambda(D) = \delta^0(D)$.

8. Orientations of Graphs

The purpose of this chapter is to discuss various aspects of orientations of (multi)graphs. There are many ways of looking at such questions. We can ask which graphs can be oriented as a digraph of a certain type (e.g. a locally semicomplete digraph). We can try to obtain orientations containing no directed cycles of even length, or no long paths. We can try to relate certain parameters of a graph to the family of all orientations of this graph (e.g. what does high chromatic number imply for orientations of a graph). We can also look for conditions which guarantee orientations with high arc-strong connectivity or high in-degree at every vertex, etc. There are hundreds of papers dealing with orientations of graphs in one way or another and we can only cover some of these topics. Hence we have chosen some of those mentioned above. Finally we also study briefly the theory of submodular flows which generalizes standard flows in networks and turns out to be a very useful tool (not only theoretically, but also algorithmically) for certain types of connectivity questions as well as orientation problems. We illustrate this by applying the submodular flow techniques to questions about orientations of mixed graphs as well as to give short proofs of the Lucchesi-Younger Theorem and Nash-Williams' orientation theorem. We recall that n and m usually stand for the number of vertices and arcs (edges) of the (di)graph in question.

8.1 Underlying Graphs of Various Classes of Digraphs

In this section we discuss the underlying undirected graphs of several generalizations of tournaments. As can be seen, these include classes of undirected graphs that are very interesting in practical applications such as comparability graphs, proper circular arc graphs and chordal graphs. For much more information about these classes and their relations to each other, the reader is encouraged to consult the books [133] by Brandstädt, [331] by Golumbic, and [613] by Prisner. Here we will just define those classes that we need. A graph G is a **circular arc graph** if there exists a family of circular arcs indexed by the vertices of the graph such that two vertices are adjacent if and only if the two corresponding arcs intersect. This family of circular arcs form a **representation** of G. A **proper circular arc graph** is a circular arc graph which has a representation by circular arcs, none of which is properly contained in another. A graph G is **chordal** if every cycle of length at least 4 has a chord, that is, G has no induced cycle of length four or more. Finally, G is a **comparability graph** if it has a transitive orientation (that is, there exists a transitive oriented graph T such that $UG(T)$ is isomorphic to G).

We will always use Δ to denote the maximum degree of the undirected graph in question.

8.1.1 Underlying Graphs of Transitive and Quasi-Transitive Digraphs

Since every transitive digraph is also quasi-transitive, every comparability graph has a quasi-transitive orientation. The next theorem by Ghouila-Houri shows that the other direction also holds.

Theorem 8.1.1 *[316] A graph G has a quasi-transitive orientation if and only if it has a transitive orientation.*

Proof: To illustrate the usefulness of the decomposition theorem for quasi-transitive digraphs (Theorem 4.8.5), we give a proof which is quite different from the one in [316]. We prove the non-trivial part of the statement by induction on the number of vertices. The claim is easily verified when $n \leq 3$ so we proceed to the induction step, assuming $n \geq 4$. Suppose D is a quasi-transitive orientation of G and that D is not transitive. If D is not strongly connected then it follows from Theorem 4.8.5 that we can decompose D as $D = T[W_1, W_2, \ldots, W_t]$, $t = |V(T)| \geq 2$, where T is transitive and each W_i is a strong quasi-transitive digraph. As $t \geq 2$ it follows by induction that we can reorient each $UG(W_i)$ as a transitive digraph T_i, $i = 1, 2, \ldots, t$. This gives a transitive orientation $D' = T[T_1, T_2, \ldots, T_t]$ of G.

Suppose now that D is strong. By Theorem 4.8.5, D can be decomposed as $D = S[W_1, W_2, \ldots, W_s]$, $s = |V(S)| \geq 2$, where S is a strong semicomplete digraph and each W_i is either a single vertex or a non-strong quasi-transitive digraph. It follows by induction (as above) that we can orient each $UG(W_i)$ as a transitive digraph T'_i, $i = 1, 2, \ldots, s$. Let TT_s be the transitive tournament on s vertices. Then $D' = TT_s[T'_1, T'_2, \ldots, T'_s]$ is a transitive orientation of G. $\square$

The following construction is due to Ghouila-Houri [316]. Let $G = (V, E)$ be an undirected graph. Construct a graph G_{qtd} from G as follows: $V(G_{qtd}) = \bigcup_{uv \in E(G)} \{x_{uv}, x_{vu}\}$ and there is an edge from x_{uv} to x_{wz} precisely if $w = v$ and $uz \notin E$, or $u = z$ and $vw \notin E$. In particular there is an edge $x_{uv}x_{vu}$ for each $uv \in E$. See Figure 8.1 for an illustration of this construction. Note that, if $x_{uv}x_{vw}$ is an edge of G_{qtd}, then so is $x_{wv}x_{vu}$. Every edge of G_{qtd} corresponds to a forbidden pair of oriented edges of G. The interest in this construction lies in the following very useful fact.

Theorem 8.1.2 *[316] A graph G is a comparability graph (and hence has a transitive orientation) if and only if G_{qtd} is bipartite.*

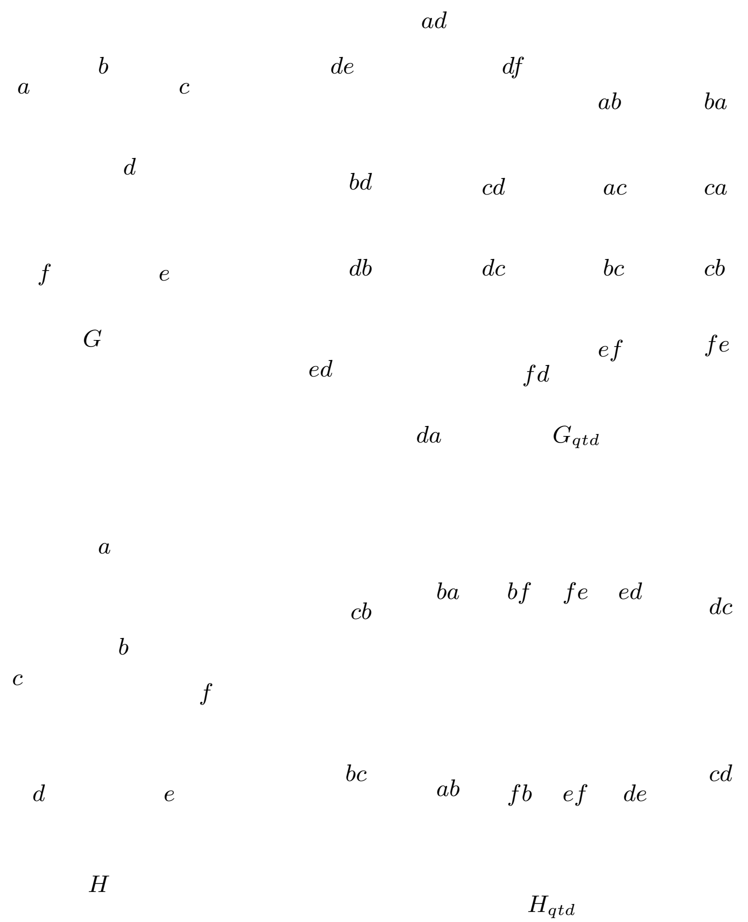


Figure 8.1 An illustration of the construction of G_{qtd} for two graphs. Due to space considerations we have dropped the x's in the name of the vertices of G_{qtd}, H_{qtd}. The graph G is a comparability graph. The graph H is not a comparability graph. Note that bf, cb, dc, ed, fe, bf is a 5-cycle in H_{qtd}.

Proof: Suppose $G = (V, E)$ is a comparability graph and let $T = (V, A)$ be a transitive orientation of G. In G_{qtd} the vertices X_1 corresponding to the arcs of T (that particular orientation of the edge uv for each $uv \in E$) form an independent set. By symmetry of the definition of the edges of G_{qtd}, the remaining vertices X_2 of G_{qtd} also induce an independent set. Hence G_{qtd} is bipartite with bipartition (X_1, X_2).

Conversely, suppose that G_{qtd} is bipartite with bipartition (X, Y). Because G_{qtd} contains a perfect matching consisting of edges of the form $x_{uv}x_{vu}$ it follows that $|X| = |Y|$ and X contains precisely one of the vertices x_{uv}, x_{vu} for each $uv \in E$. It follows from the definition of G_{qtd} that orienting the edges corresponding to the vertices in X (Y) results in a quasi-transitive orientation D of G. (If $x_{uv} \in X$, then orient uv from u to v, otherwise orient it from v to u.) By Theorem 8.1.1, G has a transitive orientation. $\square$

Corollary 8.1.3 *Comparability graphs can be recognized in time $O(\Delta m)$, where m is the number of edges in the input graph.*

Proof: This follows from Theorem 8.1.2 and the fact that the number of edges in G_{qtd} is $O(\Delta|E|)$. Note that we can check whether a given undirected graph is bipartite in linear time using BFS (Exercise 8.2). $\square$

For various results on recognition of comparability graphs see the papers [330] by Golumbic, [411] by Hell and Huang, [574] by Morvan and Viennot and [577] by Muller and Spinrad.

Consider the comparability graph G in Figure 8.1 and suppose that our goal is to obtain a quasi-transitive orientation of G. If we choose the orientation $a{\rightarrow}d$, then this forces the edge between d and e to be oriented as $e{\rightarrow}d$. This in turn forces the orientations $c{\rightarrow}d$ and $b{\rightarrow}d$ and each of these force $f{\rightarrow}d$. Similarly it can be seen that the five edges ad, bd, cd, de, df force each other. It is easy to see that the corresponding ten vertices in G_{qtd} form one connected component of G_{qtd}.

It is not difficult to see that this observation holds for arbitrary comparability graphs, i.e. if x_{uv} and x_{wz} are in the same connected component of G_{qtd} and $wz \neq vu$, then once we decide on an orientation for the edge uv in G, that orientation forces one on the edge wz. An **implication class for $\boldsymbol{G = (V, E)}$** is a maximal set of edges E' with the property that in every orientation of G as a quasi-transitive digraph the choice of an orientation of one edge $e \in E'$ forces the orientation of all other edges in E'.

By our remark above the implication classes for G coincide with the connected components of G_{qtd}. More precisely the connected component C of G_{qtd} corresponds to the implication class $E' = \{uv \in E : x_{uv} \in V(C)\}$. It is not difficult to see that the implication classes form a partition of E. Given G_{qtd} we can obtain the implication classes of G just by finding the connected components of G_{qtd}. Hence we can find the implication classes in time $O(\Delta m)$ (recall that G_{qtd} has $O(\Delta m)$ edges).

Let G be a comparability graph and suppose we want to find a transitive orientation of G. We can obtain a quasi-transitive orientation just by picking an arbitrary edge from each implication class, choosing an orientation for this edge and then orient the remaining edges in that class the way they are forced to be oriented. The problem is that this orientation will in general not be transitive. Consider for example the graph G in Figure 8.1. Since each of the edges ab, bc and ac form an implication class of size one, there is nothing that prevents us from orienting these three edges as the 3-cycle $a{\rightarrow}b{\rightarrow}c{\rightarrow}a$.

We now describe a simple and very useful technique, due to Hell and Huang [411], for obtaining a transitive orientation of a given comparability graph G. Let $1, 2, \ldots, n$ be a fixed labelling of the vertices of G. We say that a vertex x_{ij} of G_{qtd} is **lexicographically smaller** than a vertex x_{rs} if either $i < r$ or $i = r$ and $j < s$.

The **lexicographic 2-colouring** of G_{qtd} is the unique 2-colouring (on colours A, B) which is obtained as follows. Mark all vertices of G_{qtd} non-coloured. Next, as long as there are uncoloured vertices, choose the lexicographically smallest vertex x_{ij} which is not coloured yet and colour it A. Colour all other vertices in the same connected component as they are forced (that is, by A if the distance from x_{ij} is even and by B otherwise). When all vertices of G_{qtd} are coloured the process stops.

The usefulness of lexicographic 2-colourings comes from the following result (see also Theorem 8.1.9).

Theorem 8.1.4 *[411] Let G be a comparability graph with vertices $1, 2, \ldots, n$ and let $f : V(G_{qtd}) \rightarrow \{A, B\}$ be the lexicographic 2-colouring of $V(G_{qtd})$. Define an orientation D of G such that an edge ij is oriented as $i \rightarrow j$ precisely when x_{ij} receives colour A by the colouring f. Then D is a transitive orientation of G.*

Proof: Exercise 8.4. □

Note that, if we apply the lexicographic 2-colouring procedure to a non-comparability graph, then this will be discovered after G_{qtd} has been formed when we try to 2-colour a non-bipartite connected component H of G_{qtd}. The algorithm will discover that H is not bipartite and hence G does not have any orientation as a quasi-transitive digraph. Thus we have obtained another proof of Theorem 8.1.1 (the lexicographic 2-colouring algorithm either finds a transitive orientation of G, or concludes that G has no quasi-transitive orientation).

The whole algorithm (including the construction of G_{qtd}) can be performed in time $O(\Delta m)$, where m is the number of edges of G, since we can find the connected components of G_{qtd} using BFS.

8.1.2 Underlying Graphs of Locally Semicomplete Digraphs

For a given proper circular-arc graph G with a prescribed circular-arc representation we get a natural order on the vertices of G by fixing a point on the circle and labeling the vertices $v_1, v_2, \ldots, v_n$ according to the clockwise ordering of the right endpoints of their intervals (circular arcs) on the circle with respect to this point. Since every proper circular-arc graph has a representation in which no two arcs cover the whole circle [331], we may assume that we are working with such a representation. Now it is not difficult to see that the following process leads to a round local tournament orientation of G (see Chapter 4 for the definition of a round local tournament[1]): orient

[1] Hell and Huang use the name **local transitive tournament** instead of round local tournament [411].

the edge between v_i and v_j from v_i to v_j just if the left endpoint of the jth interval is contained in the ith interval. Thus we have the following result due to Skrien (see also [44, 410, 436]):

Proposition 8.1.5 *[675] Every proper circular-arc graph has an orientation as a round local tournament.* □

In fact, Hell and Huang showed that the other direction holds as well.

Theorem 8.1.6 *[411] A connected graph is a proper circular arc graph if and only if it is orientable as a round local tournament.*

Proof: We proved one direction above. To prove the other direction assume that D is a round local tournament and that $v_1, v_2, \ldots, v_n$ is a round enumeration of $V(D)$. If no such labelling is given, then we can find one in time $O(n+m)$ (Exercise 8.6). Now represent $UG(D)$ by circular arcs as follows. Let ϵ be a fixed number such that $0 < \epsilon < 1$. Make an n-scale-clock on a cycle and associate with the vertex v_i the circular arc from i to $i + d^+_D(i) + \epsilon$ in the clockwise order for $i = 1, 2, \ldots, n$ (indices modulo n). It is easy to check that this gives a proper circular arc representation of $UG(D)$. Note that here we use the fact that the out-neighbours of every vertex of D induce a transitive tournament (see Chapter 4) to see that no arc is properly contained in any other arc. □

By Theorem 8.1.6, the class of underlying graphs of locally semicomplete digraphs contains the class of proper circular arc graphs. The next result, due to Skrien [675] (see also [410, 436]) says that there are no other graphs that can be oriented as locally semicomplete digraphs.

Theorem 8.1.7 *[675] The underlying graphs of locally semicomplete digraphs are precisely the proper circular arc graphs.* □

Bang-Jensen, Hell and Huang [410] showed that, just as in the case of comparability graphs, there is a useful auxiliary graph G_{ltd} related to orientations as a local tournament digraph: Let $G = (V, E)$ be given and define G_{ltd} as follows: $V(G_{ltd}) = \bigcup_{uv \in E(G)} \{x_{uv}, x_{vu}\}$ and there is an edge from x_{uv} to x_{wz} precisely if $v = z$ and $uw \notin E$, or $u = w$ and $vz \notin E$. Furthermore, the edge $x_{uv}x_{vu}$ is in $E(G_{ltd})$ for each $uv \in E$. The proof of the following result is left as Exercise 8.7.

Theorem 8.1.8 *[410] The graph G has an orientation as a local tournament digraph if and only if the graph G_{ltd} is bipartite.* □

Suppose G is a proper circular arc graph. Then it follows from Theorem 8.1.7 and Theorem 8.1.8 that G_{ltd} is bipartite. Again each connected component of G_{ltd} corresponds to an implication class E' of edges of G. Hence we can find a local tournament orientation of G by fixing the orientation of one arc from each implication class arbitrarily and then giving all remaining arcs the forced orientation.

If our goal is to find a representation of G as a proper circular arc graph, then we are not interested in just any local tournament orientation of G, but we need an orientation as a round local tournament (compare with Theorem 8.1.6). Again we can use the lexicographic method which was defined in Section 8.1.1 for this. Since G_{ltd} is bipartite, we can apply the lexicographic 2-colouring procedure which was defined in Section 8.1.1. It follows from the next theorem and the proof of Theorem 8.1.6 that the lexicographic method is also of use in recognition of proper circular arc graphs.

Theorem 8.1.9 *[411] Let G be a proper circular arc graph and let $f : V(G_{ltd}) \rightarrow \{A, B\}$ be the lexicographic 2-colouring of $V(G_{ltd})$. Define an orientation D of G such that an edge ij is oriented as $i \rightarrow j$ precisely when x_{ij} receives colour A by the colouring f. Then D is a round local tournament orientation of G.* □

This shows that using the lexicographic method one can obtain an $O(\Delta m)$ algorithm for recognizing and representing proper circular arc graphs.

In fact an even faster and optimal algorithm for recognizing proper circular arc graphs has been found by Deng, Hell and Huang [190]. This algorithm also uses the fact that a graph is a proper circular arc graph if and only if it has an orientation as a round local tournament.

Theorem 8.1.10 *[190] There is an $O(n+m)$ algorithm to find a local tournament orientation of a graph G or to report that G does not admit such an orientation. Moreover, if a local tournament orientation exists, the algorithm also identifies all balanced arcs.* □

We will define the notion of a balanced arc in the next subsection.

8.1.3 Local Tournament Orientations of Proper Circular Arc Graphs

In this subsection we describe a deep result by Huang [435, 436] which gives a complete characterization of all the possible local tournament orientations of a given proper circular arc graph. In order to state Theorem 8.1.12 below we need several definitions.

Let $G = (V, E)$ be an undirected graph. An edge xy of G is **balanced** if every vertex $z \in V - \{x, y\}$ is adjacent to both or none of x and y. An edge is **unbalanced** if it is not balanced. If all edges of G are unbalanced, then G is **reduced** and otherwise G is **reducible**. It follows from this definition that a graph which is not reduced can be decomposed as described in the next lemma. See Figure 8.2 for an illustration.

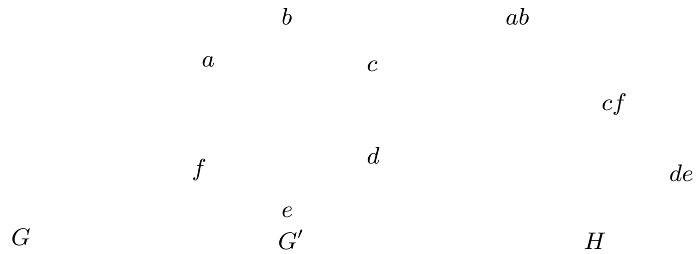


Figure 8.2 A reduced graph G and a reducible graph G'. The graph G' can be reduced to the graph H by identifying the pairs $\{a, b\}$, $\{c, f\}$ and $\{d, e\}$.

Lemma 8.1.11 *If G is not a reduced graph, i.e. it has a balanced edge, then there exist a reduced subgraph H of G and complete subgraphs $K_{a_1}, K_{a_2}, \ldots, K_{a_h}$ of G such that $G = H[K_{a_1}, K_{a_2}, \ldots, K_{a_h}]$, $h = |V(H)|$* [2]*. Furthermore we can find this (unique) decomposition in time $O(n^3)$.*

Proof: We leave the easy proof to the reader. □

Actually such a decomposition can be found even faster in $O(n^2)$ time, see the paper [217] by Ehrenfeucht, Gabow, McConnell, and Sullivan.

Let $G = (V, E)$ be a proper circular arc graph. As we mentioned in the last subsection one can partition E into disjoint non-empty subsets $E_1, \ldots, E_r$ with the property that, if we fix the orientation of one edge in each E_i, then there is precisely one way to orient all the remaining edges in E so that the resulting digraph is a local tournament digraph. In other words, the orientation of one edge in E_i implies the orientation of all other edges in E_i. As in the last section we call the sets $E_1, \ldots, E_r$ the implication classes of G (see Theorem 8.1.12 and Theorem 8.1.13 below).

Theorem 8.1.12 *[436, Huang] Let G be a connected proper circular arc graph and let $C_1, \ldots, C_k$ be the connected components of $\overline{G}$. Then one of the following two statements holds.*

(a) $\overline{G}$ is bipartite, the set of all unbalanced edges of G with both ends in a fixed C_i form an implication class and the set of all unbalanced edges of G between two distinct C_i and C_j form an implication class (see Figure 8.3).
(b) $\overline{G}$ is not bipartite, $k = 1$, and all unbalanced edges of G form one implication class. □

Observe that an edge forms an implication class by itself if and only if it is balanced. Hence Theorem 8.1.12 can be reformulated as follows.

[2] Here the **composition** $H[G_1, G_2, \ldots, G_{|V(H)|}]$ is defined analogously to the composition of digraphs in Section 1.3.

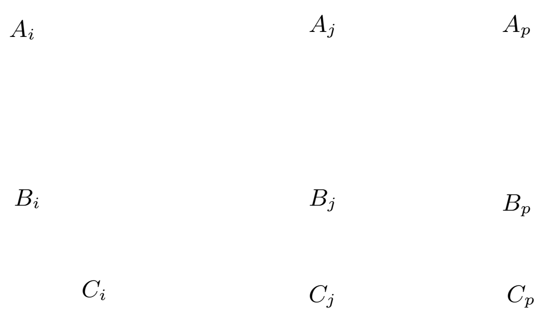


Figure 8.3 Implication classes for orientations of a graph G as a local tournament digraph.The sets C_i, C_j, C_p denote distinct connected components of $\overline{G}$. For each component a bipartition A_r, B_r is shown. The edges shown inside C_i form one implication class and the edges shown between C_j and C_p form another implication class.

Theorem 8.1.13 (Huang) *[436] Let G be a proper circular-arc graph which is reduced (that is, every edge is unbalanced), let $\bar{G}$ denote the complement graph of G and let $C_1, \ldots, C_k$ denote the connected components of $\bar{G}$.*

(a) If $\bar{G}$ is not bipartite, then $k = 1$ and (up to a full reversal) G has only one orientation as a locally tournament digraph, namely the round orientation.
(b) If $\bar{G}$ is bipartite then every orientation of G as a locally tournament digraph can be obtained from the round locally tournament digraph orientation D of G by repeatedly applying one of the following operations:
  (I) reverse all arcs in D that go between two different C_i's,
  (II) reverse all arcs in D that have both ends inside some C_i. □

It is also possible to derive a similar result characterizing all possible orientations of G as a locally semicomplete digraph. We refer the reader to [436] for the details.

As an example of the power of Huang's result (Theorems 8.1.12 and 8.1.13) we state and prove the following corollary which was implicitly stated in [436] (see also Exercise 4.33).

Corollary 8.1.14 *If D is a locally tournament digraph such that $\overline{UG(D)}$ is not bipartite, then $D = R[S_1, \ldots, S_r]$, where R is a round locally tournament digraph on r vertices and each S_i is a strong tournament.*

Proof: If $UG(D)$ is reduced, then this follows immediately from Theorem 8.1.13, because according to Theorem 8.1.13, there is only one possible locally tournament digraph orientation of $UG(D)$. So suppose that $UG(D)$ is not reduced. By Lemma 8.1.11, $UG(D) = H[K_{a_1}, \ldots, K_{a_h}]$, $h = |V(H)|$, where H is a reduced proper circular arc graph, each K_{a_i} is a complete graph, and some $a_i \geq 2$. Because we can obtain an isomorphic copy of H as a subgraph of $UG(D)$ by choosing an arbitrary vertex from each K_{a_i}, we conclude, from Theorem 8.1.12, that in D all arcs between two distinct K_{a_i}, K_{a_j} have the same direction (note that $\overline{H}$ is non-bipartite). Thus $D = R[S_1, \ldots, S_r]$, where (up to reversal of all arcs) R is the unique round locally tournament digraph orientation of H and each S_i is the tournament $D\langle V(K_{a_i})\rangle$. Note that $D\langle V(K_{a_i})\rangle$ may not be a strong tournament, but according to Corollary 4.11.7 we can find a round decomposition of D so that this is the case. □

8.1.4 Underlying Graphs of Locally In-Semicomplete Digraphs

The structure of the underlying graphs of locally in-tournament digraphs is more complicated than in the case of local tournaments and quasi-transitive digraphs. In [725] Urrutia and Gavril studied locally in-tournament digraphs under another name **fraternally oriented graphs**. This name, although used in several papers (e.g. [292] by Galeana-Sánchez, [307] by Gavril, [308] by Gavril, Toledano Laredo and de Werra and [309, 725] by Gavril and Urritia), is somewhat misleading, since it may easily be confused with the name **fraternally orientable** which is used for an undirected graph with an orientation as a fraternally oriented graph.

In [725] an algorithm for recognizing graphs orientable as locally in-tournament digraphs (as well as finding a locally in-tournament digraph orientation if one exists) is given. The complexity is $O(nm)$ which is worse than the simple algorithm based on 2-satisfiability given in Proposition 8.1.15 below.

In the paper [725] Urrutia and Gavril also gave a characterization in terms of forbidden subgraphs of graphs orientable as locally in-tournament digraphs. Unfortunately, the characterization is not in terms of minimal forbidden subgraphs. In fact, the characterization is merely a structural observation of what happens when the algorithm of [725] fails to find a locally in-tournament digraph orientation.

In Section 1.10 we mentioned that algorithms for the 2-SAT problem are useful for certain orientation problems. The proposition below gives one example of this.

Proposition 8.1.15 *[81] Graphs that are orientable as locally in-tournament digraphs can be recognized in $O(\Delta m)$ time.*

Proof: Let a graph $G = (V, E)$ be given, and let $D = (V, A)$ be an arbitrary orientation of the edges of G, where $A = \{a_1, a_2, \ldots, a_m\}$. If a_i is an orientation of an edge yz of G, then the reverse orientation of that edge is denoted by $\overleftarrow{a_i}$. We now construct an instance of the 2-SAT problem as follows. The set of variables is $X = \{x_1, \ldots, x_m\}$. The variables are interpreted as follows. If

$x_i = 1$, then we keep the orientation a_i, otherwise we take the opposite orientation $\overleftarrow{a_i}$. The clauses consist of those pairs of literals $(\ell_i + \ell_j)$ for which $\overline{\ell_i}, \overline{\ell_j}$ correspond to arcs with the same terminal vertex and nonadjacent initial vertices in D. It is easy to see that G is orientable as a locally in-tournament digraph if and only if the above-defined instance of 2-SAT is satisfiable. By Theorem 1.10.5 the complexity of 2-SAT is $O(K)$ where K is the number of clauses. Hence, it follows from the way we construct the clauses above that we can recognize graphs orientable as locally in-tournament digraphs in time $O(\Delta m)$. $\square$

The construction used in the proof above is illustrated in Figure 8.4. Part (a) shows an undirected graph G; part (b) an arbitrary orientation D of G. The instance of 2-satisfiability corresponding to this orientation contains one variable for each arc of D and the following clauses:

$$(\bar{x}_{ab} + \bar{x}_{cb}), (\bar{x}_{ad} + \bar{x}_{cd}), (x_{cb} + x_{ce}), (x_{cd} + x_{ce}), (\bar{x}_{ce} + \bar{x}_{fe}),$$
$$(\bar{x}_{ce} + \bar{x}_{he}), (\bar{x}_{fe} + \bar{x}_{he}), (\bar{x}_{fg} + \bar{x}_{hg}), (\bar{x}_{ce} + x_{eg}).$$

Part (c) shows an orientation of G as an in-tournament digraph corresponding to the truth assignment $(x_{ab}, x_{ad}, x_{cb}, x_{cd}, x_{ce}, x_{db}, x_{eg}, x_{fe}, x_{fg}, x_{he}, x_{hg}) = (0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0)$.

b f c e a g d h
(a)

b f c e a g d h
(b)

b f c e a g d h
(c)

Figure 8.4 An undirected graph G and two orientations of G.

In Exercise 1.68 a useful correspondence between the 2-SAT problem and the problem of deciding the existence of an independent set of size $n/2$ in graphs with a perfect matching was indicated. Using this correspondence, it is no surprise that for graphs which are orientable as in-tournament digraph there is a construction similar to the one used in Theorem 8.1.2 for comparability graphs. (In Theorem 8.1.8 we saw a similar one for the underlying graphs of locally semicomplete digraphs.)

Let $G = (V, E)$ be an undirected graph and define the undirected graph G_{itd} as follows: $V(G_{itd}) = \bigcup_{uv \in E(G)} \{x_{uv}, x_{vu}\}$ and there is an edge from x_{uv} to x_{wz} precisely if $w = v$ and $z = u$, or $v = z$ and $uw \notin E$.

The proof of the following lemma is left to the reader as Exercise 8.1. As mentioned above, it is useful to compare this lemma with Exercise 1.68.

Lemma 8.1.16 *A graph $G = (V, E)$ of order n is orientable as a locally in-tournament digraph if and only if the graph G_{itd} has an independent set of size n.* □

B_1 B_2 B_3

Figure 8.5 The digraphs B_1, B_2, B_3

Let $\mathcal{B}$ be the family of the three digraphs shown in Figure 8.5 and let F be any subset of $\mathcal{B}$ other than $\{B_1\}$ or $\{B_2\}$. Skrien [675] characterized the classes of those graphs which can be oriented without a member of F as an induced subdigraph. These are the classes of complete graphs, comparability graphs, proper circular arc graphs, and nested interval graphs. Since each of the forbidden configurations contains just two arcs, 2-SAT could be used to solve the recognition problem for each of these four classes, all in time $O(\Delta m)$.

The **intersection graph** $\Gamma(F')$ of a family $F' = \{S_x : x \in V\}$ of sets has vertex set V and two distinct vertices x, y are adjacent whenever $S_x \cap S_y \neq \emptyset$. A graph G is **representable** in the graph H if G is isomorphic to the intersection graph of a family of connected subgraphs $\{H_x : x \in V(G)\}$ of H. It seems interesting that three of the four classes above can be defined by representability. In the case of the underlying graphs of locally in-tournament digraphs, we do not know of a similar characterization (see Theorem 8.1.19 for a partial result).

If we consider another kind of representability involving pointed sets, then such a representation does indeed exist. A **pointed set** is a pair (X, p) consisting of a set X and one element $p \in X$. Maehara [542] defines the **catch digraph** $\Omega(F)$ of a family $F = \{(S_x, p_x) : x \in V\}$ of pointed sets as the digraph with vertex set V and an arc from x to y if $p_y \in S_x$, for $x \neq y \in V$. Obviously the underlying graph of $\Omega(\{(S_x, p_x) : x \in V\})$ is a spanning subgraph of $\Gamma(\{S_x : x \in V\})$ for any family of pointed sets. The converse only holds in special cases:

Lemma 8.1.17 *[81, 725] If D is a locally in-semicomplete digraph, then $\Omega(\{(N_D^+[x], x) : x \in V\}) = D$ and $\Gamma(\{N_D^+[x] : x \in V\}) = UG(D)$.*

Proof: The first statement is obvious. Let now x and y be distinct vertices of D such that $N_D^+[x] \cap N_D^+[y] \neq \emptyset$. Then $x \to y$ or $y \to x$ or x and y have some common successor z. In the last case again $x \to y$ or $y \to x$, since D is locally in-semicomplete. Then $UG(D) = \Gamma(\{N_D^+[x] : x \in V\})$ by the remarks above. $\square$

Gavril and Urrutia found the following characterization of locally in-semicomplete digraphs in terms of catch digraphs and representability:

Theorem 8.1.18 *[725] A digraph $D = (V, A)$ is locally in-semicomplete if and only if it is the catch digraph of a family $(\{(S_x, p_x) : x \in V\})$ such that $UG(D)$ equals $\Gamma(\{S_x : x \in V\})$.*

Proof: Let D be the catch digraph of $(\{(S_x, p_x) : x \in V\})$ such that $UG(D)$ is the intersection graph G of $(\{S_x : x \in V\})$. Choose any predecessors x, z of a vertex y. Then $p_y \in S_x \cap S_z$, which implies $xz \in E(G)$. But then $x \to z$ or $z \to x$ in D. The converse follows from Lemma 8.1.17. $\square$

An undirected graph is **unicyclic** if it has precisely one cycle. The next result, due to Prisner, and the corollaries below show that the class of graphs orientable as in-tournament digraphs is quite large.

Theorem 8.1.19 *[612] Every graph G that is representable in a unicyclic graph is orientable as a locally in-tournament digraph.*

Proof: We give a proof due to Bang-Jensen, Huang and Prisner [81]. Let $\{H_x : x \in V(G)\}$ be a representation of G in a unicyclic graph H with cycle $C = z_0, z_1, \ldots, z_{\ell-1}, z_0$. The numbering is done clockwise around the cycle (the reader should think of this as drawn in the plane). We may assume that H is connected. For vertices x of G whose representative H_x contains all vertices of the cycle C, we define $p_x := z_0$. If H_x contains some but not all of the vertices of C, then it contains just a subpath, since H_x is connected. For such vertices x we take p_x as the first vertex of this path in the clockwise orientation. If $H_x \cap C = \emptyset$, then there is a unique vertex of H_x separating the rest of H_x from C and we let p_x be that vertex.

By Theorem 8.1.18, it suffices to show that the catch digraph D of the family $\{(V(H_x), p_x) : x \in V(G)\}$ is an orientation of G. Let xy be an edge of G, that is, $H_x \cap H_y \neq \emptyset$. Let z be a vertex of $H_x \cap H_y$. If $H_x \cap C$ and $H_y \cap C$ are nonempty, then it is easy to see that $p_y \in V(H_x \cap C)$ or $p_x \in V(H_y \cap C)$. Thus $x \to y$ or $y \to x$ in D.

So suppose without loss of generality that $H_x \cap C = \emptyset$. Then there is exactly one path from z to C. The vertex p_x must lie on this path, and if

$H_y \cap C = \emptyset$, then so must the vertex p_y. We may assume without loss of generality that p_x lies on the (p_y, z)-subpath. Now $p_x \in V(H_y)$ and $y \to x$ in D. If $H_y \cap C \neq \emptyset$, then the whole path from z to C must lie inside H_y, whence $y \to x$ in D. $\square$

Figure 8.6 A locally in-tournament digraph whose underlying graph is not representable in a unicyclic graph

The converse is not true: Figure 8.6 shows a locally in-tournament digraph whose underlying graph is not representable in any unicyclic graph. It can be easily shown that in any graph G representable in an unicyclic graph the following must hold. Any vertex x of an induced cycle of length at least 4 must be adjacent to at least one vertex from any other induced cycle in $G - x$ (Exercise 8.12). But this property is certainly not obeyed by the underlying graph of the digraph of Figure 8.6.

A **cactus** is a connected graph in which every block[3] is a cycle or an edge (see Figure 8.7). The following conjecture was stated implicitly by Bang-Jensen, Huang and Prisner in [81].

Conjecture 8.1.20 *[81] Every graph orientable as a locally in-tournament digraph is representable in a cactus.*

Note that the opposite is not true: no cactus with at least two induced cycles of length ≥ 4 can be oriented as a locally in-tournament digraph. This implies the claim since every graph can be represented in some subdivision of itself. Just subdivide each edge once and take the set representing a vertex v as the star at that vertex in the subdivided graph.

Theorem 8.1.19 has several consequences. We list some of them below.

[3] A **block** of an undirected graph G is a maximal connected subgraph without a cut-vertex.

Figure 8.7 A cactus.

Corollary 8.1.21 *Every chordal graph and every circular arc graph is orientable as a locally in-tournament digraph.*

Proof: Chordal graphs are representable in trees (see [331, page 82]) and hence in unicyclic graphs. By definition, every circular arc graph is representable in some unicyclic graph. Now the claim follows from Theorem 8.1.19. □

Another (non-trivial) corollary is the following by Bang-Jensen, Huang and Prisner. For a proof see [81].

Corollary 8.1.22 *[81] Every graph with exactly one induced cycle of length greater than 3 is orientable as a locally in-tournament digraph.* □

We close this subsection with a characterization, due to Bang-Jensen, Huang and Prisner, of those line graphs which are orientable as locally in-tournament digraphs (a graph G is a **line graph** if there exists an undirected graph H such that G is the intersection graph of the edges in H (considered as subsets of $V(H)$ of size two). We write $G = L(H)$ if G is the line graph of H.

Let $C = x_0, x_1, \ldots, x_{\ell-1}, x_0$ be a cycle. A chord $x_i x_j$ with $i, j \in \{0, 1, \ldots, \ell-1\}, i < j$, is called a ***p*-chord** for $p = min\{j-i, i+\ell-j\}$. Two chords $x_i x_j$ and $x_k x_m$ are **crossing** if without loss of generality$0 \leq i < k < j < m \leq \ell - 1$.

Theorem 8.1.23 *[81] For any connected graph G, the following are equivalent:*

(i) $L(G)$ is orientable as a locally in-tournament digraph,

(ii) With at most one exception, every block of G is K_2 or K_3, and the exception is either K_4 or a cycle with (possibly) non-crossing 2-chords. □

8.2 Fast Recognition of Locally Semicomplete Digraphs

In this section we study the recognition of locally semicomplete digraphs. We show how to obtain an $O(n^2)$ algorithm using the structural characterizations of Theorems 8.1.12 and 8.1.13 as well as the linear algorithm of Theorem 8.1.10 for recognizing and representing proper circular arc graphs. We include this section to show another application of the main results from Subsection 8.1.3.

We will concentrate on local tournaments, but the results can be extended to general locally semicomplete digraphs (see Exercise 8.10 and [76]).

It is easy to see that local tournaments can be recognized in polynomial time. Given an oriented graph D, to test whether D is a local tournament it is enough to verify the following property for each arc (x, y) of D: the vertex x must be adjacent to every vertex which dominates y and the vertex y must be adjacent to every vertex which is dominated by x. If the property is satisfied for each arc of D, then D is a local tournament; otherwise D is not a local tournament. It is easy to see that this verification can be done in $O(\Delta(D)m)$ time, where $\Delta(D)$ is the maximum degree of D.

Below we shall show how to obtain an $O(n^2)$ algorithm to recognize local tournaments. The description follows [76]. We point out that in [76] it was claimed that the algorithm is linear. This is not quite true, since we use the transformation to the complement graph as an important subroutine and the size of the complement graph is generally not linear in the size of the original graph.

In our algorithms, we assume that digraphs D are represented by the lists of in-neighbours and out-neighbours. This allows us to get all in- and out-neighbours of a vertex v with $O(d(v))$ operations, where $d(v)$ is the degree of v in $UG(D)$. We also need additional information suitable to decide, in time $O(1)$, whether, for given vertices u and v, it is the case that u dominates v, or v dominates u, or neither. It is possible to obtain, in time $O(m)$, a version of the adjacency matrix of D (with valid entries certified by the means of an additional stack), which allows us to do this, cf. [6, Exercise 2.12]. If we ignore the complexity of taking the complement graph, then this representation is needed to give a linear algorithm.

Suppose that D is a local tournament. If D' is obtained from D by reversing a balanced arc[4], then D' is also a local tournament. Thus we can arbitrarily and independently reverse any balanced arc and still have a local tournament. We can also reverse unbalanced arcs, in suitable combinations. Let $C_1, C_2, \ldots, C_k$ be the components of $\overline{UG(D)}$. We define a **partial reversal** of D to be an operation which reverses all unbalanced arcs within some C_i, or reverses all unbalanced arcs between two fixed C_i and C_j. Partial reversals also preserve the property of being a local tournament. This follows from Theorem 8.1.13, but to see it directly, suppose that D'' is obtained from D by performing a partial reversal. If D'' is not a local tournament, then D'' contains three vertices x, y, z such that y and z are two non-adjacent in-neighbours or out-neighbours of x. Assume that y and z are non-adjacent in-neighbours of x. (A similar discussion applies when y and z

[4] An arc is **balanced** if the corresponding edge is balanced in $UG(D)$.

are non-adjacent out-neighbours of x.) Note that both (y,x) and (z,x) are unbalanced and y and z are in the same component of $\overline{G(D)}$. Thus a partial reversal either reverses both or neither of the arcs (y,x) and (z,x). Hence D contains either both (y,x) and (z,x) or both (x,y) and (x,z), contradicting the fact that D is a local tournament.

It follows from Theorem 8.1.13 that the two operations described above are sufficient to obtain all local tournament orientations of a fixed proper circular arc graph G, starting from any one fixed local tournament orientation of G.

Our strategy to obtain an $O(n^2)$ algorithm to recognize local tournaments combines Theorems 8.1.13 and 8.1.10.

Suppose that D is the input oriented graph. We assume that D is connected as otherwise we can consider each component of D separately. Let $G = UG(D)$. By Theorem 8.1.7, G is a proper circular arc graph if and only if it can be oriented as a local tournament. Thus we first test whether G admits a local tournament orientation. This can be done in time $O(n+m)$, by Theorem 8.1.10. If G does not admit a local tournament orientation, then we simply report that D is not a local tournament. Otherwise the algorithm of Theorem 8.1.10 finds a local tournament orientation D' of G, and identifies all balanced arcs.

We mark an edge by T if it has the same direction in both D and D' and by F if it has opposite orientations in D and D'. By the earlier observation that balanced arcs can be reversed arbitrarily, we only need to check the $T-F$ assignment for the unbalanced edges of G. If there are two unbalanced edges of G in one component or between two fixed components of $\overline{G}$, such that one is marked by T and the other is marked by F, then D is a not local tournament by Theorem 8.1.12. Otherwise D is a local tournament because, according to the observations made above, D can be obtained from D' by performing partial reversals and changing directions of some balanced arcs. It is easy to see that the above verifications can be implemented in time $O(n^2)$ (again we stress that the only reason why the algorithm is not of complexity $O(n+m)$ is that we need to find the connected components of the complement graph). Summarizing, we have the following algorithm:

Local tournament recognition algorithm
Input: An oriented graph D.
Output 'yes' if D is a local tournament digraph and 'no' otherwise.

1. If the underlying graph G of D does not admit a local tournament orientation, then D is not a local tournament. Return the answer 'no'.
2. Find a local tournament orientation D' of G.
3. For every edge e of $UG(D)$ mark e by T if it has the same orientation in both D and D' and by F if it has opposite orientations in D and D'.
4. Construct the complement graph $\overline{UG(D)}$ of $UG(D)$ and find the connected components of $\overline{UG(D)}$.

6. Find the set of unbalanced edges in $UG(D)$.
7. If there are two unbalanced edges e, e' of G such that e is marked by T and e' is marked by F and end vertices of e, e' are all within one component or both edges go between two fixed components of $\overline{UG(D)}$, then D is not a local tournament. Return the answer 'no'.
8. Otherwise D is a local tournament. Return the answer 'yes'

In the case of digraphs that are not strongly connected one can obtain a simpler $O(n+m)$ algorithm to decide whether the given digraph is locally semicomplete based on Theorem 4.11.6. We leave this as an exercise (Exercise 8.11), see also [76].

8.3 Orientations With no Even Cycles

It can be seen from Section 10.6 that the problem of deciding whether a given digraph has an even cycle is polynomially solvable, but very complicated. The corresponding problem for undirected graphs is easy (see Exercise 8.17). Here we will consider a somewhat opposite orientation problem where we wish to achieve orientations with no even cycles. Since we can concentrate on strong components when looking for even cycles, we only consider strong orientations without even cycles. Clearly we can also concentrate on graphs that are non-bipartite since otherwise every cycle will be even and the answer is trivial. It is also clear that it suffices to consider graphs which are 2-connected.

Let G be an undirected graph and let us call an orientation D of G **odd** if there is no directed cycle of even length in D. The following problem was posed by Bang-Jensen in 1992 (see e.g. [313]).

Problem 8.3.1 *Is there a polynomial algorithm which given an undirected graph G either returns a strong odd orientation D of G or a proof (in the form of a certificate that can be checked in polynomial time) that G has no such orientation?*

This seems to be a very hard problem and so far only a partial answer (Theorem 8.3.3 below) is known. In order to state Theorem 8.3.3, we need the following definitions. An **odd-K_4** is an undirected graph which is a subdivision of the complete graph on four vertices in which each of the four 3-cycles of K_4 become odd cycles (see Figure 8.8 (a)). An **odd necklace** is any undirected graph which can be obtained from an odd number t of odd cycles $C_1, C_2, \dots, C_t$ by identifying one vertex of C_i with one vertex of C_{i+1} (modulo t) in such a way that $|V(C_i) \cap V(C_j)| = 1$ if $|i-j| = 1 \pmod k$ and $|V(C_i) \cap V(C_j)| = 0$ otherwise (see Figure 8.8(b)).

The proof of the following lemma is left as Exercise 8.16.

Lemma 8.3.2 *[313] Let G be a graph which is either an odd-K_4, or an odd necklace. Then every strong orientation of G has an even cycle.* $\square$

odd

odd odd

odd odd

odd

odd odd

(a) (b)

Figure 8.8 Illustration of an odd-K_4 and an odd necklace. Each of the six dashed lines in the odd-K_4 in part (a) correspond to internally disjoint paths and the word *odd* inside a cycle in part (b) indicates that the length of the bounding cycle is odd

However graphs that contain odd-K_4's may have strong odd orientations as shown in Figure 8.9. Note that in this orientation the 2-connected subgraph corresponding to the odd-K_4 is not oriented as a strong digraph.

Figure 8.9 A strong odd orientation of a graph with an odd-K_4 (shown as fat arcs).

Gerards and Shepherd proved the following result:

Theorem 8.3.3 *[313] Let G be 2-connected and non-bipartite. If G contains neither an odd-K_4 nor an odd necklace as a subgraph, then G has a strong odd orientation.* $\square$

By Lemma 8.3.2, Theorem 8.3.3 can be reformulated as

Theorem 8.3.4 *[313] Let G be an undirected graph. Then each 2-connected non-bipartite subgraph of G has a strong odd orientation if and only if G contains neither an odd-K_4 nor an odd necklace as a subgraph.* $\square$

The proof of Theorem 8.3.3 is based on a constructive characterization of graphs with no odd-K_4's and no odd necklaces [313, Theorem 7, Corollary 8] (see also [311]).

It is shown in [313] that graphs which contain no odd-K_4 and no odd necklace can be recognized in polynomial time. Furthermore the proof of Theorem 8.3.3 in [313] is constructive and implies that there is a polynomial algorithm for Problem 8.3.1 for graphs with no odd-K_4 and no odd chain.

For further results on orientations of graphs with no odd-K_4 see the papers [310, 312] by Gerards.

How many edges can a graph G have before every strong orientation of G has an even cycle? Since every strong orientation of a complete graph on n vertices is pancyclic by Theorem 1.5.1 it is clear that there is some upper bound on the number of edges (as a function on n) for graphs which have strong orientations without even cycles.

Let A and B be disjoint sets of size $\lfloor (n-1)/2 \rfloor$ and $\lceil (n-1)/2 \rceil$ respectively. Form a graph H_n by taking $V(H_n) = A \cup B \cup v$, where v is a new vertex and $E(H_n) = \{ab : a \in A, b \in B\} \cup \{vc : c \in A \cup B\}$. Then $|E(H_n)| = \lfloor (n+1)^2/4 \rfloor - 1$ and we can orient H_n so that it is strong and all cycles are 3-cycles just by orienting all arcs from v to A, from A to B and from B to v.

Let $C_n = v_0 v_1 \dots v_{n-2} v_{n-1} v_0$ be a cycle. Let L_n be obtained from C_n by adding all chords $v_i v_j$ such that $i - j$ is a positive even number. It is not difficult to check that each graph L_n has $|E(L_n)| = \lfloor (n+1)^2/4 \rfloor - 1$ and that L_n has a strong orientation with no even cycles (Exercise 8.18).

These two classes show that the following result, due to Chung, Goddard and Kleitman, is best possible in terms of the number of edges. We formulate it as a theorem for oriented graphs.

Theorem 8.3.5 *[158] Every strong oriented graph for which the number of arcs is at least $\lfloor (n+1)^2/4 \rfloor = f(n) + 1$ contains an even cycle. Furthermore every strong oriented graph D with $f(n)$ arcs which has no even cycle consists of a maximal hamiltonian arc-critical subdigraph H of D on an odd number ($2r+1$, for some r) of vertices and an acyclic bipartite tournament B on the remaining vertices, having the partite sets as equal in size as possible, each vertex of which is joined to $r+1$ vertices of H.* □

By a maximal hamiltonian arc-critical subdigraph of D we mean a subdigraph on, say, n' vertices which has $f(n')$ edges, is hamiltonian, and is maximal with respect to these conditions (that is, every subdigraph of D with $n'' > n'$ vertices is either non-hamiltonian or has less than $f(n'')$ arcs).

Although Theorems 8.3.3 and 8.3.5 do give some information as to which graphs have strong orientations without even cycles, there are large classes of graphs for which they give no information. One such class is the cubic graphs one can obtain by joining two odd cycles of the same length by a perfect matching. The Petersen graph[5] is one of these graphs. It is easy to see that the Petersen graph (an orientation of which is shown in Figure 8.10) contains

[5] The Petersen graph, due to the Danish pioneer of graph theory, Julius Petersen (1839-1910), is very important in several problems on undirected graphs (see e.g. [735]).

an odd-K_4 and hence is not covered by Theorem 8.3.3. In Exercise 8.15 the reader is asked to prove that every strong orientation of the Petersen graph contains an even cycle.

Obviously an oriented graph has an even cycle if it has two cycles whose length differ by one. Hence the following problem may be interesting to study. The analogous problem was considered for undirected graphs by Bondy and Vince in [129].

Problem 8.3.6 *Is there a polynomial algorithm to decide whether a given 2-connected graph has a strong orientation without two cycles whose length differ by one?*

8.4 Colourings and Orientations of Graphs

In this section we discuss connections between a very important parameter of an undirected graph G, its chromatic number, and properties of orientations of G.

Recall that the chromatic number of an undirected graph $G = (V, E)$, denoted $\chi(G)$ is the smallest natural number k for which V can be partitioned into disjoint independent sets $V_1, V_2, \ldots, V_k$. A more popular and obviously equivalent definition is that $\chi(G)$ is the smallest number k such that we can assign each vertex $v \in V$ a colour from the set $\{1, 2, \ldots, k\}$ without ever using the same colour for vertices that are adjacent (joined by an edge) in G. A ***k*-colouring** of an undirected graph G is any function mapping $V(G)$ to $\{1, 2, \ldots, k\}$. A k-colouring is **proper** if $f(u) \neq f(v)$ for every edge $uv \in E(G)$. For convenience we also define $\chi(D)$ for every digraph as $\chi(D) = \chi(UG(D))$.

For an arbitrary digraph $\mathrm{lp}(D)$ denotes the length of a longest path in D. The first relation we will discuss is between the number $\mathrm{lp}(D)$ and $\chi(G)$ for an arbitrary orientation D of G.

If $\chi(G) = k$, then we can obtain an acyclic orientation D of G with $\mathrm{lp}(D) = k - 1$ just by orienting all edges between V_i and V_j from V_i to V_j for all $1 \leq i < j \leq k$, where $V_1, V_2, \ldots, V_k$ is a partition of V into k disjoint independent sets. Hence if $\chi(G)$ is small then G has an orientation without long directed paths. The interesting thing is that the opposite direction also holds as was discovered independently by Gallai, Roy and Vitaver.

Theorem 8.4.1 (The Gallai-Roy-Vitaver theorem) *[297, 646, 727] For every digraph D, $\chi(D) \leq \mathrm{lp}(D) + 1$.*

Proof: Let $D = (V, A)$ be given and let $T = (V, A')$ be a maximal acyclic subdigraph of D. Define the function $f : V \to \mathcal{Z}_0$ by letting $f(v)$ equal the number of vertices in a longest path starting in v in T. Since T is acyclic, f

is well-defined. Assume that f takes values in the set $\{1, 2, \ldots, k\}$ (it is easy to see that all values in this set are taken by f). Let $V_i = \{v \in V : f(v) = i\}$.

We claim that V_i is an independent set in D for each $i = 1, 2, \ldots, k$. Clearly this will imply that $\chi(D) \leq \text{lp}(D) + 1$. Suppose $u, v \in V_i$ for some i and that there is an arc from u to v in D. Let P be a path with i vertices starting at v in T. Clearly, the arc uv does not belong to T, since otherwise uP is a path with $i+1$ vertices, contradicting the fact that $f(u) = i$ (here we used that T is acyclic to see that uP is indeed a path). By the maximality of T we get that $T + uv$ contains a cycle consisting of a path P' from v to u in T and the arc uv. Let P'' be a path with i vertices starting at u in T. Since T is acyclic the paths P' and P'' have only u in common. But now $P'P''$ is a path starting in v in T with more than i vertices, a contradiction. □

Gallai asked [297] whether every graph G has an orientation with precisely one path of length $\chi(G)$. This is not true, as shown by an example by Youngs [752]. For a detailed discussion of this topic and related problems see the book by Jensen and Toft [459].

An alternative formulation of Theorem 8.4.1 is that the chromatic number of a graph is given by

$$\chi(G) = \min\{\text{lp}(D) + 1 : D \text{ is an orientation of } G\}.^6$$

For any orientation D of an undirected graph G, we obtain an upper bound k on $\chi(G)$ from Theorem 8.4.1. It follows from the fact that the problem of finding the minimum k such that an undirected graph has a k-colouring is $\mathcal{NP}$-hard (as shown by Karp [474]) that it is an $\mathcal{NP}$-hard problem to find an orientation D of a given undirected graph G which minimizes $\text{lp}(D)$. The next theorem by Tuza shows that given an orientation D of G one can find a colouring using at most $\text{lp}(D) + 1$ colours fast.

Theorem 8.4.2 *[723] If D is a digraph such that $\text{lp}(D) < k$, then a proper k-colouring of $UG(D)$ can be found in time $O(n + m)$.* □

Bondy obtained the following generalization of Theorem 8.4.1 to strong digraphs. Note that Camion's theorem is a direct consequence of Theorem 8.4.3.

Theorem 8.4.3 *[124] Every strong digraph contains a directed cycle of length at least $\chi(D)$.* □

Minty showed that one can also measure the chromatic number of a graph by how much one can balance oriented cycles in orientations.

---

[6] Combining this with the famous 4-Colour Theorem by Appel and Haken [27] which says that every planar graph has chromatic number at most four, we see that the 4-Colour Theorem is equivalent to the statement that every planar graph has an orientation such that no directed path has length more than 3.

Theorem 8.4.4 *[567] If G has an orientation such that every oriented cycle contains at least $|V(C)|/k$ arcs in each direction, then $\chi(G) \leq k$.* □

This was strengthened by Tuza as follows.

Theorem 8.4.5 *[723] If G has an orientation such that every cycle of length $|V(C)| \equiv 1$ (modulo k) contains at least $|V(C)|/k$ arcs in each direction, then $\chi(G) \leq k$.* □

For more relations between chromatic number and paths and cycles in digraphs see Bondy's survey [126, Section 4.4] and the paper [685] by Szigeti and Tuza.

8.5 Orientations and Nowhere Zero Integer Flows

In this section, unless otherwise stated, we assume that all undirected multigraphs in question are connected.

Let $G = (V, E)$ be an undirected multigraph. A ***k*-flow** on G is an assignment of an orientation a to each edge $e \in E$ as well as an integer $x(a)$ from the set $\{1, 2, \ldots, k-1\}$ such that for each vertex v the sum of the values of x on arcs into v equals the sum of the values of x on arcs leaving v. That is, x is a circulation in the resulting oriented multigraph D. Hence we can think of a k-flow on a multigraph G as a pair (D, x) where $D = (V, A)$ is an orientation of G and x is an integer circulation in D with the property that $x(a) \in \{1, 2, \ldots, k-1\}$ for each $a \in A$. Below we use this notation. The flow x is sometimes called a **nowhere-zero *k*-flow** to stress the fact that x never takes the value zero on an arc. We say that G has a k-flow if there exists a k-flow on G. It is easy to see that a multigraph G has a k-flow for some k if and only if each connected component of G has a k-flow. Furthermore, it is easy to show that a connected multigraph with a cut-edge ($\lambda(G) = 1$) cannot have a k-flow for any k (see Exercise 8.22). It is easy to see that a pseudograph G has a k-flow if and only if the multigraph H that we obtain by deleting all loops from G has a k-flow. This is why we assume that we are working with a multigraph rather than a pseudograph below.

For convenience, we will always specify the value of a flow x on an arc uv by $x(uv)$, rather than x_{uv} as we did in Chapter 3. We start with a very easy result on 2-flows.

Proposition 8.5.1 *A multigraph G has a 2-flow if and only if all degrees of G are even.*

Proof: Clearly, if G has a 2-flow x then all degrees are even, since x is a circulation which only takes the value 1. Suppose now that all degrees of G are even. We may assume that G is connected as otherwise we consider each component in turn. By Euler's theorem, G has a closed walk $W =$

$w_0w_1w_2w_3\dots w_{m-1}w_m$, where $w_0 = w_m$ which uses each edge precisely once. Let D be the orientation obtained by orienting the edge w_iw_{i+1} from w_i to w_{i+1} for $i = 0, 1, \dots, m-1$. Then $(D, x \equiv 1)$ is a 2-flow in G. $\square$

For any abelian[7] group $(\Gamma, +)$ we can define a flow in a multigraph $G = (V, E)$ as follows. A ***Γ*-flow** in G is a pair (D, x) where D is an orientation of G, x maps $A(D)$ to the non-zero elements $\{g_1, g_2, \dots, g_{|\Gamma|-1}\}$ of Γ and x satisfies

$$\sum_{uv\in A(D)} x(uv) = \sum_{vw\in A(D)} x(vw) \qquad \text{for all } v \in V, \tag{8.1}$$

where addition is in the group Γ and $|\Gamma|$ denotes the number of elements in the group Γ. That is, x is a circulation which takes values from $\Gamma - g_0$, where g_0 is the neutral element of $(\Gamma, +)$.

Tutte proved the following important theorem, relating k-flows on a multigraph G to arbitrary group valued circulations on orientations of G.

Theorem 8.5.2 (Tutte) *[720] If $(\Gamma, +)$ is a finite abelian group, then an undirected multigraph G has a Γ-flow if and only if it has a k-flow, where $k = |\Gamma|$.* $\square$

An important step in proving Theorem 8.5.2 is to demonstrate the following theorem by Tutte. Although we do not prove Theorem 8.5.2, we still prove Theorem 8.5.3 and then use it below. The group $\mathcal{Z}_k$ is the additive group of integers modulo k.

Theorem 8.5.3 *[720] Let $G = (V, E)$ be an undirected multigraph and $k \geq 1$ an integer. Then G has a k-flow if and only if G has a $\mathcal{Z}_k$-flow.*

Proof: If (D, x) is a k-flow in G, then $x(a) \in \{1, 2, \dots, k-1\}$ for each $a \in A$ and

$$\sum_{uv\in A(D)} x(uv) - \sum_{vw\in A(D)} x(vw) = 0 \equiv 0 \qquad \text{(modulo } k).$$

Hence (D, x) is also a $\mathcal{Z}_k$-flow in G.

Suppose now that (D', x') is a $\mathcal{Z}_k$-flow in G. Since all calculations are modulo k, we may assume that $x'(a) \in \{1, 2, \dots, k-1\}$ for each $a \in A$. By the definition of a $\mathcal{Z}_k$-flow we also have

$$\sum_{uv\in A(D')} x'(uv) - \sum_{vw\in A(D')} x'(vw) \equiv 0 \text{ (modulo } k).$$

For a given $\mathcal{Z}_k$-flow $(D = (V, A), x)$, we let the balance vector b_x be defined as in (3.5), that is,

[7] Recall that an additive group $(\Gamma, +)$ is **abelian** if $a + b = b + a$ holds for all elements a, b of Γ.

$$b_x(v) = \sum_{vw \in A(D)} x(vw) - \sum_{uv \in A(D)} x(uv).$$

Now assume that (D', x') is chosen among all $\mathcal{Z}_k$-flows in G such that the sum

$$\phi(D', x') = \sum_{v \in V(D')} |b_{x'}(v)| \tag{8.2}$$

is minimized. We show that $\phi(D', x') = 0$, implying that (D', x') is a k-flow in G. Suppose this is not the case. Then let

$$P = \{v \in V : b_{x'}(v) > 0\}, M = \{v \in V : b_{x'}(v) < 0\}.$$

It follows from standard flow considerations (compare with Section 3.1) that $P, M \neq \emptyset$. By Theorem 3.3.1 we conclude that there is a path Q from P to M in D'. Let (D'', x'') be obtained by reversing all arcs of Q and changing the flow of each arc $a \in A(Q)$ to $k - x'(a)$ while leaving the flow on all arcs not on Q unchanged. It is easy to see that (D'', x'') is a $\mathcal{Z}_k$-flow in G and that $\phi(D'', x'') = \phi(D', x') - 2k$ (which is still at least zero since every vertex in P (M) contributes a positive (negative) multiple of k to the balance vector). This contradicts the choice of (D', x') and hence we must have $\phi(D', x') = 0$ implying that (D', x') is a k-flow. □

The usefulness of Theorem 8.5.2 is illustrated several times below. The point is that, as we shall see below, it is sometimes considerably easier to establish that a multigraph has a Γ-flow than it is to prove directly that it has a $|\Gamma|$-flow.

A multigraph is **cubic** if every vertex has degree 3.

Proposition 8.5.4 *A cubic multigraph G has a 3-flow if and only if G is bipartite.*

Proof: Suppose first that G is cubic and bipartite with bipartition (X, Y). Let D be the orientation obtained by orienting all edges from X to Y. Let $x \equiv 1$, then (D, x) is a $\mathcal{Z}_3$-flow in G. By Theorem 8.5.3, G has a 3-flow (D', x').

Suppose now that G is cubic and has a 3-flow (D, x). Since the only values of x are 1 and 2, it is easy to see that taking X (Y) as those vertices which are the tail (head) of an arc whose x-value is 2, we obtain a partition of $V(G)$ into two independent sets. Thus G is bipartite with bipartition (X, Y). □

A multigraph G is ***r*-edge-colourable** if one can assign each edge a number from the set $\{1, 2, \ldots, r\}$ in such a way that all edges incident to the same vertex receive different numbers. Such an assignment is also called an ***r*-edge-colouring** of G. By Exercise 3.56, every cubic bipartite multigraph is 3-edge-colourable. For general 3-edge-colourable cubic multigraphs it may not be possible to find a 3-flow (see Exercise 8.29), but one can always find a 4-flow as the next result shows.

Theorem 8.5.5 *A cubic multigraph G has a 4-flow if and only if G is 3-edge-colourable.*

Proof: By Theorem 8.5.2, G has a 4-flow if and only if it has a $\mathcal{Z}_2 \times \mathcal{Z}_2$-flow[8] . Observe that the non-zero elements of $\mathcal{Z}_2 \times \mathcal{Z}_2$ are their own inverses. Furthermore these three elements sum up to the zero element in $\mathcal{Z}_2 \times \mathcal{Z}_2$. This shows that at every vertex of G precisely one edge has flow equal to $(1,0),(0,1)$ and $(1,1)$ respectively. Thus if (D,x) is a $\mathcal{Z}_2 \times \mathcal{Z}_2$-flow in G, then we can consider the elements $(0,1),(1,0),(1,1)$ as edge colours and we obtain that G is 3-edge-colourable. This argument works the other way also and hence the claim is proved. $\square$

Theorem 8.5.6 *A multigraph G has a 4-flow if and only if it contains two eulerian subgraphs G_1, G_2 such that $E(G) = E(G_1) \cup E(G_2)$.*

Proof: Exercise 8.28.

Theorem 8.5.7 *[455] Every 4-edge-connected multigraph G has a 4-flow.*

Proof: Let $G = (V,E)$ be 4-edge-connected. By Theorem 9.5.5, G has two edge-disjoint spanning trees T_1, T_2. Every edge $e \in E - E(T_1)$ forms a unique cycle C_e with $E(T_1)$. Let E_1 be the modulo 2 sum of the edge sets of all cycles of the form C_e, $e \in E - E(T_1)$. Then the subgraph G_1 of G induced by E_1 is eulerian and contains all edges of $E - E(T_1)$. Similarly there is an eulerian subgraph G_2 which contains all edges of $E - E(T_2)$. Hence $E(G) = E(G_1) \cup E(G_2)$, because T_1 and T_2 are edge-disjoint, and the claim follows from Theorem 8.5.6. $\square$

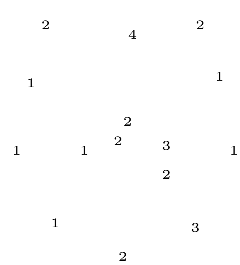


Figure 8.10 The Petersen graph with a 5-flow (D,x) indicated. Notice that the value 4 is only used once.

By Theorem 8.5.5 and the existence of 2-edge-connected cubic multigraphs which are not 3-edge-colourable (the most famous example being the

[8] The additive group $(\mathcal{Z}_2 \times \mathcal{Z}_2, +)$ has elements $\{(0,0),(1,0),(0,1),(1,1)\}$ and addition is coordinate-wise.

Petersen graph, see Figure 8.10 for an orientation of the Petersen graph) we conclude that not all 2-edge-connected multigraphs have a 4-flow. However Tutte conjectured that 4 can be replaced by 5.

Conjecture 8.5.8 (Tutte's 5-flow conjecture) *[720] Every multigraph which is 2-edge-connected has a 5-flow.*

The next lemma (described as a folklore result by Seymour in [663]) shows that it is sufficient to prove the conjecture for multigraphs which are cubic and 3-connected.

Lemma 8.5.9 *If $k \geq 3$ and $G = (V, E)$ is a 2-edge-connected multigraph which does not have a k-flow, but every 2-edge-connected multigraph $H = (V'', E'')$ with $|V''| + |E''| < |V| + |E|$ has a k-flow, then G is cubic and 3-connected.*

Proof: Suppose first that G has a cut-vertex z such that $V - z$ has connected components $H_1, \ldots, H_p$, $p \geq 2$. By the minimality of G, each of the multigraphs $H_i + z$, $i = 1, \ldots, p$ have a k flow and using these we easily obtain a k-flow for G. Hence we may assume that G is 2-connected.

Suppose $\{e, e'\}$ is a 2-edge-cut in G. Let $e = st$ and let $U' \cup W'$ be a bipartition of V such that $s \in U'$, $t \in W'$ and there is no edge between U' and W' in $G - \{e, e'\}$. Let $U = U' - s$ and $W = W' - t$. By the definition of U, W and the fact that G has no cut-vertex there is precisely one edge between U and W in G, namely e'. Now let the multigraph $G' = (V', E')$ be obtained from G by contracting e into one vertex v_e and deleting the loop created this way. Since $|V'| + |E'| < |V| + |E|$ and contraction cannot decrease edge-connectivity, it follows from the assumption on G that there is a k-flow (D', x') in G'.

In D' we may assume without loss of generality that e' is oriented as an arc a' from W to U. Let $r = x'(a')$. Since x' is a circulation the following must hold:

$$\sum_{w \in W} x'(v_e w) - \sum_{w' \in W} x'(w' v_e) = r$$

$$\sum_{u \in U} x'(v_e u) - \sum_{u' \in U} x'(u' v_e) = -\, r\ .$$

In $G - e$ the vertex s (t) is adjacent only to vertices in U (W). Let D'' be the orientation obtained by using the orientations prescribed by D' on the edges of G and orienting the edge st from s to t. Define x'' by $x''(a) = x'(a)$ for all arcs except st where we take $x''(st) = r$. Then (D'', x') is a k-flow in G, contradicting the assumption. Hence it follows that G is 3-edge-connected.

If G has a vertex s of degree at least 4, then it follows from a result of Fleischner [239] (see Exercise 8.38) that s has neighbours u, v so that replacing the edges su, sv by the edge uv we obtain a 2-edge-connected multigraph G^*[9]. By the minimal choice of G, there is a k-flow (D^*, x^*) in G^* and it is easy to obtain a k-flow in G from this (just replace the arc between u and v in D^* by a path of length 2 via s in G, using the two edges su, sv and send the appropriate amount of flow along that path). This contradicts the choice of G and hence we conclude that G is cubic. It follows from Exercise 8.23 that G is 3-connected. □

A major breakthrough on Tutte's 5-flow conjecture came when Jaeger [455] proved that every 2-edge-connected multigraph has an 8-flow. His proof was surprisingly short and elegant. The reader is asked to give a proof of Jaeger's result in Exercise 8.31.

The strongest result so far is due to Seymour.

Theorem 8.5.10 *[663, Seymour] Every 2-edge-connected multigraph has a nowhere zero 6-flow.* □

Since the proof is based on arguments that do not involve directed graphs, we will not give the proof in detail here (see Seymour's original paper [663] or the books by Diestel [191] and Fleischner [241]). It follows from Lemma 8.5.9 that it suffices to prove the result for 3-connected cubic multigraphs. Seymour proves that the edge set of such a multigraph G can be covered by two multigraphs G_1, G_2 such that G_1 is eulerian and G_2 has a 3-flow x'. It follows from Theorem 8.5.1 that G_1 has a 2-flow. Since $E(G) = E(G_1) \cup E(G_2)$ it is easy to obtain a $\mathcal{Z}_2 \times \mathcal{Z}_3$-flow in G using x, x' and hence, by Theorem 8.5.2, G has a 6-flow.

An algorithmic version of Seymour's proof, leading to a polynomial algorithm for finding a 6-flow in any 2-edge-connected multigraph, was given by Younger [751].

Recently, Bienia, Goddyn, Gvozdjak, Sebő and Tarsi proved the following interesting result. The case when $k \geq 5$ is an obvious consequence of Theorem 8.5.10.

Theorem 8.5.11 *[118] If G has a nowhere-zero flow with at most $k-1$ distinct values, then G has a k-flow.* □

For much more information on nowhere-zero flows we refer the reader to the books by Fleischner [241] and Jensen and Toft [459], the papers [456, 457] by Jaeger as well as [664] by Seymour. In particular Chapter 13 in the book by Jensen and Toft [459] contains a lot of useful information about the subject and the important open problems.

[9] In the language of Section 7.5 the result says that there is a feasible splitting su, sv (with respect to 2-edge-connectivity) for some pair of neighbours u, v of s.

8.6 Orientations Achieving High Arc-Strong Connectivity

Let us recall that an orientation D of a multigraph $G = (V, E)$ is obtained by assigning one of the two possible orientations to each edge of G (in particular two parallel edges may receive opposite orientations). By Robbins' theorem, an undirected multigraph $G = (V, E)$ has a strongly connected orientation if and only if G is 2-edge-connected.

Below we describe two generalizations of Robbins' theorem, due to Nash-Williams, both of which are much deeper than Robbins' theorem, especially the one in Theorem 8.6.4.

In order to illustrate to usefulness of the splitting technique which was discussed in Chapter 7, we prove Theorem 8.6.3 below using a splitting result for undirected graphs. This theorem, due to Lovász, is analogous to Theorem 7.5.2. The reader is asked to prove this theorem in Exercise 8.37. Analogously to the directed case, we denote by $\lambda(x, y)$ the maximum number of edge-disjoint xy-paths in G and we say that a graph $G = (V + s, E)$ with a special vertex s is k-edge-connected in V if $\lambda(x, y) \geq k$ holds for all $x, y \in V$.

Theorem 8.6.1 (Lovász's splitting theorem) *[522] Let $G = (V + s, E)$ be a multigraph with a designated vertex s of even degree and suppose that G is k-edge-connected in V, for some $k \geq 2$. Then for every edge st there exists an edge su such that after splitting off the pair st, su the new graph is still k-edge-connected in V[10].* □

An undirected multigraph $G = (V, E)$ is **minimally k-edge-connected** if G is k-edge-connected ($\lambda(G) = k$), but $\lambda(G - e) = k - 1$ for every edge $e \in E$. The following theorem by Mader is analogous to Theorem 7.10.3. The proof is left to the reader as Exercise 8.36.

Theorem 8.6.2 *[532] Every minimally k-edge-connected multigraph has a vertex of degree k.* □

Now we can prove the following famous result of Nash-Williams:

Theorem 8.6.3 (Nash-Williams' orientation theorem) *[583] An undirected multigraph $G = (V, E)$ has a k-arc-strong orientation D if and only if G is $2k$-edge-connected.*

Proof: The proof idea used below is due to Lovász [522]. Suppose G has a k-arc-strong orientation D. Thus for every non-empty proper subset X of V we have $d^+_D(X), d^-_D(X) \geq k$. This implies that in G we have $d(X) \geq 2k$ and hence, G is $2k$-edge-connected.

[10] As for directed graphs (see Section 7.5), **splitting off** the pair (su, sv) means that we replace the edges su, sv by a new edge uv (or a copy of that edge if it already exists).

To prove the other direction we proceed by induction on the number of edges in G. Let $G = (V, E)$ be $2k$-edge-connected. If $|E| = 2k$, then G is just two vertices x, y joined by $2k$ copies of the edge xy. Clearly this multigraph has a k-arc-strong orientation. Thus we may proceed to the induction step. Since adding arcs to a directed multigraph cannot decrease its arc-strong connectivity, it suffices to consider the case when G is minimally $2k$-edge-connected.

By Theorem 8.6.2, G contains a vertex s such that $d_G(s) = 2k$. Apply Lovász's splitting theorem to G with s as the special vertex and conclude that we can pair off the $2k$ edges incident to s in G in k pairs $(su_1, sv_1), \ldots, (su_k, sv_k)$ in such a way that deleting s and adding the edges $u_1v_1, \ldots, u_kv_k$ to $G - s$ results in a $2k$-edge-connected graph H. Since H has fewer edges than G it follows by induction that H has an orientation D' which is k strong.

By Exercise 7.27, we can obtain a k-arc-strong orientation of G by adding the arcs $u_1s, u_2s, \ldots, u_ks$ and the arcs $sv_1, sv_2, \ldots, sv_k$ to H. $\square$

Actually, Nash-Williams proved the following much stronger result which clearly contains Theorem 8.6.3 as a special case.

Theorem 8.6.4 (Nash-Williams' strong orientation theorem) *[583]* *An undirected graph G has an orientation D such that there are $\lfloor \frac{1}{2}\lambda_G(x, y)\rfloor$ arc-disjoint (x, y)-paths in D for every pair of vertices $x, y \in V$.* $\square$

It is beyond the scope of this book to give a complete proof here. The original proof by Nash-Williams [583] is quite complicated and so are alternative proofs by Mader (using a local edge-connectivity version of Theorem 8.6.1 [536]) and Frank [259]. It remains a real challenge to find a short and transparent proof for this important theorem.

We will outline the main idea of Nash-Williams' proof (the two other proofs use the same approach). The first observation is that, if G is eulerian, then the statement is easy to prove (Exercise 8.34). So we may assume that G is not eulerian. We can make it eulerian by adding any matching on the odd degree vertices[11]. If we could add a matching M on the odd vertices in such a way that after orienting $G + M$ as an eulerian digraph D' and then removing the arcs corresponding to M we still have

$$\lambda_D(x, y) = \lfloor \frac{1}{2}\lambda_G(x, y)\rfloor \qquad \text{for all } x, y \in V, \tag{8.3}$$

where $D = D' - M$[12], then we would have obtained the desired orientation.

[11] Recall that by Exercise 1.5, every graph has an even number of vertices of odd degree.

[12] By this we mean the oriented graph obtained from D by removing the arcs corresponding to M.

Let us see which conditions the matching M should satisfy in order to give rise to the desired orientation D as above. Following Frank [259] we use the notation $\tilde{f} = 2\lfloor f/2 \rfloor$ whenever f is an integer valued function. Let R be defined as follows: $R(\emptyset) = R(V) = 0$ and for every $\emptyset \neq X \neq V$ we let $R(X) = \max\{\lambda_G(x,y) : x \in X, y \in V - X\}$. We call R the requirement function for G. By Menger's Theorem for undirected edge-connectivity (8.3) is equivalent to requiring that

$$d^-_D(X) \geq \tilde{R}_G(X)/2 \qquad \text{for all } X \subset V. \tag{8.4}$$

A matching M on the odd vertices of G is a **good odd-vertex pairing** if

$$d_M(X) \leq d_G(X) - \tilde{R}_G(X) \qquad \text{for all } X \subset V. \tag{8.5}$$

Here $d_M(X)$ denotes the number of edges from M with precisely one end in X. Suppose M is a good odd-vertex pairing for G. Let D' be an eulerian orientation of $G + M$ and let $D = D' - M$. Then we have

$$\begin{aligned} d^-_D(X) &\geq d^-_{D'}(X) - d_M(X) \\ &= (d_G(X) + d_M(X))/2 - d_M(X) \\ &= (d_G(X) - d_M(X))/2 \\ &\geq \tilde{R}_G(X)/2, \end{aligned} \tag{8.6}$$

implying that (8.4) and hence (8.3) holds.

Thus if we can find a good odd-vertex pairing, then we get the desired orientation easily. The main point then is to prove the next theorem.

Theorem 8.6.5 *[583, Nash-Williams] Every undirected graph has a good odd-vertex pairing.* □

Instead of trying to find orientations where $\lambda_D(x,y)$ and $\lambda_D(y,x)$ are as close as possible for all pairs of vertices, one may also look for different measures for the quality of an orientation. Pekéc (private communication, October 1997) posed the following problem:

Problem 8.6.6 *Let G be an undirected graph and define M_{opt} as*

$$M_{opt} = \max\{\sum_{x,y \in V(D)} \lambda_D(x,y) : D \textit{ is an orientation of } G\}.$$

Is there a nice characterization for M_{opt}? In particular, can M_{opt} be calculated in polynomial time?

Not much is known about orientations that achieve high vertex-strong connectivity. The following conjecture by Frank is still open. Note that for $k = 1$ the conjecture follows from Robbins' theorem. Compare also with Section 7.14.

Conjecture 8.6.7 *[262, Frank] A graph $G = (V, E)$ has a k-strong orientation if and only if $G - X$ is $2(k - j)$-edge-connected for every set X of j vertices $(0 \le j \le k)$.*

8.7 Orientations Respecting Degree Constraints

In this section we first consider orientations of multigraphs which satisfy prescribed constraints on their semi-degrees. Then we consider the more general case when we have restrictions on certain subsets of the vertices (possibly all proper subsets of the vertex set). A set function f on a groundset S is **supermodular** if $f(X) + f(Y) \le f(X \cap Y) + f(X \cup Y)$ holds for every choice of sets $X, Y \subseteq S$. Recall that f is submodular on S if $f(X) + f(Y) \ge f(X \cap Y) + f(X \cup Y)$ holds for every choice of sets $X, Y \subseteq S$. The function f is **modular** if it is both submodular and supermodular[13].

8.7.1 Orientations with Prescribed Degree Sequences

We saw in Section 3.11.3 that given a directed multigraph $D = (V, A)$ and numbers $a_1, a_2, \dots, a_n$ such that $\sum_{i=1}^{n} a_i \le |A|$, we can use algorithms for maximum flows to decide whether D has an spanning subdigraph D' such that $d^-_{D'}(v_i) = a_i$ for $i = 1, 2, \dots, n$.

We start by showing that we can also solve a similar orientation problem using flows. Namely, given an undirected multigraph $G = (V, E)$, $V = \{1, 2, \dots, n\}$, and numbers $a_1, a_2, \dots, a_n$ such that $\sum_{i=1}^{n} a_i = |E|$, does G have an orientation D for which $d^-_D(i) = a_i$, for $i = 1, 2, \dots, n$?

First, form the reference orientation $H = (V, A)$ of G by orienting an edge ij from i to j whenever $i < j$. Form the network $\mathcal{N} = (V, A, l \equiv 0, u \equiv 1)$ by giving each arc of A capacity one and lower bound zero. Let us interpret a feasible integer flow x in $\mathcal{N}$ as an orientation $D' = (V, A')$ of G as follows. If $x_{ij} = 1$ then A' contains the arc ij and otherwise it contains the arc ji. Then for a given flow x we see that for each $i = 1, 2, \dots, n$, the vertex i will satisfy

$$d^-_{D'}(i) = \sum_{ji \in A} x_{ji} + (d^+_H(i) - \sum_{ij \in A} x_{ij}).$$

Since we want D' to have in-degree a_i at vertex i, for $i = 1, 2, \dots, n$, we obtain the following restriction on the balance vector b_x of x:

$$d^+_H(i) - a_i = \sum_{ij \in A} x_{ij} - \sum_{ji \in A} x_{ji} = b_x(i), \qquad \text{for } i = 1, 2, \dots, n. \tag{8.7}$$

[13] Note that a modular function f with $f(\emptyset) = 0$ satisfies $f(X) = \sum_{x \in X} f(x)$.

Thus we have reduced the orientation problem to that of deciding whether there exists a feasible flow x in $\mathcal{N}$ which has balance vector b_x as in (8.7). Hence, by Lemma 3.2.2, we can use any polynomial algorithm for maximum flow to solve the orientation problem and find the desired orientation if one exists.

Based on the reduction above and the feasibility theorem for flows (Theorem 3.8.4) one may derive necessary and sufficient conditions for the existence of an orientation with a prescribed in-degree sequence (or equivalently, out-degree sequence). One such feasibility theorem which is particularly well-known is for orientations of complete graphs as tournaments. The **score** of a vertex in a tournament is its out-degree. Landau proved the following characterization for score sequences of tournaments (the reader is asked to give a proof in Exercise 8.41):

Theorem 8.7.1 (Landau's theorem) *[508] A sequence $(s_1, s_2, \ldots, s_n)$ of integers satisfying $0 \leq s_1 \leq s_2 \leq \ldots \leq s_n$ is the score sequence of some tournament on n vertices if and only if*

$$\sum_{i=1}^{k} s_i \geq \binom{k}{2}, \qquad k = 1, 2, \ldots, n, \ \textit{with equality when } k = n.$$

□

For a very nice collection of different proofs of Landau's theorem we refer the reader to the survey paper [630] by Reid.

Harary and Moser [402] characterized score sequences of strong tournaments.

Theorem 8.7.2 *[402] A sequence $s_1 \leq s_2 \leq \ldots \leq s_n$ of non-negative integers with $n \geq 3$ is the out-degree sequence of some strong tournament if and only if for each j, $1 \leq j \leq n-1$,*

$$\sum_{i=1}^{j} s_i > \binom{j}{2}$$

and

$$\sum_{i=1}^{n} s_i = \binom{n}{2}.$$

□

Below we denote for an undirected graph $G = (V, E)$ and a subset $X \subset V$, the number of edges of E with at least one end (both ends) in X by $e_G(X)$ ($i_G(X)$). Furthermore we denote by $c(G)$ the number of connected components of G. Frank proved the following theorem which deals with bounds on the in-degrees of an orientation:

Theorem 8.7.3 *[268] Let $G = (V, E)$ be an undirected graph. Let $f : V \to \mathcal{Z}_0$ and $g : V \to \mathcal{Z}_+ \cup \{\infty\}$ be modular functions on V such that $f \leq g$. Then the following holds:*

(a) There exists an orientation D of G such that

$$d^-_D(v) \geq f(v) \qquad \text{for all } v \in V \tag{8.8}$$

if and only if

$$e_G(X) \geq f(X) \qquad \text{for all } X \subset V. \tag{8.9}$$

(b) There exists an orientation D' of G such that

$$d^-_{D'}(v) \leq g(v) \qquad \text{for all } v \in V \tag{8.10}$$

if and only if

$$i_G(X) \leq g(X) \qquad \text{for all } X \subset V. \tag{8.11}$$

(c) There exists an orientation D^* of G satisfying both (8.8) and (8.10) if and only if there is one satisfying (8.8) and one satisfying (8.10)[14].

Proof: We consider (a) first. If D satisfies (8.8) then (8.9) follows easily from the following calculation

$$\begin{aligned} f(X) = \sum_{v \in X} f(v) \leq \sum_{v \in X} d^-_D(v) & \\ = e_G(X) - d^+(X) \leq e_G(X). & \end{aligned} \tag{8.12}$$

Suppose now that (8.9) holds but there is no orientation which satisfies (8.8). Choose D among all possible orientations of G as one which minimizes

$$\sum_{\{v \in V : f(v) > d^-_D(v)\}} (f(v) - d^-_D(v)). \tag{8.13}$$

Let x be a vertex for which $f(x) > d^-_D(x)$. Let X consist of those vertices $u \in V$ for which there is a directed (x, u)-path in D. Note that by the definition of X we have $d^+_D(X) = 0$ or $X = V$. Since $f(X) \leq e_G(X)$ it is easy to see (using that $x \in X$) that there is some vertex $u \in X$ such that $d^-(u) > f(u)$. Let P be any (x, u)-path in D. Let D' be obtained from D by reversing the orientation of every arc on P. Now it is easy to see that D' either satisfies (8.8) or achieves a smaller count for (8.13). This contradiction completes the proof that (8.8) holds.

[14] Frank calls the phenomenon formulated in part (c) of the theorem the **linking principle** [259, 263].

To prove (b) we do as follows. Let g' be modular on V such that $g'(v) = \min\{d_G(v), g(v)\}$. It is easy to see that G has an orientation D satisfying $d^-_D(v) \leq g'(v)$ for all $v \in V$ if and only if it has one satisfying $d^-_D(v) \leq g(v)$ for all $v \in V$. On the other hand G has an orientation satisfying (8.10) with respect to g' if and only if it has an orientation satisfying (8.8) with respect to $f(v) = d_G(v) - g'(v)$, $v \in V$ (just consider the converse of such an orientation). By (a) such an orientation exists if and only if $e_G(X) \geq f(X)$ for each $X \subseteq V$. Using that $\sum_{x \in X} d_G(x) = e_G(X) + i_G(X)$ we conclude that $e_G(X) \geq f(X)$ if and only if $i_G(X) \leq g'(X)$. This proves (b).

To prove that (c) holds, we choose among all orientations satisfying (8.10) an orientation D which minimizes (8.13). If the sum for this D is zero, then we are done. Otherwise observe that the only vertex whose in-degree is increased by reversing the path P (as in the proof of (a)) is the vertex x for which we have $d^-_D(x) < f(v) \leq g(v)$ and hence we still have $d^-_{D'}(x) \leq g(v)$ and get the same contradiction as in the proof of (a). $\square$

The non-constructive proof above can easily be turned into a polynomial algorithm which finds the desired orientations or a proof that none exists (Exercise 8.42).

We also point out that using the approach from the beginning of this subsection, Theorem 8.7.3 can be proved using flows (Exercise 8.43).

Although Theorem 8.7.3 is fairly simple to prove, it has several consequences. One of these is Hall's theorem which characterizes the existence of a perfect matching in a bipartite graph (Theorem 3.11.3). To see that Theorem 8.7.3 implies Hall's theorem, it suffices to see that a bipartite graph $B = (U, V, E)$ has a perfect matching if and only if it has an orientation D in which every vertex in U has in-degree one and every vertex in $v \in V$ has in-degree $d_B(v) - 1$. We leave the details to the reader as Exercise 8.44. The next result, due to Ford and Fulkerson, can also be derived from Theorem 8.7.3. The proof of this is left as Exercise 8.40.

Corollary 8.7.4 *[246] Let $M = (V, A, E)$ be a mixed graph. Let $G = (V, E)$ be the undirected part and let $D = (V, A)$ be the directed part of M. The edges from G can be oriented so that the resulting directed multigraph[15] is eulerian if and only if $d_G(v) + d^-_D(v) + d^+_D(v)$ is even for each $v \in V$ and the following holds:*

$$d_G(X) \geq d^-(X) - d^+(X) \qquad \textit{for all } X \subseteq V. \tag{8.14}$$

$\square$

The following common generalization of Robbins' theorem (Theorem 1.6.2) and Theorem 8.7.3 was obtained by Frank in [268].

[15] Recall that a mixed graph may have an edge and an arc with the same end vertices.

Theorem 8.7.5 *[268] Let $G = (V, E)$ be an undirected graph which is 2-edge-connected. Let $f : V \to \mathcal{Z}_0$ and $g : V \to \mathcal{Z}_+ \cup \{\infty\}$ be modular functions on V such that $f \leq g$. Then the following holds:*

(a) There exists a strong orientation D of G such that

$$d^-_D(v) \geq f(v) \qquad \text{for all } v \in V \tag{8.15}$$

if and only if

$$e_G(X) \geq f(X) + c(G - X) \qquad \text{for all } X \subset V. \tag{8.16}$$

(b) There exists a strong orientation D' of G such that

$$d^-_{D'}(v) \leq g(v) \qquad \text{for all } v \in V \tag{8.17}$$

if and only if

$$i_G(X) + c(G - X) \leq g(X) \qquad \text{for all } \emptyset \neq X \subset V. \tag{8.18}$$

(c) There exists a strong orientation D^* of G satisfying both (8.15) and (8.17) if and only if there is one satisfying (8.15) and one satisfying (8.17). □

8.7.2 Restrictions on Subsets of Vertices

The purpose of this subsection is to study more general problems on orientations with degree conditions on subsets of vertices rather than just the vertices themselves.

Let $G = (V, E)$ be an undirected graph and let $h : 2^V \to \mathcal{Z}_+ \cup \{0\}$ satisfy $h(\emptyset) = h(V) = 0$. The function h is **fully *G*-supermodular**[16] if

$$h(X) + h(Y) \leq h(X \cap Y) + h(X \cup Y) + d_G(X, Y) \tag{8.19}$$

holds for all pairs of subsets of V (recall that $d_G(X, Y)$ denotes the number of edges in G with one end in $X - Y$ and the other in $Y - X$). If (8.19) is required to hold only for intersecting (crossing) sets, then we say that h is **intersecting (crossing) *G*-supermodular**. A set function h on G is **symmetric** if $h(X) = h(V - X)$ for every $X \subset V$. The following quite general theorem was proved in [251]. It allows one to find conditions for the existence of k-arc-strong orientations satisfying certain degree constraints on the vertices (see e.g. [259, page 98]).

[16] This strange looking definition will be easier to understand when one considers the relation between orientations of mixed graphs and submodular flows in Section 8.9. In particular, see (8.43).

Theorem 8.7.6 (Frank's orientation theorem) *[251] Let G be an undirected graph and let h be a non-negative crossing G-supermodular function on subsets of V. There exists an orientation D of G which satisfies*

$$d^-_D(X) \geq h(X) \qquad \textit{for all } X \subset V \tag{8.20}$$

if and only if both

$$e_{\mathcal{F}} \geq \sum_{V_i \in \mathcal{F}} h(V_i) \tag{8.21}$$

and

$$e_{\mathcal{F}} \geq \sum_{V_i \in \mathcal{F}} h(V - V_i) \tag{8.22}$$

hold for every partition $\mathcal{F} = \{V_1, V_2, \dots, V_t\}$ of V, where $e_{\mathcal{F}}$ denotes the number of edges connecting different V_i's. If h is intersecting G-supermodular, then (8.21) alone is necessary and sufficient. If h is fully G-supermodular, or h is symmetric and crossing supermodular, then it suffices to require (8.21) and (8.22) only for partitions of V into two sets. □

It is an easy exercise (Exercise 8.51) to show that Frank's orientation theorem implies Nash-Williams' orientation theorem.

Frank shows in [259] how to derive Theorem 8.7.6 from the theory of submodular flows discussed in Section 8.8. See also Exercise 8.66.

8.8 Submodular Flows

In all of this section we consider set functions which are integer valued and zero on the empty set. The purpose of this section is to introduce a very useful generalization of flows, due to Edmonds and Giles [215] and to show how many important theorems in graph theory and combinatorial optimization are special cases of this theory.

Let $D = (V, A)$ be a directed multigraph and let $r : A \rightarrow \mathcal{R}$ be a function on A. We use the notation

$$r^+(U) = \sum_{a \in (U, \bar{U})} r(a), \qquad r^-(U) = \sum_{a \in (\bar{U}, U)} r(a). \tag{8.23}$$

That is, $r^+(U)$ ($r^-(U)$) is the sum of the r values on arcs leaving (entering) U and $\overline{U} = V - U$[17].

In Chapter 3 it is shown that every feasible flow in a network $\mathcal{N} = (V, A, l, u, b)$ can be modeled as a circulation in an augmented network. Recall that for a circulation x in a network $\mathcal{N}$ we require that for every vertex v,

[17] Note that the function r^+ is a generalization of d^+ for any directed multigraph D, since taking $r \equiv 1$ we obtain d^+.

the flow into v equals the flow out of v. This easily translates to non-empty proper subsets of the vertex set V, i.e. for every circulation x and every non-empty proper subset U of V, $x^-(U) = x^+(U)$. The flows we will consider below do not in general satisfy this property, but there is a bound $b(U)$ on the difference between the flow into U and the flow out of U.

Let $\mathcal{F}$ be a family of subsets of S and let $b : \mathcal{F} \rightarrow \mathcal{Z} \cup \{\infty\}$ be a function defined on $\mathcal{F}$. The function b is **fully submodular** on $\mathcal{F}$ if the inequality

$$b(X) + b(Y) \geq b(X \cap Y) + b(X \cup Y) \tag{8.24}$$

holds for every choice of members X, Y of $\mathcal{F}$. If (8.24) is only required to hold for intersecting (crossing) members of $\mathcal{F}$, then b is **intersecting (crossing)** submodular on $\mathcal{F}$. By an intersecting (crossing) **pair** $(\mathcal{F}, b)$ we mean a family $\mathcal{F}$ which is intersecting (crossing) and a function b which is submodular on intersecting (crossing) subsets of $\mathcal{F}$.

8.8.1 Submodular Flow Models

Let $f : A \rightarrow \mathcal{Z} \cup \{-\infty\}$ and $g : A \rightarrow \mathcal{Z} \cup \{\infty\}$ be functions on the arc set of a directed multigraph $D = (V, A)$. Let $\mathcal{F}$ be a family of subsets of V such that $\emptyset, V \in \mathcal{F}$ and let $b : \mathcal{F} \rightarrow \mathcal{Z} \cup \{\infty\}$ be fully submodular on $\mathcal{F}$. A function $x : A \rightarrow \mathcal{R}$ is a **submodular flow** with respect to $\mathcal{F}$ if it satisfies

$$x^-(U) - x^+(U) \leq b(U) \qquad \text{for all } U \in \mathcal{F}. \tag{8.25}$$

A submodular flow x is **feasible** with respect to f, g if $f(a) \leq x(a) \leq g(a)$ holds for all $a \in A$. The set of feasible submodular flows (with respect to given f, g and $(\mathcal{F}, b)$ form a polyhedron called the **submodular flow polyhedron**) $Q(f, g; (\mathcal{F}, b))$ [259].

Submodular flows were introduced by Edmonds and Giles in [215]. In that paper it was only required that the function b is crossing submodular on a crossing family $\mathcal{F}$, something which gives much more flexibility in applications (see Subsection 8.8.4). However, as remarked in [259] the crossing submodular functions define the same class of polyhedra as do fully submodular functions.

Submodular flow polyhedra have very nice properties which makes submodular flows a very powerful tool in combinatorial optimization (see e.g. Subsection 8.8.4).

Theorem 8.8.1 (The Edmonds-Giles theorem) *[215] Let $D = (V, A)$ be a directed multigraph. Let $\mathcal{F}$ be a crossing family of subsets of V such that $\emptyset, V \in \mathcal{F}$, let $b : \mathcal{F} \rightarrow \mathcal{Z} \cup \{-\infty\}$ be crossing submodular on $\mathcal{F}$ with $b(\emptyset) = b(V) = 0$, and let $f \leq g$ be modular functions on A such that $f : A \rightarrow \mathcal{Z} \cup \{-\infty\}$ and $g : A \rightarrow \mathcal{Z} \cup \{\infty\}$. The linear system*

$$\{f \leq x \leq g \text{ and } x^-(U) - x^+(U) \leq b(U) \qquad \text{for all } U \in \mathcal{F}\} \tag{8.26}$$

is totally dual integral. That is, if f, g, b are all integer valued, then the linear program $\min \{c^T x : x$ *satisfies* $(8.26)\}$ *has an integer optimum solution (provided it has a solution). Furthermore, if c is integer valued, then the dual linear program has an integer valued optimum solution (provided it has a solution).* □

In the definition of a submodular flow, we have followed Frank [254, 255, 259, 263, 264, 274] and Schrijver [655]. Sometimes the definition of a submodular flow is slightly different (see e.g. the original paper by Edmonds and Giles [215] or the book by Fujishige [280]), namely x is required to satisfy

$$f \leq x \leq g \text{ and } x^+(U) - x^-(U) \leq b(U) \qquad \text{for all } U \in \mathcal{F}. \tag{8.27}$$

There is really no difference in these two definitions, since we see that if x satisfies (8.26), then $-x$ satisfies (8.27) with respect to the same submodular function b and the bounds $-g \leq -f$.

One can also use supermodular functions in the definition as shown in the next lemma. Hence there are several models to choose from when one wants to model a problem as a submodular flow problem. Depending on the problem at hand, one model may be easier to use than another. For an illustration of this see Section 8.8.4, where we use several different definitions.

Lemma 8.8.2 *Let $D = (V, A)$ be a directed multigraph and let $\mathcal{F}$ be a crossing family of subsets of V such that $\emptyset, V \in \mathcal{F}$. If p is a crossing supermodular function on $\mathcal{F}$ with $p(\emptyset) = p(V) = 0$, then any $x : A \to \mathcal{R}$ which satisfies*

$$x^-(U) - x^+(U) \geq p(U) \qquad \textit{for all } U \in \mathcal{F} \tag{8.28}$$

is a submodular flow.

Proof: To see this, observe that the function $b(U) = -p(\overline{U})$ is crossing submodular on the crossing family $\overline{\mathcal{F}}$ defined as the complements of sets in $\mathcal{F}$. Furthermore, by (8.23), (8.28) is equivalent to $x^-(\overline{U}) - x^+(\overline{U}) \leq -p(U) = b(\overline{U})$ for all $\overline{U} \in \overline{\mathcal{F}}$. □

8.8.2 Existence of Feasible Submodular Flows

The following theorem, characterizing when a feasible submodular flow exists with respect to functions f, g and b, is due to Frank:

Theorem 8.8.3 (Feasibility theorem for fully submodular flows) *[254] Let $D = (V, A)$ be a directed multigraph, let $f \leq g$ be modular functions on A such that $f : A \to \mathcal{Z} \cup \{-\infty\}$ and $g : A \to \mathcal{Z} \cup \{\infty\}$ and let b be a fully submodular function on 2^V. There exists an integer valued feasible submodular flow if and only if*

$$f^-(U) - g^+(U) \leq b(U) \qquad \textit{for all } U \subseteq V. \tag{8.29}$$

In particular there exist a feasible integer valued submodular flow if and only if there exists any feasible submodular flow.

Proof: We follow the proof by Frank in [259]. Suppose first that there exists a feasible submodular flow x. Then we have $f^-(U)-g^+(U) \leq x^-(U)-x^+(U) \leq b(U)$, showing that (8.29) holds.

Suppose now that (8.29) holds. Define the set function p as follows

$$p(U) = f^-(U) - g^+(U). \tag{8.30}$$

Claim: The function p is fully supermodular, that is, $p(U) + p(W) \leq p(U \cap W) + p(U \cup W)$ for all $U, W \subseteq V$. Furthermore, equality only holds if $f(a) = g(a)$ for all arcs with one end in $U - W$ and the other in $W - U$.
Proof of Claim: Since f and g are modular as set functions, we get, by considering the contribution of each arc in A:

$$\begin{aligned} p(U) + p(W) &= (f^-(U) - g^+(U)) + (f^-(W) - g^+(W)) \\ &= (f^-(U) + f^-(W)) - (g^+(U) + g^+(W)) \\ &= (f^-(U \cap W) + f^-(U \cup W) + f(U,W)) \\ &\quad -(g^+(U \cap W) + g^+(U \cup W) + g(U,W)) \\ &= (p(U \cap W) + p(U \cup W)) - (g(U,W) - f(U,W)), \end{aligned}$$

where $f(U,W)$ counts the f values on arcs with one end in $U - W$ and the other in $W - U$[18].

From this it follows that p is supermodular (since $f \leq g$) and that equality only holds if $f(a) = g(a)$ for all arcs with one end in $U - W$ and the other in $W - U$. This completes the proof of the claim.

An arc $a \in A$ is **tight** if $f(a) = g(a)$ and a subset $U \subset V$ is **tight** if $p(U) = b(U)$. Suppose that there is no feasible flow with respect to f, g and b in D and that f, g are chosen so that the number of tight arcs plus the number of tight sets is maximum.

If every arc $a \in A$ is tight, then take $x(a) = f(a) = g(a)$ for every $a \in A$. Now we have $x^-(U) - x^+(U) = f^-(U) - g^+(U) \leq b(U)$ and hence x is a feasible submodular flow in D, a contradiction.

Hence we may assume that there is some arc a_0 such that $f(a_0) < g(a_0)$. Suppose that there is no tight set which is entered by a_0. Then we can increase $f(a_0)$, until either the new value $f'(a_0)$ equals $g(a_0)$, or we find a tight set U (with respect to f', g) which is entered by a_0. It follows that the new functions f', g have a higher count of tight arcs plus tight sets. Hence, by the choice of f, g, there exists a feasible submodular flow x with respect to f', g. Obviously x is also feasible with respect to f, g, contradicting the assumption. Hence the arc a_0 must enter a tight set U.

Similarly we can prove (by lowering g otherwise) that the arc a_0 must also leave some tight set W. Now we have, using the Claim, (8.29) and the fact that $p(U) = b(U), p(W) = b(W)$:

[18] Again this definition generalizes the corresponding definition of $d(X,Y)$ in Chapter 7.

$$\begin{aligned} p(U \cap W) + p(U \cup W) &\geq p(U) + p(W) \\ &= b(U) + b(W) \\ &\geq b(U \cap W) + b(U \cup W) \\ &\geq p(U \cap W) + p(U \cup W), \end{aligned}$$

implying that equality holds everywhere above. However this contradicts the second part of the Claim since $f(a_0) < g(a_0)$ and we have argued that the arc a_0 leaves U and enters W. This contradiction completes the proof. $\square$

Note that the special case of Theorem 8.8.3 when $b \equiv 0$ says that $x^-(U) - x^+(U) = 0$ for all subsets $U \subseteq V$. In particular $x^-(v) = x^+(v)$ for all $v \in V$. That is, every feasible submodular flow with respect to f, g and $b \equiv 0$ is circulation and conversely. It is easy to see that the characterization in Theorem 8.8.3 in the case $b \equiv 0$ is exactly the condition in Hoffman's circulation Theorem (Theorem 3.8.2).

In fact, the proof of Theorem 8.8.3 in some sense resembles that of Theorem 3.8.2. Thus it is natural to ask how easy it is to find a feasible solution, or detect that none exists. This can be read out of the proof above: the essential step is to decide whether an arc enters or leaves a tight set (or both). This requires that we can find $\min\{b(U) - p(U) : a \in (U, \overline{U})\}$ and $\min\{b(U) - p(U) : a \in (\overline{U}, U)\}$ for every arc a of the directed multigraph D. This is a special case of the problem of minimizing a submodular function, that is, finding the minimum value of the submodular function in question over a prescribed family of sets. This can be done in polynomial time for arbitrary submodular functions using the ellipsoid method as shown by Grötschel, Lovász, and Schrijver [338]. However, the ellipsoid method, though polynomial, is not of practical use, since it is highly inefficient.

It was an open problem for several decades whether there exists a polynomial combinatorial algorithm for minimizing a submodular function b over a family $\mathcal{F}$, that is, to find $\min\{b(U) : u \in \mathcal{F}\}$. For submodular functions which are symmetric (that is, $b(X) = b(V - X)$) Queyranne [617] has given such a polynomial algorithm (Nagamochi and Ibaraki proved a slightly more general result [582]). Queyranne's algorithm is a generalization of the algorithm by Nagamochi and Ibaraki [580] for finding the edge-connectivity of a graph via maximum adjacency orderings which was mentioned in Section 7.4. Recently Schrijver [660] solved the problem completely by describing a strongly polynomial time algorithm for minimizing an arbitrary submodular function given by a value-giving oracle. Schrijver's algorithm does not use the ellipsoid method or any other linear programming method. A similar result was obtained independently by Iwata, Fleischer and Fujishige [447].

It should be noted that even though the special problem we described above of finding the minimum of $b(U)$ over those $U \in \mathcal{F}$ that contain either the head or the tail, but not both, of a fixed arc $a \in A$, and b is fully submodular seems to be a very special case of the problem of minimizing an arbitrary submodular function, it is in fact equivalent to that problem. Let $\mathcal{F}$ be a crossing family on a ground-set S and let b be a crossing submodular function on $\mathcal{F}$. Let D be the complete directed multigraph on the vertex set S. Let $\mathcal{F}_{uv} = \{X \in \mathcal{F} : u \in X, v \notin X\}$. Then $\mathcal{F}_{uv}$ is a crossing family and clearly

$$\min\{b(X) : X \in \mathcal{F}\} = \min\{\min\{b(Y) :\in \mathcal{F}_{uv}\} : u, v \in S\}. \tag{8.31}$$

Hence if we have a polynomial algorithm to minimize arbitrary submodular functions over families of the type $\mathcal{F}_{uv}$, then there is one for arbitrary crossing families.

As mentioned earlier, one can also define submodular flows for functions b that are intersecting, respectively crossing, submodular functions (defined on a family of subsets of the directed multigraph D which is intersecting, respectively crossing). In the case of intersecting and in particular for crossing submodular flows the feasibility theorem is much more complicated. A collection $U_1, U_2, \ldots, U_k$ of subsets of a ground set S are **co-disjoint** if their complements are pairwise disjoint (that is, $U_i \cup U_j = S$ for all $i \neq j$). Frank proved the following two feasibility theorems for intersecting and crossing submodular flows:

Theorem 8.8.4 (Feasibility theorem for intersecting submodular flows) *[255] Let $D = (V, A)$ be a directed multigraph and let f, g be real valued modular functions such that $f \leq g$. Let $\mathcal{F}'$ be an intersecting family of subsets of V such that $\emptyset, V \in \mathcal{F}'$ and let b' be an intersecting submodular function on $\mathcal{F}'$. Then there exists a feasible submodular flow with respect to f, g and b' if and only if*

$$f^-(\bigcup_{i=1}^{t} X_i) - g^+(\bigcup_{i=1}^{t} X_i) \leq \sum_{i=1}^{t} b'(X_i) \tag{8.32}$$

holds whenever $X_1, X_2, \ldots, X_t$ are disjoint members of $\mathcal{F}'$. Furthermore, if f, g, b' are all integer valued functions and (8.32) holds, then there exists a feasible integer valued submodular flow with respect to f, g and b'. □

Theorem 8.8.5 (Feasibility theorem for crossing submodular flows) *[255] Let $D = (V, A)$ be a directed multigraph and let f, g be real valued modular functions such that $f \leq g$. Let $\mathcal{F}''$ be a crossing family of subsets of V such that $\emptyset, V \in \mathcal{F}''$ and let b'' be a crossing submodular function on $\mathcal{F}''$. Then there exists a feasible submodular flow with respect to f, g and b'' if and only if*

$$f^-(\bigcup_{i=1}^{t} X_i) - g^+(\bigcup_{i=1}^{t} X_i) \leq \sum_{i=1}^{t} \sum_{j=1}^{q_i} b''(X_{ij}) \tag{8.33}$$

holds for every subpartition $\{X_1, X_2, \ldots, X_t\}$ of V such that each X_i is the intersection of co-disjoint members $X_{i1}, X_{i2}, \ldots, X_{iq_i}$ of $\mathcal{F}''$. Furthermore, if f, g, b'' are all integer valued functions and (8.33) holds, then there exists a feasible integer valued submodular flow with respect to f, g and b''. □

Finding a feasible submodular flow or a configuration which shows that none exists is much more difficult than finding a feasible circulation in a network (recall Section 3.8). Frank [255] gave a combinatorial algorithm for finding a feasible integer valued submodular flow with respect to bounds f, g and a pair $(\mathcal{F}, b)$ which is either intersecting or crossing submodular. The algorithm is polynomial provided one has an algorithm for minimizing the involved submodular functions. For this task we can apply the recent algorithms of Schrijver and Iwata, Fleischer and Fujishige which we mentioned above.

8.8.3 Minimum Cost Submodular Flows

Let $D = (V, A)$ be a directed multigraph and let $f : A \rightarrow \mathcal{Z} \cup \{-\infty\}$, $g : A \rightarrow \mathcal{Z} \cup \{\infty\}$ be functions on the arc set of D. Let $c : A \rightarrow \mathcal{R}$ be a cost function on the arcs of D. Let $\mathcal{B} \subseteq 2^V$ be a crossing family with $\emptyset, V \in \mathcal{B}$. Let $b : 2^V \rightarrow \mathcal{Z} \cup \{\infty\}$ be crossing submodular on $\mathcal{B}$ with $b(\emptyset) = b(V) = 0$. Denote the network defined by D and these functions by $\mathcal{N}_S = (V, A, f, g, (\mathcal{B}, b), c)$. The **minimum cost submodular flow problem** is as follows:

$$\begin{aligned}
&\text{Minimize} \quad \sum_{a \in A} c(a)x(a) \\
&\text{subject to} \\
&\qquad x^-(U) - x^+(U) \leq b(U) \qquad \text{for all } U \in \mathcal{B} \\
&\qquad f(a) \leq x(a) \leq g(a) \qquad \text{for all } a \in A.
\end{aligned}$$

A feasible submodular flow with respect to f, g and b which achieves this minimum is called an **optimal submodular flow** in $\mathcal{N}_S$.

This problem, which again generalizes the minimum cost circulation problem from Chapter 3, is very interesting because it forms a common extension of many problems on (di)graphs as well as problems from other areas (see e.g. the book [280] by Fujishige). Recall also Theorem 8.8.1.

Fujishige proved the following (see also the papers [170] by Cunningham and Frank and Frank's paper [254]):

Theorem 8.8.6 *[281] The minimum cost submodular flow problem can be solved in polynomial time provided a polynomial algorithm for minimizing the relevant submodular functions is available.* □

8.8.4 Applications of Submodular Flows

In this section we will illustrate the usefulness of submodular flows as a tool to obtain short proofs of important results as well as algorithms for various connectivity problems.

We start with Nash-Williams' orientation theorem (Theorem 8.6.3). The approach taken is due to Frank [256] (the same idea was used by Jackson [451]). Let $G = (V, E)$ be an undirected graph. Let D be an arbitrary orientation of G. Clearly G has a k-arc-strong orientation if and only if it is possible to reorient some arcs of D so as to get a k-arc-strong directed multigraph. Suppose we interpret the function $x : A \rightarrow \{0, 1\}$ as follows: $x(a) = 1$ means that we reorient a in D and $x(a) = 0$ means that we leave the orientation of a as it is in D. Then G has a k-arc-strong orientation if and only if we can choose x so that the following holds:

$$d_D^-(U) + x^+(U) - x^-(U) \geq k \qquad \text{for all } \emptyset \neq U \subset V. \tag{8.34}$$

This is equivalent to

$$x^-(U) - x^+(U) \leq (d_D^-(U) - k) = b(U) \qquad \text{for all } U \subset V, U \neq \emptyset, V, \tag{8.35}$$

$$b(\emptyset) = b(V) = 0. \tag{8.36}$$

Observe that the function b is crossing submodular on $\mathcal{F} = 2^V$ (it is not fully submodular in general, since we have taken $b(\emptyset) = b(V) = 0$). Thus we have shown that G has a k-arc-strong orientation if and only if there exists a feasible integer valued submodular flow in D with respect to the functions $f \equiv 0, g \equiv 1$ and b.

Suppose now that G is $2k$-edge-connected, that is, $d_G(X) \geq 2k$ for all proper non-empty subsets of V. We claim that $x \equiv \frac{1}{2}$ is a feasible submodular flow. This follows from the following calculation:

$$\begin{aligned}
d_D^-(U) + x^+(U) - x^-(U) &= d_D^-(U) + \frac{1}{2}d_D^+(U) - \frac{1}{2}d_D^-(U) \\
&= \frac{1}{2}d_D^-(U) + \frac{1}{2}d_D^+(U) \\
&\geq \frac{1}{2}(2k - d_D^+(U)) + \frac{1}{2}d_D^+(U) \\
&= k.
\end{aligned}$$

Hence it follows from the integrality statement of Theorem 8.8.5 and the equivalence between (8.34) and (8.35) that there is a feasible integer valued submodular flow x in D with respect to f, g and b. As described above this implies that G has a k-arc-strong orientation where the values of x prescribe which arcs to reverse in order to obtain such an orientation from D.

Notice that by formulating the problem as a minimum cost submodular flow problem, we can also solve the weighted version where the two possible orientations of an edge may have different costs and the goal is to find the cheapest k-arc-strong orientation of the graph (Exercise 8.64). By Theorem 8.8.6, the optimal (minimum cost feasible) submodular flow in D with respect to the functions $f \equiv 0, g \equiv 1$ and b (as defined in (8.35)) can be found in polynomial time (see Exercise 8.63).

The following useful result, mentioned by Frank in [254], follows from the discussion above and Theorem 8.8.6

Corollary 8.8.7 *[254] There is a polynomial algorithm for finding the minimum number of arcs to reverse in a directed multigraph D in order to obtain a k-arc-strong reorientation of D.* □

Similarly, combining the discussion above with Frank's algorithm for finding a feasible submodular flow (or deciding that none exists) with respect to a crossing submodular function, we obtain the following result (see Exercise 8.39 for a different proof based on Lovász's splitting theorem):

Corollary 8.8.8 *[254] There is a polynomial algorithm for finding a k-arc-strong orientation of a given undirected multigraph G or verify that G has no such orientation.* □

The following theorem by Frank can also be derived from the formulation of the k-arc-strong orientation as a submodular flow problem (see Fujishige's book [280]).

Theorem 8.8.9 *[253] If D and D' are k-arc-strong orientations of an undirected graph G, then there exists a sequence of k-arc-strong orientations $D = D_0, D_1, \dots, D_r = D'$ of G such that for each $i = 1, 2, \dots, r$, D_i is obtained from D_{i-1} by reversing all arcs in a directed path or a directed cycle.* □

Frank [253] gives a direct and short proof of this without using submodular flows, but his proof uses submodular arguments (see Exercises 8.47-8.50).

In [275] Frank and Tardós showed how to reduce the following problem to a submodular flow problem. Given a directed graph $D = (V, A)$ and a special vertex s, find a minimum set of new arcs to add to D such that the resulting directed multigraph contains k internally disjoint paths from s to v for every $v \in V - s$. The similar problem where we only want arc-disjoint (s, v)-paths is solvable via matroid intersection algorithms (see Exercise 9.57). In the special case when D has already $k - 1$ arc-disjoint (s, v)-path for all $v \in V$ the problem can also be solved by the Frank-Fulkerson algorithm, which is discussed in Section 9.11.

As another prominent illustration of the generality of submodular flows, let us now show how the Lucchesi-Younger theorem on coverings of arc-disjoint directed cuts[19] (Theorem 7.15.2) can be proved using a formulation of the problem as a minimum cost submodular flow problem and the duality theorem for linear programming. This application of submodular flows was first pointed out by Edmonds and Giles [215].

We wish to find a minimum set of arcs which cover all directed cuts in D. We assume that D is connected, since otherwise some dicut has no arcs at all and clearly no cover exists (recall Section 7.15). Let $x : A \rightarrow \{0, 1\}$ and let us interpret the value of $x(a)$ as follows. If $x(a) = 1$ then we choose a to be in the cover and otherwise (if $x(a) = 0$) a is not chosen. Since the set of chosen arcs must cover all directed cuts, we have the requirement

$$x^-(W) \geq 1 \qquad \text{for all } \emptyset \neq W \subset V \text{ such that } d^+_D(W) = 0. \tag{8.37}$$

Let $\mathcal{F} = \{W : d^+_D(W) = 0\}$. Then $\emptyset, V \in \mathcal{F}$ and (8.37) is equivalent to

$$x^+(W) - x^-(W) \leq b(W) \qquad \text{for all } W \in \mathcal{F}, \tag{8.38}$$

where $b(\emptyset) = b(V) = 0$ and $b(W) = -1$ for all $W \in \mathcal{F} - \{\emptyset, V\}$.

By our remark on different formulations of submodular flow problems, we see that this (having the form of (8.27)) is indeed a submodular flow formulation. Hence by assigning cost one to each arc we can formulate the problem of finding an optimal cover of the directed cuts as the following minimum cost submodular flow problem (in the form of (8.27)).

$$\begin{aligned}
\mathcal{LY}: \text{ Minimize } & \sum_{a \in A} x(a) \\
\text{subject to } & \\
& x^-(W) \geq 1 \qquad \text{for all } W \in \mathcal{F} - \{\emptyset, V\} \\
& 0 \leq x(a) \leq 1 \qquad \text{for all } a \in A.
\end{aligned}$$

Taking dual variables y_W for each member W of $\mathcal{F}$ and $\epsilon(a)$ for each arc $a \in A$, we get that the dual of $\mathcal{LY}$ is

$$\begin{aligned}
\mathcal{LY}^*: \text{ Maximize } & \sum_{W \in \mathcal{F} - \{\emptyset, V\}} y_W - \sum_{a \in A} \epsilon(a) \\
\text{subject to } & \\
& -\epsilon(a) + \sum_{a \in (\overline{W}, W)} y_W \leq 1 \qquad \text{for all } a \in A \\
& y_W \geq 0 \qquad \text{for all } W \in \mathcal{F}
\end{aligned}$$

[19] Recall from Chapter 7 that a directed cut is a set of arcs of the form $(U, V - U)$ where $d^-(U) = 0$.

$$\epsilon(a) \geq 0 \qquad \text{for all } a \in A.$$

Eliminate the variables $\epsilon(a)$ from $\mathcal{LY}^*$ and notice that, if $y_W = 0$ for all members $W \in \mathcal{F}$ which are entered by a, then the optimal choice for $\epsilon(a)$ is $\epsilon(a) = 0$. We get that $\mathcal{LY}^*$ is equivalent to the problem

$$\mathcal{LY}^{**} : \text{Maximize} \quad \sum_{W \in \mathcal{F} - \{\emptyset, V\}} y_W + \sum_{a \in A} \min\{0, [1 - \sum_{a \in (\overline{W}, W)} y_W]\} \tag{8.39}$$

subject to

$$y_W \geq 0 \qquad \text{for all } W \in \mathcal{F}. \tag{8.40}$$

By the Edmonds-Giles theorem, there exist an integer valued optimum solution $\{y_W : w \in \mathcal{F}\} \cup \{\epsilon(a) : a \in A\}$ to $\mathcal{LY}^*$ and hence to $\mathcal{L}^{**}$. Notice that, if some variable y_W in such a solution is 2 or more, then we can decrease its value to 1 without changing the value of the objective function in (8.39). Hence there exists an optimal solution to $\mathcal{LY}^{**}$ in which all values are 0 or 1. It follows from the optimality of the solution that, if $y_W = y_{W'} = 1$, then we can assume that no arc enters both of W, W' (otherwise we may put $y_{W'} = 0$ without changing the value of the objective function). This shows that the cuts corresponding to the non-zero values of y are arc-disjoint and hence we have shown that the size of an optimal cover equals the maximum number of arc-disjoint directed cuts, which is exactly the statement of Theorem 7.15.2. Furthermore, by Theorem 8.8.6, we obtain the following corollary:

Corollary 8.8.10 *There exists a polynomial algorithm which given a directed multigraph $D = (V, A)$ finds a minimum dijoin $A' \subseteq A$ of D.* □

Note that we can minimize the function b from (8.38) over a given collection of sets in polynomial time (using flows). Namely, the minimum value is -1 if the collection contains a member of $\mathcal{F}$ and 0 otherwise.

It follows from the formulation of the minimum directed cut covering problem as a submodular flow problem and Theorem 8.8.6 that we can also solve the minimum cost version of the problem even if there are non-uniform costs on the arcs and we want to find a minimum cost cover of the directed cuts. Furthermore, we can also solve the problem of finding a set of arcs which cover each directed cut at least k times for each k (simply replace the number -1 by $-k$ in (8.38)).

For much more material on submodular flows the reader is referred to the papers [254, 255, 259, 263, 264] by Frank, [274] by Frank and Tardos, Fujishige's book [280] and the paper [655] by Schrijver. In particular [274] and [280] give a lot of interesting results on the structure of submodular flows and the relation between submodular flows and other models such as independent flows and polymatroidal flows. Finally Schrijver's paper [655] is a very useful overview of the various models and their interrelations.

8.9 Orientations of Mixed Graphs

We conclude this chapter with some remarks on the orientation of mixed graphs where the goal is to satisfy degree and/or connectivity requirements. Note that in this section a mixed graph may contain multiple edges and/or arcs. Also recall that when we speak of orienting a mixed (multi)graph this means that we assign an orientation to every edge and leave the original arcs unchanged (implying that the result may not be an oriented graph).

Orientation problems for mixed graphs are generally much harder than for undirected graphs. One illustration of this is displayed in Figure 8.11. This example, due to Tardos (see [263]), shows that the linking principle for strong connectivity orientations does not hold for general mixed graphs (compare this with Theorem 8.7.5).

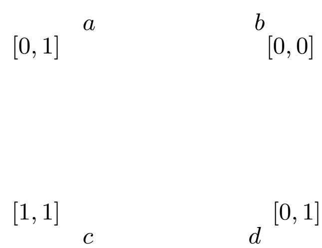


Figure 8.11 A mixed graph M with prescribed lower and upper bounds on the desired in-degrees in the directed multigraph induced by the arc between a and c and the arc between b and d in an orientation D of M. It is easy to see that by orienting the edges ac, bd as $a{\rightarrow}c, d{\rightarrow}b$ we obtain a strong orientation satisfying the lower bounds on the directed multigraph induced by the newly oriented arcs. Similarly if we orient the same edges as $c{\rightarrow}a, b{\rightarrow}d$ we obtain a strong orientation which satisfies the upper bounds on the directed multigraph induced by the newly oriented arcs. However, there is no strong orientation which satisfies the lower and upper bounds simultaneously on the directed multigraph induced by the newly oriented arcs.

Not every $2k$-arc-strong mixed graph has a k-arc-strong orientation (Exercise 8.54) but Jackson proved the following extension of Theorem 8.6.3. The proof is left to the reader as Exercise 8.53.

Theorem 8.9.1 *[451] Let $M = (V, A, E)$ be a mixed graph. Let $G = (V, E)$ and $D = (V, A)$ denote the undirected, respectively the directed part of M and define k by*

$$k = \min\{\frac{1}{2}d_G(X) + d^+_D(X) : X \text{ is a proper, non-empty subset of } V\}.$$

Then the edges of E can be oriented in such a way that the resulting directed multigraph is k-arc-strong. □

It is not difficult to see that one can formulate the problem of orienting a mixed graph so as to get a k-arc-strong directed multigraph as a submodular flow problem. We can use the same approach as in Subsection 8.8.4. The only change is that we insist that $x(a) = 0$ for original arcs (Exercise 8.53).

Jackson [451] conjectured that Theorem 8.9.1 could be extended to local connectivities and hence providing a generalization of Nash-Williams' strong orientation theorem (Theorem 8.6.4). However, examples by Enni [218] show that this conjecture is false. In the case when the directed part of $M = (V, A, E)$ is eulerian such an extension is indeed possible. In [218] Enni shows how to extend Theorem 8.6.5 to the case of mixed graphs when the directed part $D = (V, A)$ is eulerian.

We remark that there seems to be no easy way of formulating orientation problems concerning local connectivities as submodular flow problems.

When we consider orientation problems where the input is a mixed graph $M = (V, A, E)$ which we wish to orient so as to satisfy a certain lower bound $h(X)$ on the in-degree of every subset X of vertices, then we cannot in general apply a theorem like Frank's orientation theorem (Theorem 8.7.6). The reason for this is that even if the function $h(X)$ 'behaves nicely', we have to take into account the arcs in A because these will contribute to the in-degree of the final oriented graph D'. To give an example, consider a mixed graph $M = (V, A, E)$ and let $h(X) = k$ for all non-empty proper subsets of V and $h(\emptyset) = h(V) = 0$. That is, we are looking for a k-arc-strong orientation of M. When we want to apply a theorem like Theorem 8.7.6 we have to consider the revised in-degree lower bound h' given by $h'(X) = k - d^-_D(X)$, where $D = (V, A)$ is the directed graph induced by the arcs already oriented in M. The function h' is easily seen to be crossing G-supermodular, where $G = (V, E)$ is the undirected part of M (Exercise 8.62). However h' is typically negative on certain sets and hence Theorem 8.7.6 cannot be applied.

As we mentioned above, for the particular lower bound $h(X) = k$, whenever $\emptyset \neq X \neq V$, the problem can be formulated as a submodular flow problem. This is no coincidence as we show below.

Let $G = (V, E)$ be an undirected graph. Let $h : 2^V \rightarrow \mathcal{Z} \cup \{-\infty\}$ be crossing G-supermodular with $h(\emptyset) = h(V) = 0$. Let $D = (V, A)$ be an arbitrary but fixed orientation of G. Let $x : A(D) \rightarrow \{0, 1\}$ be a vector and define an orientation $D' = (V, A')$ of G by taking $A' = \{a : a \in A, x(a) = 0\} \cup \{\overleftarrow{a} : a \in A, x(a) = 1\}$. Here $\overleftarrow{a}$ denotes the opposite orientation of the arc a (compare with Section 8.8.4). Then D' will satisfy

$$d^-_{D'}(U) \geq h(U) \qquad \text{for all } U \subset V \tag{8.41}$$

if and only if $d^-_D(U) - x^-(U) + x^+(U) \geq h(U)$ for all $U \subset V$, or equivalently

$$x^-(U) - x^+(U) \leq d^-_D(U) - h(U) = b''(U) \qquad \text{for all } U \subset V. \tag{8.42}$$

Since d_D^- satisfies (7.2) and h is crossing G-supermodular[20], we conclude that whenever U, W are crossing sets the following holds:

$$\begin{aligned} b''(U) + b''(W) &= (d_D^-(U) - h(U)) + (d_D^-(W) - h(W)) \\ &= d_D^-(U \cap W) + d_D^-(U \cup W) + d_G(U, W) - (h(U) + h(W)) \\ &\geq d_D^-(U \cap W) + d_D^-(U \cup W) + d_G(U, W) - (h(U \cap W) \\ &+ h(U \cup W) + d_G(U, W)) \\ &= b''(U \cap W) + b''(U \cup W). \end{aligned} \tag{8.43}$$

Thus the function b'' is crossing submodular on $\mathcal{F}'' = 2^V - \{\emptyset, V\}$ and the equivalence of (8.41) and (8.42) shows that there is a one to one correspondence between orientations satisfying (8.41) and integer valued solutions to the submodular flow problem defined by (8.42) and $0 \leq x \leq 1$. This shows that we can use submodular flow algorithms to solve the orientation problem. We can also derive a characterization of the existence of an orientation satisfying (8.41) from Theorem 8.8.5. We do this below as an illustration of how to use the feasibility theorem for crossing submodular flows (Theorem 8.8.5).

Suppose there exists an integer valued feasible submodular flow with respect to the crossing submodular function b'' defined above. By (8.33) this means that

$$f^-(\bigcup_{i=1}^{t} X_i) - g^+(\bigcup_{i=1}^{t} X_i) \leq \sum_{i=1}^{t} \sum_{j=1}^{q_i} b''(X_{ij}) \tag{8.44}$$

holds for every subpartition $\mathcal{P} = \{X_1, X_2, \ldots, X_t\}$ of V such that each X_i is the intersection of co-disjoint subsets $X_{i1}, X_{i2}, \ldots, X_{iq_i}$ of V.

We derive an expression that relates only to G and h using (8.44). To do so, it is helpful to study Figure 8.12.

Using that $f \equiv 0$ and $g \equiv 1$ and the definition of b'' we see that (8.44) is equivalent to

$$-d_D^+(\bigcup_{i=1}^{t} X_i) \leq \sum_{i=1}^{t} \sum_{j=1}^{q_i} (d_D^-(X_{ij}) - h(X_{ij})). \tag{8.45}$$

For fixed i the sum $\sum_{j=1}^{q_i} d_D^-(X_{ij})$ counts the following arcs:

(1) Those arcs which enter X_i (the common intersection of all X_{ij}'s) from its complement, plus

[20] Note how we use the definition of a crossing G-supermodular function here to get rid of the contribution from edges with one end in $X - Y$ and the other in $Y - X$.

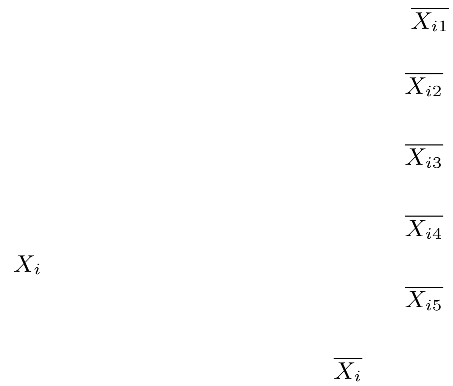


Figure 8.12 The situation when deriving Theorem 8.9.2 from Theorem 8.8.5. The set X_i is part of a subpartition $\mathcal{P}$ of V and X_i is the intersection of the five co-disjoint sets $X_{i1}, \ldots, X_{i5}$ whose complements (which form a partition of $\overline{X_i}$) are indicated in the figure. The arcs shown are those between different sets $\overline{X_{ij}}$, $\overline{X_{ir}}$ (which are the same as those arcs that go between different X_{ij}'s!) and those arcs that enter X_i.

(2) those arcs which go between different X_{ij}'s (which is the same as arcs that go from some $\overline{X_{ij}}$ to some other $\overline{X_{ir}}$). This is the same as the number of edges in G that go between two X_{ij}'s. Denote the total number of edges of this kind in G by e_i.

Using this observation we conclude that (8.45) is equivalent to

$$d^+_D(\cup_{i=1}^t X_i) + \sum_{i=1}^t d^-_D(X_i) \geq \sum_{i=1}^t (\sum_{j=1}^{q_i} h(X_{ij}) - e_i). \tag{8.46}$$

Finally, observe that the left hand side of (8.46) counts precisely those edges of G which enter some $X_i \in \mathcal{P}$. Now we have proved the following orientation theorem due do Frank:

Theorem 8.9.2 (Frank's general orientation theorem) *[259] Let $G = (V, E)$ be an undirected graph. Let $h : 2^V \rightarrow \mathcal{Z} \cup \{-\infty\}$ be crossing G-supermodular with $h(\emptyset) = h(V) = 0$. There exists an orientation D of G satisfying*

$$d^-_D(X) \geq h(X) \qquad \textit{for all } X \subset V \tag{8.47}$$

if and only if

$$e_{\mathcal{P}} \geq \sum_{i=1}^t (\sum_{j=1}^{q_i} h(X_{ij}) - e_i) \tag{8.48}$$

holds for every subpartition $\mathcal{P} = \{X_1, X_2, \ldots, X_t\}$ of V such that each X_i is the intersection of co-disjoint sets $X_{i1}, X_{i2}, \ldots, X_{iq_i}$. Here $e_{\mathcal{P}}$ counts the *number of edges which enter some member of $\mathcal{P}$ and e_i counts the number of edges which go between different sets X_{ij}, X_{ir}.* □

By our previous remark on the function $k - d_D^-$, Theorem 8.9.2 can be used to derive a necessary and sufficient condition for the existence of a k-arc strong orientation of a mixed graph. This is left to the reader as Exercise 8.56.

One might ask whether such a complicated condition involving partitions and copartitions is really necessary in Theorem 8.9.2. The following example due to Frank [263] shows that one cannot have a condition which only involves partitions or subpartitions.

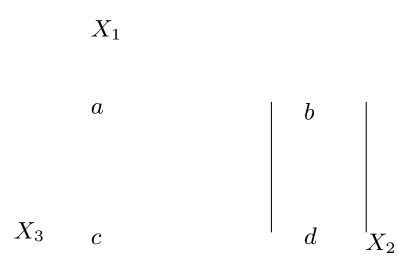


Figure 8.13 Frank's example showing that no (sub)partition type condition for the existence of an orientation satisfying (8.47) exists.

Let $G = (V, E)$ be the graph in Figure 8.13 and let the sets X_1, X_2, X_3 be as defined there. Define h by $h(\emptyset) = h(V) = 0$, $h(X_1) = h(X_3) = 1$, $h(X_2) = 2$ and $h(X) = -\infty$ for all other subsets of V. Then h is crossing G-supermodular since no two crossing sets X, Y have $h(X), h(Y) > -\infty$. It is easy to check that G has no orientation satisfying (8.47) with respect to h. On the other hand, if we decrease $h(X_i)$ by one for any $i = 1, 2, 3$, then there exists a feasible orientation with respect to the new h_i. This shows that every certificate for the non-existence of an orientation with respect to h must include all the sets X_1, X_2, X_3. It is easy to see that these three sets neither form a subpartition nor do they form a co-partition.

The example from Figure 8.13 also shows that there is no 2-arc-strong orientation of the mixed graph in Figure 8.14. Hence even for orientations of mixed graphs to obtain a uniform degree of arc-strong connectivity we cannot hope for a much simpler condition.

Since we derived Theorem 8.9.2 from Theorem 8.8.5, it is possible to get a simpler characterization if one can find such a characterization of feasibility of submodular flows with respect to a crossing pair $(\mathcal{F}'', b'')$. This was done

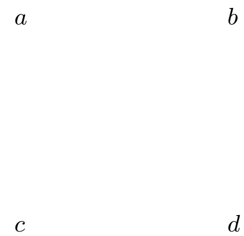


Figure 8.14 A mixed graph which has no 2-arc-strong orientation and for which every certificate for the non-existence of such an orientation must involve the three sets $\{a\}, \{b,d\}, \{a,b,c\}$ [263, Figure 2.3].

recently by Frank in [263] where a somewhat simpler (but still far from easy) characterization was found.

8.10 Exercises

8.1. Prove Lemma 8.1.16.

8.2. Show how to check whether an undirected graph is bipartite in linear time using BFS. Does your method extend to strongly connected digraphs? That is, can you check whether a strong digraph is bipartite using BFS? Hint: consider the proof of Theorem 1.8.1.

8.3. Show that, if a locally semicomplete digraph D contains a 2-cycle xyx, then the edge xy is balanced in $UG(D)$.

8.4. **(+) Lexicographic 2-colouring gives a transitive orientation of comparability graphs**. Prove Theorem 8.1.4.

8.5. Prove that, if G is a reduced proper circular-arc graph, then, up to reversing the orientation of all arcs, G has a unique orientation as a round local tournament.

8.6. **(+) Linear algorithm for recognizing round local tournaments**. Prove that there is an $O(n+m)$ algorithm which either finds a round labelling of an oriented graph D or decides that D is not a round local tournament (Huang [436]).

8.7. Prove Theorem 8.1.8.

8.8. Using the same approach as in the proof of Proposition 8.1.15 formulate the instance of 2-SAT which corresponds to the oriented graph D in Figure 8.15. Show that $UG(D)$ has no orientation as a locally in-tournament digraph.

8.9. **An orientation characterization of proper interval graphs**. A **straight enumeration** of an oriented graph is a linear ordering $v_1, v_2, \ldots, v_n$ such that for each i the vertex v_i is dominated by $v_{i-d^-(v_i)}, v_{i-d^-(v_i)+1}, \ldots, v_{i-1}$ and dominates $v_{i+1}, v_{i+2}, \ldots, v_{i+d^+(v_i)}$. Here indices are **not** modulo n, that is, $1 \leq i - d^-(v_i)$ and $i + d^+(v_i) \leq n$ for each $i = 1, 2, \ldots, n$. A digraph is **straight** if it has a straight enumeration (Deng, Hell and Huang [190]).

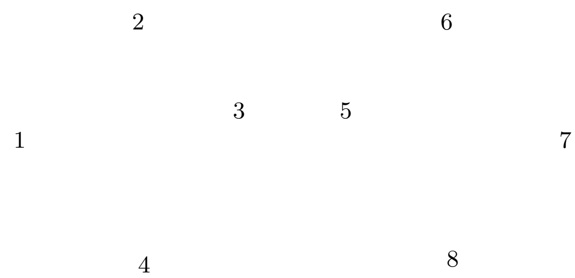


Figure 8.15 An oriented graph D.

A graph is a **proper interval graph** if it is the intersection graph of an inclusion-free family of intervals on the real line.

(a) Prove that, if D has a straight enumeration, then D is an acyclic round local tournament digraph.

(b) Prove that an undirected graph G is a proper interval graph if and only if it has a straight orientation. Hint: compare this with Theorem 8.1.6.

8.10. **(+) Recognizing locally semicomplete digraphs in $O(n^2)$ time**. Extend the results from Section 8.2 to locally semicomplete digraphs. Hint: use Exercise 8.3.(Bang-Jensen, Hell and Huang [76]).

8.11. **Recognizing non-strong locally semicomplete digraphs in linear time**. Give a simple linear algorithm to recognize non-strong locally semicomplete digraphs based on Theorem 4.11.6 (Bang-Jensen, Hell and Huang [76]).

8.12. **(+) Adjacencies between induced cycles in graphs that are representable in unicyclic graphs**. Suppose that the undirected graph G is representable in the unicyclic graph H. Prove that, if C is an induced cycle of length at least 4 in G, then every vertex x of C is adjacent to at least one vertex from every induced cycle of length at least 4 in $G - x$.

8.13. Derive Theorem 8.4.1 from Theorem 8.4.3.

8.14. **(+) Acyclic orientations such that every vertex is on an (s,t)-path**. Let $G = (V, E)$ be an undirected graph. Let s, t be special vertices and assume that, if G has a cut-vertex, then every cut-vertex v of G separates $G - v$ into two connected components, one containing s and one containing t. Prove that G has an acyclic orientation D such that every vertex of D is on an (s,t)-path (Gerards and Shepherd [313]).

8.15. **Strong orientations of the Petersen graph contain an even cycle**. Prove that every strongly connected orientation of the Petersen graph has an even cycle.

8.16. **Strong orientations of odd-K_4's and odd necklaces contain even cycles**. Prove Lemma 8.3.2.

8.17. **Undirected graphs without even cycles.** Describe the structure of those connected undirected graphs that have no even cycle.

8.18. **Graphs with strong orientations without even cycles and with the maximum number of vertices**. Prove that the graph L_n defined in Section 8.3 has a strong orientation without even cycles.

8.19. (−) Prove that Theorem 8.4.3 implies that every strong tournament has a hamiltonian cycle.

8.20. **3-colouring the Petersen graph**. Find an orientation of the Petersen graph which has no directed path of length 3. Use this to find a 3-colouring of the Petersen graph by colouring as in the proof of Theorem 8.4.1.

8.21. Figure 8.16 shows a graph G known as the Grötzsch graph. Prove that every orientation of G has a path of length 3. Find an orientation D of G such that $\mathrm{lp}(D) = 3$. Finally, show that, if e is any edge of G, then we can find an orientation of $G - e$ with no path of length 3.

Figure 8.16 The Grötzsch graph.

8.22. Prove that, if a connected graph G has a k-flow (D, x) for some k, then D is strongly connected.

8.23. Prove that a cubic graph is 3-edge-connected if and only if it is 3-connected.

8.24. (+) Prove that the Petersen graph has no 4-flow.

8.25. **Hamiltonian graphs have a 4-flow**. Prove that every hamiltonian graph has a 4-flow. Hint: use Theorem 8.5.6.

8.26. Find a 4-flow in the cubic graph in Figure 8.17.

8.27. **Converting a $\mathcal{Z}_k$-flow to a k-flow.** The proof of Theorem 8.5.3 gives rise to a polynomial algorithm to convert a given $\mathcal{Z}_k$-flow to a k-flow. Describe such an algorithm and illustrate it by converting the $\mathcal{Z}_5$-flow in the Petersen graph in Figure 8.18 to a 5-flow.

8.28. (+) Prove Theorem 8.5.6. Hint: define a $\mathcal{Z}_2 \times \mathcal{Z}_2$-flow from G_1, G_2 and vice versa.

8.29. Show that that the complete graph on 4 vertices is 3-edge-colourable and has no 3-flow.

8.30. (+) **Three spanning trees with no common edges in graphs which are 3-edge-connected**. Prove that every 3-edge-connected graph has 3

Figure 8.17 A hamiltonian cubic graph

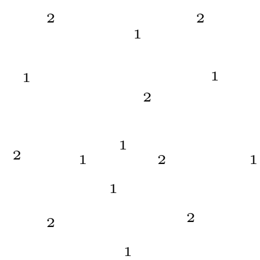


Figure 8.18 A $\mathcal{Z}_5$-flow in the Petersen graph.

spanning trees T_1, T_2, T_3 with the property that $E(T_1) \cap E(T_2) \cap E(T_3) = \emptyset$. Hint: use Theorem 9.5.5.

8.31. **(+) Jaeger's 8-flow theorem**. Prove, without using Theorem 8.5.10, that every 2-edge-connected graph G has an 8-flow. Hint: first observe that it suffices to prove the statement for 3-edge-connected graphs. By Exercise 8.30, G has three spanning trees such that no edge lies in all of these. Use this to construct a $\mathcal{Z}_2 \times \mathcal{Z}_2 \times \mathcal{Z}_2$-flow in G (compare this with the proof of Theorem 8.5.7).

8.32. **A minimum counterexample to Tutte's 5-flow conjecture has no 3-cycle**. Show that, if G is cubic 3-edge-connected and C is a 3-cycle of G, then the graph H obtained by contracting C to one vertex v in G and deleting the loops created is also cubic and 3-edge-connected. Show that every 5-flow in H can be extended to a 5-flow in G.

8.33. Show by an example that the idea of Exercise 8.32 does not always work for cycles longer than 5.

8.34. **(−) Nash-Williams' strong orientation theorem for eulerian multigraphs**. Prove Theorem 8.6.4 for eulerian graphs. Hint: consider an eulerian tour.

8.35. **Almost balanced k-arc-connected orientations**. Prove the following slight extension of Nash-Williams' orientation theorem. If $G = (V, E)$ is $2k$-connected, then it has a k-arc-strong orientation D such that $\max\{|d^+_D(x) -$

$d^-_D(x)| : x \in V(D)\} \leq 1$. Hint: follow the proof of Theorem 8.6.3 and change it appropriately when needed.

8.36. **(+) Vertices of degree k in minimally k-edge-connected graphs**. Prove that every minimally k-edge-connected graph contains a vertex of degree k. Hint: use the results analogous to Proposition 7.1.1 for undirected graphs.

8.37. **(+) Lovász's splitting theorem for undirected edge-connectivity**. Prove Theorem 8.6.1. Hint: define a set of vertices X not containing the special vertex s to be **k-dangerous** if $d(X) \leq k+1$. Clearly a splitting (su, sv) preserves k-edge-connectivity unless there is some k-dangerous set $X \subset V$ with $u, v \in X$. Observe that the degree function of an undirected graph has properties analogous to Proposition 7.1.1. Use this to show that there are at most two distinct maximal k-dangerous sets X, Y which contain a given neighbour t of s. Let X, Y be distinct maximal k-dangerous sets containing t but not s if such sets exist. Otherwise, either let X be the unique maximal k-dangerous set containing t but not s and $Y = \emptyset$ or, if no k-dangerous sets containing t but not s exists, then take $X = Y = \emptyset$. Conclude that s has a neighbour t' in $V - (X \cup Y)$ and show that (st, st') is an admissible splitting.

8.38. **(+) Splittings that do not create cut-edges.** Prove the following result due to Fleischner [239]. If G is a 2-edge-connected undirected graph and s is a vertex of degree at least 4, then there exist neighbours u, v of s such that replacing the edges su, sv by one edge uv results in a graph which is 2-edge-connected. Hint: this follows from Theorem 8.6.1 if $d_G(s)$ is even. If $d_G(s)$ is odd, then study maximal 2-dangerous sets containing neighbours of s (see also the hints for Exercise 8.37).

8.39. **(+) A polynomial algorithm for finding a k-arc-strong orientation of a $2k$-edge-connected multigraph**. Convert the proof of Theorem 8.6.3 into a polynomial algorithm which finds a k-arc-strong orientation of an arbitrary input multigraph, or outputs a proof that none exists.

8.40. Prove Corollary 8.7.4.

8.41. (+) Show how to derive Theorem 8.7.1 from Theorem 3.8.4.

8.42. Show how to convert the proof of Theorem 8.7.3 into a polynomial algorithm which either finds an orientation with the desired property, or a set violating the corresponding necessary condition.

8.43. Show how to derive Theorem 8.7.3 using the approach taken in the beginning of Subsection 8.7.1 and Exercise 3.32.

8.44. Prove that Theorem 8.7.3 implies Hall's theorem (Theorem 3.11.3).

8.45. Prove that the condition in Conjecture 8.6.7 is necessary in order to have a k-strong orientation.

8.46. **Reversing the orientation of a cycle preserves arc-strong connectivity**. Prove that, if D is k-arc-strong and C is a cycle in D, then the digraph obtained by reversing the orientation of all arcs on C is also k-arc-strong.

8.47. **(+) Converting one k-arc-strong orientation into another via reversal of cycles**. Suppose that D and D' are k-arc-strong orientations of

$G = (V, E)$ and that $d^-_D(v) = d^-_{D'}(v)$ for every $v \in V$. Prove that one can obtain D' from D by successive reversals of the orientation of a cycle in the current digraph.

8.48. **Reversal of a path while preserving k-arc-strong connectivity**. Suppose that D and D' are k-arc-strong orientations of a graph G and that there exists a vertex u such that $d^-_D(u) < d^-_{D'}(u)$. Show that D' contains a vertex v such that $d^-_D(v) > d^-_{D'}(v)$ and a (u, v)-path P. Under what conditions can we obtain a new k-arc-strong orientation of G by reversing the arcs of P?

8.49. **(+) Finding a good path to reverse.** Suppose that D and D' are k-arc-strong orientations of a graph G and that there exists a vertex u such that $d^-_D(u) < d^-_{D'}(u)$. Prove that there is always a vertex v such that $d^-_D(v) > d^-_{D'}(v)$ and a (u, v)-path P such that one can reverse all arcs of P without destroying the k-arc-strong connectivity. Hint: use your observation in Exercise 8.48. Assume that all paths are bad. Use submodularity of d^-_D to show that the maximal sets $X_1, X_2, \ldots, X_h$ containing v but not u and which have in-degree k in D are pairwise disjoint. Count those arcs that have at most one end vertex in $\cup_{i=1}^h X_i$ in both D and D' and obtain a contradiction (Frank [253]).

8.50. **Proof of Theorem 8.8.9**. Combine your observations in Exercises 8.47, 8.48 and 8.49 into a proof of Theorem 8.8.9.

8.51. Show that Theorem 8.6.3 is a special case of Theorem 8.7.6.

8.52. Let $D = (V, A)$ be a digraph and $x : A \to \mathcal{R}$ a function on the arc set of D. Show that the function $x^-(U) - x^+(U)$ is a modular function.

8.53. (+) Prove Theorem 8.9.1. Hint: use a similar approach to that used in Section 8.8.4 to prove Theorem 8.6.3 via submodular flows.

8.54. Construct $2k$-arc-strong mixed graphs with no k-arc-strong orientation. Hint: they must violate the condition in Theorem 8.9.1.

8.55. Prove directly that the condition (8.48) is necessary for the existence of an orientation satisfying (8.47). Hint: assume that D is an orientation which satisfies (8.47) and study which edges are counted by the sum $\sum_{j=1}^{q_i} d^-_D(X_{ij})$.

8.56. **(+) Orienting a mixed graph to be k-arc-strong**. Use Theorem 8.9.2 to derive a necessary and sufficient condition for a mixed graph $M = (V, A, E)$ to have a k-arc-strong orientation (Frank [259, 263]).

8.57. **(+) Orientations containing k-arc-disjoint out-branchings from a given root.** Let $G = (V, E)$ be an undirected graph with a special vertex $s \in V$ and let k be a natural number. Prove without using Theorem 8.7.6 that G has an orientation such that $d^-(X) \geq k$ for every $X \subseteq V - s$ if and only if (9.5) holds (Frank [260]).

8.58. **(+) Orienting a mixed graph in order to obtain many arc-disjoint branchings.** Consider the problem of finding an orientation of a mixed graph $M = (V, A, E)$ so that it has k arc-disjoint out-branchings rooted at a specifiedvertex s or concluding that no such orientation exist. Show how to reduce this problem to a submodular flow problem. Argue that you can also solve the minimum cost version where there may be different costs on the two possible orientations of an edge $e \in E$.

8.59. **(+) Arc-disjoint in- and out-branchings with a fixed root in orientations of graphs.** Describe an algorithm to decide whether a given undirected graph $G = (V, E)$ has an orientation D such that there exist arc-disjoint in- and out-branchings F_v^+, F_u^- where $u, v \in V$ are specified (not necessarily distinct) vertices of V. Prove that the corresponding problem for mixed graphs is $\mathcal{NP}$-complete. Hint: use Theorem 9.9.2.

8.60. **(−)** Characterize when an undirected graph $G = (V, E)$ has an orientation so that x, y are in the same strong component for specified distinct vertices $x, y \in V$.

8.61. **Orienting a mixed graph so as to get a closed trail containing two specified vertices.** Show that the following problem is $\mathcal{NP}$-complete: Given a mixed graph $M = (V, A, E)$ and two distinct vertices s, t. Decide if M has an orientation that contains are-disjoint (s, t)-, (t, s)-paths.

8.62. Let $M = (V, A, E)$ be a mixed graph and let $D = (V, A)$ be the directed part of M. Prove that for every k the function $k - d_D^-$ is crossing G-supermodular. Hint: use the fact that d_D^- is submodular.

8.63. Show how to minimize the submodular function b defined by (8.35) and (8.36) over a given collection of subsets in polynomial time. Hint: use flows to determine the in-degrees of the relevant sets.

8.64. Let k be a natural number and let $G = (V, E)$ be a graph with a cost function c that for every edge $e \in E$ assigns a cost to each of the two possible orientations of e. Show how to formulate the problem of finding a k-arc-strong orientation of G of minimum cost with respect to c as a minimum cost submodular flow problem.

8.65. **Reversing arcs in order to get many arc-disjoint out-branchings from a fixed root.** Show how to solve the following problem using submodular flows. Given a directed multigraph $D = (V, A)$, a vertex $s \in V$ and a natural number k. Determine whether it is possible to reverse the orientation of some arcs in A such that the resulting directed multigraph has k arc-disjoint out-branchings rooted at s. Argue that one can also solve the minimum cost version of the problem in polynomial time.

8.66. Derive Theorem 8.7.6 from the feasibility theorem for crossing submodular flows (Theorem 8.8.5).

9. Disjoint Paths and Trees

In this chapter we concentrate on problems concerning (arc)-disjoint paths or trees (arborescences). We embark from the 2-path problem which concerns the existence of two disjoint paths with prescribed initial and terminal vertices. We give a proof by Fortune et al. showing that the 2-path problem is $\mathcal{NP}$-complete. We proceed by studying the more general k-path problem for various classes of digraphs. We show that for acyclic digraphs, the k-path problem is polynomially solvable when k is not a part of the input. Then we describe several results on the k-path problem for generalizations of tournaments. Among other results, we show that the 2-path problem is polynomially solvable for digraphs that can be obtained from strong semicomplete digraphs by substituting arbitrary digraphs for each vertex of the semicomplete digraph. We briefly discuss the k-path problem for planar digraphs and indicate how to use the topological concept of planarity in proofs and algorithms for disjoint path problems in planar digraphs.

The next major topic is arc-disjoint branchings. We prove Edmonds' famous branching theorem and show many consequences of this very important and useful result. After discussing some related problems on branchings, we move on to arc-disjoint path problems. We show that the arc-disjoint version of the k-path problem is also $\mathcal{NP}$-complete as soon as k is at least 2. Using a nice observation due to Shiloach we show that, if a digraph does not contain two arc-disjoint paths, one from u to v and the other from x to y, for every choice of u, v, x, y, then D is not 2-arc-strong. Results on arc-disjoint paths in generalizations of tournaments as well as eulerian digraphs are described. We point out how the structural characterizations for non-2-linked eulerian digraphs resemble those for the analogous problems for undirected graphs.

We consider arc-disjoint in- and out-branchings and show that the problem to decide whether a digraph has arc-disjoint branchings F_v^+, F_v^- such that F_v^- is an in-branching rooted at v and F_v^+ is an out-branching rooted at v is $\mathcal{NP}$-complete. We describe a complete solution, due to Bang-Jensen, of this problem for tournaments and indicate how the complexity version of the problem (for tournaments) is closely related to problems concerning weak linkings in tournaments. Namely, there is a polynomial algorithm for the branching problem which uses polynomial algorithms for two weak linking problems as subroutines.

Finally, we discuss the problem of finding a minimum cost branching with a given root in a weighted digraph. We describe a generalization of this problem which also covers the case when one starts from a digraph which has k but not $k+1$ arc-disjoint branchings from a given root s and the goal is to add as few new arcs as possible in order to obtain a new digraph which has $k+1$ arc-disjoint out-branchings rooted at s. Then we give an algorithm due to Frank and Fulkerson for solving this more general version and show how the algorithm works when we apply it to the minimum cost branching problem.

9.1 Additional Definitions

Recall that an out-branching (in-branching) rooted at a vertex s in a digraph D is a spanning oriented tree T which is oriented in such a way that every vertex $x \neq s$ has $d^-_T(x) = 1$ ($d^+_T(x) = 1$). In this chapter we will also consider the following generalization of a branching. An **out-arborescence** rooted at s is an oriented tree T which is not necessarily spanning such that $s \in V(T)$ and every vertex $x \in V(T) - s$ has $d^-_T(x) = 1$. An in-arborescence with root s is defined analogously.

Recall from Chapter 7 that for a digraph $D = (V, A)$ with distinct vertices x, y we denote by $\kappa_D(x, y)$ the largest integer k such that D contains k internally disjoint (x, y)-paths. By Menger's theorem $\kappa_D(x, y)$ equals the size of a minimal (x, y)-separator.

When discussing intersections between paths P, Q we will often use the phrase 'let u be the first (last) vertex on P which is on Q'. By this we mean that if, say, P is an (x, y)-path, then u is the only vertex of $P[x, u]$ ($P[u, y]$) which is also on Q.

In some sections it is also convenient to use the notation that for a given set of arcs F and a set of vertices X of a digraph D we denote by $d^-_F(X)$, respectively $d^+_F(X)$, the number of arcs from F that enter, respectively leave, X (hence $d^-_F(X)$ is shorthand for $d^-_{D\langle F\rangle}(X)$).

Let $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$ be (not necessarily distinct) vertices of a directed multigraph D. A **(weak)** $\boldsymbol{k}$**-linking** from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ in D is a system of (arc-)disjoint paths $P_1, P_2, \ldots, P_k$ such that P_i is an (x_i, y_i)-path in D. By 'disjoint' we mean that no P_i contains any of the vertices x_j, y_j as internal vertices for $j \neq i$ (but paths may share one of both of their end-vertices). Note that in a weak k-linking the only restriction is that the paths are arc-disjoint. A directed multigraph $D = (V, A)$ is **(weakly)** $\boldsymbol{k}$**-linked** if it contains a (weak) k-linking from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ for every choice of not necessarily distinct vertices $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$. A digraph D is $\boldsymbol{k}$**-(arc)-cyclic** if it has a cycle containing the vertices (arcs) $x_1, x_2, \ldots, x_k$ ($a_1, a_2, \ldots, a_k$) for every choice of k vertices (arcs).

Recall that an $[x, y]$-path in a directed multigraph is a path which is either an (x, y)-path or a (y, x)-path.

9.2 Disjoint Path Problems

The general problem we will consider here is the existence of certain paths which are (arc)-disjoint and have specified or contain specified internal vertices. There is a close relation between path and cycle problems as can be seen from the following complexity statement. The proof is left to the reader as Exercise 9.1.

Proposition 9.2.1 *For general digraphs the following problems are equivalent from a computational point of view (that is, if one is polynomially solvable or $\mathcal{NP}$-complete, then so are each of the others).*

- *(P1) Given four distinct vertices u_1, u_2, v_1, v_2 in a digraph D. Decide whether or not D has disjoint paths connecting u_1 to v_1 and u_2 to v_2. We call this* **the 2-path problem**.
- *(P2) Given two distinct arcs e_1, e_2 in a digraph D. Decide whether or not D has a cycle through e_1 and e_2.*
- *(P3) Given two distinct vertices u and v in a digraph D. Decide whether or not D has a cycle through u and v.*
- *(P4) Given two distinct vertices u and v in a digraph D. Decide whether or not D has disjoint cycles C_x, C_y such that $x \in C_x$ and $y \in C_y$.*
- *(P5) Given three distinct vertices x, y, z. Decide whether D has an (x, z)-path which also contains the vertex y.*

We prove in Theorem 9.2.3 that the 2-path problem is $\mathcal{NP}$-complete. Hence it follows from Proposition 9.2.1 that all the problems mentioned in Proposition 9.2.1 are $\mathcal{NP}$-complete.

It is interesting to note that although problems (P1)-(P5) are all very hard for general digraphs, the difficulty of these problems may vary considerably for some classes of digraphs. For instance problem (P3) is trivial for tournaments (and the more general locally semicomplete digraphs) since such a cycle exists if and only if x and y are in the same strong component of D. Problem (P4) is also easy for semicomplete digraphs, since such cycles exist if and only if there exist disjoint 3-cycles C, C' one containing x and the other containing y (Exercise 9.14). However problems (P1) and (P2) are considerably more difficult to prove polynomial for tournaments (see Theorem 9.3.12). Note that (P2) and also (P5) may be considered as special cases of (P1) if we drop the requirement that the vertices must be distinct in (P1).

The following generalization of the 2-path problem is known as the ***k*-path problem.** Given a digraph D and distinct vertices $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$. Does D have a collection of disjoint paths $P_1, P_2, \ldots, P_k$ such that P_i is an (x_i, y_i)-path, $i = 1, 2, \dots, k$? The result below shows that it suffices to consider distinct vertices when proving that a digraph is k-linked (the proof is left as Exercise 9.3).

Proposition 9.2.2 *For every $k \geq 1$ a digraph $D = (V, A)$ is k-linked if and only if it contains disjoint (x_i, y_i)-paths $P_1, P_2, \dots, P_k$ for every choice of distinct vertices $x_1, x_2, \dots, x_k, y_1, y_2, \dots, y_k$.* □

Below we study the k-path problem. We start by showing that the problem is $\mathcal{NP}$-complete already when $k = 2$. We show that there is no degree of vertex-strong connectivity which guarantees the existence of such paths. Then in succeeding sections we go on to special classes of graphs such as acyclic digraphs and generalizations of tournaments. There the reader will see that quite a lot can be said about the problem and that it still is not trivial for these classes of digraphs.

9.2.1 The Complexity of the k-Path Problem

We start with the following result by Fortune, Hopcroft and Wyllie showing that already for $k = 2$ the k-path problem is very difficult for general digraphs.

Theorem 9.2.3 *[247] The 2-path problem is $\mathcal{NP}$-complete.*

Since this theorem is very important and the gadget construction[1] used in the proof is quite illustrative, we give the proof in detail below. We follow the proof in [247].

First we need a lemma whose proof is left as Exercise 9.4.

Lemma 9.2.4 *[247] Consider the digraph S shown in Figure 9.1 (a). Suppose there are two disjoint paths P, Q passing through S such that P leaves S at A and Q enters S at B. Then P must enter S at C and Q must leave S at D. Furthermore, there exists exactly one more path R passing through S which is disjoint from P, Q and this is either*

$$(8, 9, 10, 4, 11) \qquad \textit{or} \qquad (8', 9', 10', 4', 11'),$$

depending on the actual routing of P. □

The digraph S in Figure 9.1 is called a **switch**. We can stack arbitrarily many switches on top of each other and still have the conclusion on Lemma 9.2.4 holding for each switch. The way we stack is simply by identifying the C and D arcs of one switch with the A and B arcs of the next (see Figure 9.2). A switch can be represented schematically as in Figure 9.1(c), or, when we want to indicate stacking of switches, as in Figure 9.1(b).

[1] Quite often $\mathcal{NP}$-completeness proofs are constructed by piecing together certain **gadgets** about which one can prove certain properties. Based on these properties one then shows that the whole construction has the desired properties. For other instances of this technique, see e.g. Chapter 11.

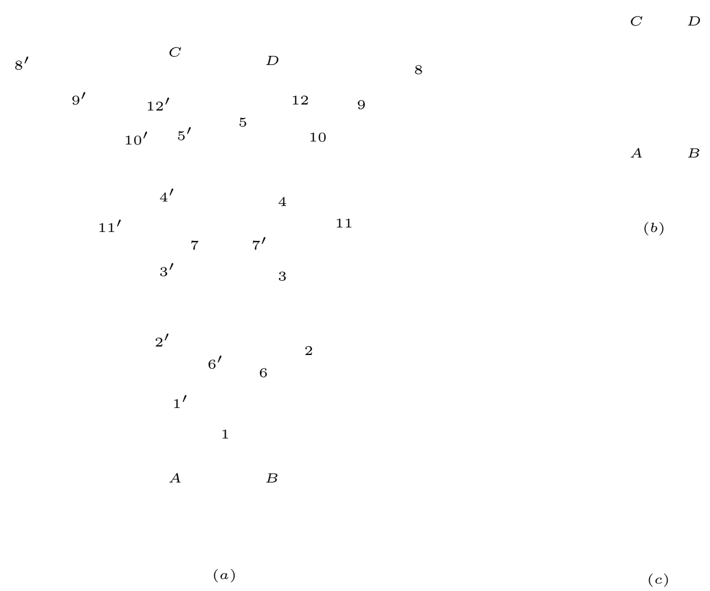


Figure 9.1 Part (a) shows a switch S. Part (b) and (c) show schematic pictures of a switch ([247, Fig. 1]). In (c) the two vertical arcs correspond to the paths (8,9,10,4,11), respectively , (8',9',10',4',11'). Note that for convenience, we label the arcs, rather than the vertices in this Figure.

Proof of Theorem 9.2.3: The reduction is from 3-SAT (recall the definition from Section 1.10). Let $\mathcal{F} = C_1 * C_2 * \ldots * C_r$ be an instance of 3-SAT with variables $x_1, x_2, \ldots, x_k$. For each variable x_i we let H_i be the digraph consisting of two internally disjoint (u,v)-paths of length r (the number of clauses in $\mathcal{F}$). We associate one of these paths with the literal x_i and the other with the literal $\overline{x}_i$. We are now ready to explain the construction of the digraph $D[\mathcal{F}]$ and show that it contains disjoint (u_1,v_1)-, (u_2,v_2)-paths if and only if $\mathcal{F}$ is satisfiable.

See Figure 9.3. We form a chain $H_1 \to H_2 \to \ldots \to H_k$ on the subdigraphs corresponding to each variable (see the middle of the figure, H_i corresponds to the variable x_i). With each clause C_i we associate three switches, one for each literal it contains. The left paths of these switches (that is, the paths in the left hand part of the figure) all start at the vertex n_{i-1} and end at n_i. The right path of each switch is substituted for a (private) arc of H_i such that the arc is taken from the path which corresponds to x_i if the literal is x_i and from the path which corresponds to $\overline{x}_i$ if the literal is $\overline{x}_i$. The substitution is shown for the clause $C_i = x_1 + \overline{x}_2 + x_5$ in the figure. By the choice of the lengths of the paths in H_i we can make this substitution so that different arcs in H_i are substituted by different switches corresponding to several clauses, all of which contain the literal x_i or $\overline{x}_i$. The switches corresponding to the

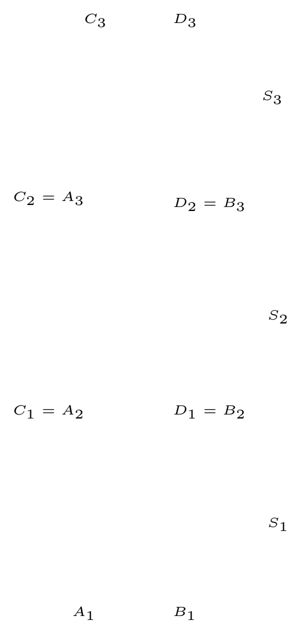


Figure 9.2 Stacking 3 switches on top of each other.

clause C_i are denoted $S_{i,1}, S_{i,2}, S_{i,3}$. We stack these switches in the order $S_{1,1}S_{1,2}S_{1,3} \ldots S_{r,1}, S_{r,2}S_{r,3}$ as shown in the right part of the figure. A two way arc between a clause and some H_j (shown only for C_i) indicates a switch that is substituted for these arcs. Note that this is the same switch which is shown in the right hand side of the figure! Finally, we join the D arc of the switch $S_{r,3}$ to the vertex z_1 of H_1, add an arc from w_k in H_k to n_0 and choose vertices u_1, u_2, v_1, v_2 as shown (that is, u_2 is the tail of the C arc for $S_{r,3}$, u_1 is the tail of the B arc of $S_{1,1}$ and v_2 is the head of the A arc of $S_{1,1}$). This completes the description of $D[\mathcal{F}]$.

We claim that $D[\mathcal{F}]$ contains disjoint (u_1, v_1)-, (u_2, v_2)-paths if and only if $\mathcal{F}$ is satisfiable.

Suppose first that $D[\mathcal{F}]$ has disjoint (u_1, v_1)-, (u_2, v_2)-paths P, Q. It follows from the definition of $D[\mathcal{F}]$ that the paths P and Q will use all the arcs that go between two switches (i.e. those arcs that are explicitly shown in the right hand side of Figure 9.3). Hence, by Lemma 9.2.4, after removing the arcs of Q and the arcs of P from u_1 to the first vertex z_1 of H_1, the only remaining way to pass through a switch $S_{i,j}$ is to use either the right path or the left path of $S_{i,j}$ but not both! By the construction of $D[\mathcal{F}]$, P must traverse the subdigraphs corresponding to the variables in the order $H_1, H_2, \ldots, H_k$ and each time P uses precisely one of the two paths in H_i

Figure 9.3 A schematic picture of the digraph $D[\mathcal{F}]$.

(recall again that some of the arcs in H_i in Figure 9.3 correspond to the right path of some switch). Let T be the truth assignment which sets $x_i := 1$ if P uses the path corresponding to $\overline{x}_i$ and let $x_i := 0$ in the opposite case. We show that this is a satisfying truth assignment for $\mathcal{F}$.

It follows from the construction of $D[\mathcal{F}]$ and the remark above on arcs used by Q and the first part of P from u_1 to H_1 that the path P contains all the vertices $n_0, n_1, \ldots, n_r$ in that order. Since each of the paths from n_j to n_{j+1} are part of a switch for every $j = 0, 1, \ldots r-1$, we must use the left path of precisely one of these switches to go from n_j to n_{j+1}. By Lemma 9.2.4, every time we use a left path of a switch, the right path cannot also be used. From this we see that for each clause C_j, $j = 1, 2, \ldots r$, it must be the case that at least one of the literals y (in particular the one whose left path we could use) of C_j becomes satisfied by our truth assignment. This follows because P must use the path corresponding to $\overline{y}$ in the middle. Thus we have shown that $\mathcal{F}$ is satisfiable.

Suppose now that T' is a satisfying truth assignment for $\mathcal{F}$. Then for every variable x_i which is true (false) we can use the subpath corresponding to $\overline{x}_i$ (x_i) in H_i. For each clause C_j we can fix one literal which is true and use the left path of the switch that corresponds to that literal (that path cannot be blocked by the way we chose subpaths inside the H_i's). By Lemma 9.2.4 we can find disjoint paths P, Q such that P starts in u_1 and ends in the initial vertex z_1 of H_1 and Q is a (u_2, v_2)-path in the right part of $D[\mathcal{F}]$. Furthermore, by the same lemma, after removing the vertices of P and Q, we still have the desired paths corresponding to each literal available. This shows that we can route the disjoint (u_1, v_1)-, (u_2, v_2)-paths in $D[\mathcal{F}]$. □

The digraph $D[\mathcal{F}]$ above is not strongly connected and one may ask whether the problem becomes easier if we require high vertex-strong connectivity. However, using Theorem 9.2.3 Thomassen [710] proved that the 2-path problem remains $\mathcal{NP}$-complete even for highly connected digraphs.

Lynch proved that for undirected graphs the k-path problem is $\mathcal{NP}$-complete when k is part of the input [529].The case $k = 2$ was proved to be polynomially solvable by Seymour [662], Shiloach [670] and Thomassen [697] and a complete characterization was obtained by Seymour [662] and Thomassen [697]. The results in [662, 697] (see also Jung's paper [470]) imply that every 6-connected undirected graph is 2-linked (see also the remark at the end of Section 9.4). For fixed $k \geq 3$ the k-path problem is also polynomially solvable [642]. This is just one of many important consequences of the deep work of Robertson and Seymour on Graph Minors. The interesting thing is that [642] only proves the existence of an $O(n^3)$ algorithm for fixed k (the constant depending heavily on k). To date no actual algorithm has been given, even in the case $k = 3$.

The following result due to Thomassen shows that for directed graphs the situation is quite different from the undirected case. Namely, there is no degree of vertex-strong connectivity which will guarantee a directed graph to be 2-linked.

Theorem 9.2.5 *[710] For every natural number k there exists an infinite family of k-strong and non-2-linked digraphs D_k.* □

In fact, Thomassen proved that even for the special case of cycles through two fixed vertices (Problem (P3) of Proposition 9.2.1) no degree of vertex-strong connectivity suffices to guarantee such a cycle. Recall that a digraph $D = (V, A)$ is 2-cyclic if it has a cycle containing x, y for every choice of distinct vertices $x, y \in V$.

Theorem 9.2.6 *[710] For every natural number k there exists an infinite family of k-strong digraphs D'_k which are not 2-cyclic.* □

9.2.2 Sufficient Conditions for a Digraph to be k-Linked

In this section we briefly discuss some sufficient conditions for a digraph to be k-linked for some (prescribed) k. Not surprisingly, if a digraph has sufficiently many arcs then it is k-linked. The next result due to Manoussakis shows that digraphs which are close to being complete are k-linked. The proof is left as Exercise 9.5.

Theorem 9.2.7 *[545] Let $D = (V, A)$ be a digraph of order n and let k be an integer such that $n \geq 2k \geq 2$. If $|A| \geq n(n-2) + 2k$ then D is k-linked.* □

The proof of Theorem 9.2.7 in [545] is based on the following lemma.

Lemma 9.2.8 *[545] If $D-x$ is k-linked for some vertex $x \in V$ which satisfies $d^+(x), d^-(x) \geq 2k-1$, then D is k-linked.*

Proof: Let $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k \in V(D)$ be an arbitrary collection of terminals. We wish to prove that D contains internally disjoint paths $P_1, P_2, \ldots, P_k$ where P_i is an (x_i, y_i)-path for $i = 1, 2, \ldots, k$. By the assumption that $D-x$ is k-linked, it suffices to consider the case when $x = x_i$ for some i or $x = y_j$ for some j. Since x is one of the terminals, it follows that among the $2k$ terminals at most $2k-1$ of these are out-neighbours (in-neighbours) of x.

Since a path from x to an out-neighbour u of x can be taken to be just the arc xu and hence cannot interfere with the other paths we wish to find, we may assume that, if $x = x_i$ for some i, then $y_i \not\in N^+(x)$ and similarly if $x = y_j$ for some j then $x_j \not\in N^-(x)$. Let T denote the set of distinct terminals. Now it is easy to see that for every desired path P_i starting at x we may choose a private member u_i of $N^+(x) - T$ and replace x_i by $x'_i = u_i$. Similarly for every desired path P_j ending at x we may choose a private member v_j of $N^-(x) - T$ and replace y_j by $y'_j = v_j$. If x'_r (y'_s) was not introduced by the replacements above we let $x'_r = x_r$ ($y'_s = y_s$). Now the existence of the desired linking follows by taking a k-linking in $D - x$ for the set of terminals $x'_1, x'_2, \ldots, x'_k, y'_1, y'_2, \ldots, y'_k$. □

The requirement on the number of arcs in Theorem 9.2.7 is very strong and hence the result is not very useful. However Manoussakis showed by an example that the number of arcs in Theorem 9.2.7 is best possible [545].

The next result, due to Heydemann and Sotteau, shows that for 2-linkings one can also get a sufficient condition in terms of $\delta^0(D)$. The proof is easy and is left as Exercise 9.6. See also Theorem 9.2.10 below.

Theorem 9.2.9 *[426] If a digraph D satisfies $\delta^0(D) \geq n/2 + 1$, then D is 2-linked.* □

The condition above is still quite restrictive and one would expect a stronger result to hold. Examples from [426] show that we cannot weaken the degree condition. However, we can strengthen the result in the following way.

Theorem 9.2.10 *If a digraph D satisfies $\delta^0(D) \geq n/2 + 1$, then for every choice of distinct vertices $x, y, u, v \in V$, D contains internally disjoint (x, y)-, (u, v)-paths P, Q such that $V(P) \cup V(Q) = V$.*

Proof: Let $X = V - \{x, y, u, v\}$ and construct D' from $D - \{x, y, u, v\}$ by adding two new vertices p and q such that

$$N^-_{D'}(p) = N^-_D(v) \cap X, N^+_{D'}(p) = N^+_D(x) \cap X,$$
$$N^-_{D'}(q) = N^-_D(y) \cap X, N^+_{D'}(q) = N^+_D(u) \cap X.$$

It is easy to see that for every $w \in V - \{x, y, u, v\}$, $d^-_{D'}(w) \geq d^-_D(w) - 2$ and $d^+_{D'}(w) \geq d^+_D(w) - 2$. Hence the resulting digraph D' which has $n' = n - 2$ vertices satisfies $\delta^0(D') \geq n'/2$. By Corollary 5.6.3, D' has a hamiltonian cycle C. Let p^+, q^+ (p^-, q^-) denote the successors (predecessors) of p, q on C. Then $xC[p^+, q^-]y$ and $uC[q^+, p^-]v$ are the desired paths which cover V. □

Theorem 9.2.9 was extended by Manoussakis to 3-linkings.

Theorem 9.2.11 *[545] If a digraph D has $n \geq 9$ vertices and $\delta^0(D) \geq n/2 + 2$, then D is 3-linked.* □

Based on Theorems 9.2.9 and 9.2.11, Manoussakis posed the following problem. Note that $f(n, k) \leq n - 1$, since the complete digraph is k-linked.

Problem 9.2.12 *[545] Determine the minimum function $f(n, k)$ such that every digraph D on n vertices which satisfies $\delta^0(D) \geq f(n, k)$ is k-linked.*

According to Manoussakis [545], Hurkens proved that $f(n, 4) = n/2 + 3$ when $n \geq 13$ and Manoussakis mentions that perhaps $f(n, k) \leq n/2 + k - 1$ holds for $n \geq 4k - 3$.

Let us conclude this section with a result in connection with problem (P3) of Proposition 9.2.1. It is easy to see that, if a digraph is 2-linked, then it is also 2-arc-cyclic and hence 2-cyclic. Heydemann and Sotteau proved that, if we only want a digraph to be 2-cyclic, then it is possible to weaken the condition in Theorem 9.2.7 somewhat.

Theorem 9.2.13 *[426] Every strong digraph $D = (V, A)$ with $\delta^0(D) \geq 2$ and $|A| \geq n^2 - 5n + 15$ is 2-cyclic.* □

9.2.3 The k-Path Problem for Acyclic Digraphs

When the digraph considered is acyclic there is enough structure to allow an efficient solution of the k-path problem for every fixed k Perl and Shiloach [602] proved that the 2-path problem is polynomially solvable for acyclic digraphs. In their elegant proof they showed how to reduce the 2-path problem for a given acyclic digraph to a simple path finding problem in another digraph. Fortune, Hopcroft and Wyllie extended Perl and Shiloach's result to arbitrary k. The proof of this result below is an extension of the proof by Perl and Shiloach (see also Thomassen's survey [707]).

Theorem 9.2.14 *[247] For each fixed k, the k-path problem is polynomially solvable for acyclic digraphs.*

Proof: Let $D = (V, A)$ be acyclic and let $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$ be distinct vertices of D for which we wish to find a k-linking from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$. We may assume that $d^-_D(x_i) = d^+_D(y_i) = 0$ for $i = 1, 2, \ldots, k$, since such arcs play no role in the problem and can therefore be deleted.

Form a new digraph $D' = (V', A')$ whose vertex set is the set of all k-tuples of distinct vertices of V. For any such k-tuple $(v_1, v_2, \ldots, v_k)$ there is at least one vertex, say v_r, which cannot be reached by any of the other v_i by a path in D. (Here we used that D is acyclic.) For each out-neighbour w of v_r such that $w \notin \{v_1, v_2, \ldots, v_k\}$, we let A' contain the arc $(v_1, v_2, \ldots, v_{r-1}, v_r, v_{r+1}, \ldots, v_k) \rightarrow (v_1, v_2, \ldots, v_{r-1}, w, v_{r+1}, \ldots, v_k)$. Only arcs as those described above are in A'.

We claim that D' has a directed path from the vertex $(x_1, x_2, \ldots, x_k)$ to the vertex $(y_1, y_2, \ldots, y_k)$ if and only if D contains disjoint paths $P_1, P_2, \ldots, P_k$ such that P_i is an (x_i, y_i)-path for $i = 1, 2, \ldots, k$.

Suppose first that D' has a path P from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$. By definition, every arc of P corresponds to one arc in D. Hence we get a collection of paths $P_1, P_2, \ldots, P_k$ such that P_i is an (x_i, y_i)-path by letting P_i contain those arcs that correspond to a shift in the ith vertex of a k-tuple, $i = 1, 2, \ldots, k$. Suppose two of these paths, P_i, P_j are not disjoint. Then it follows from the assumption that $d^-_D(x_i) = d^+_D(y_i) = 0$ for $i = 1, 2, \ldots, k$ and the definition of D' that there is some vertex $u \in V - \{x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k\}$ such that $u \in V(P_i) \cap V(P_j)$. Let w (z) be the predecessor of u on P_i (P_j). We may assume without loss of generality that the arc on P corresponding to wu is used before that corresponding to zu. This means that at the time we change from w to u in the ith coordinate, the jth coordinate corresponds to a vertex z' which can reach u in D (through z). Now it follows from the definition of the arcs in A' that we could not have changed the ith coordinate again before we have used the arc corresponding to zu in D'. However that would lead to a k-tuple which contains two copies of the same vertex u from D, contradicting the definition of D'. Hence P_i and P_j must be disjoint.

Suppose now that D contains disjoint paths $Q_1, Q_2, \ldots, Q_k$ such that Q_i is an (x_i, y_i)-path, $i = 1, 2, \ldots, k$. Then we can construct a path from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ in D' as follows. Start with the tuple $(x_1, x_2, \ldots, x_k)$. At any time we choose a coordinate j of the current k-tuple $(z_1, z_2, \ldots, z_k)$ such that the vertex z_j is not in $\{y_1, y_2, \ldots, y_k\}$ and z_j cannot be reached in D by any other vertex from the tuple. Note that such a vertex exists since D is acyclic and $d^+(y_i) = 0$ for $i = 1, 2, \ldots, k$. It is easy to show by induction that we will always have $z_j \in V(Q_j)$. Now we use the arc $z_j w$ corresponding to the arc out of z_j on Q_j and change the j'th coordinate from z_j to w. If follows from the fact that $Q_1, \ldots, Q_k$ are disjoint that this will produce a path from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ in D'.

Given any instance $(D, x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k)$ we can produce the digraph D' in time $O(k!n^{k+2})$ by forming all possible k-tuples and deciding which arcs to add based on the definition of D'. Then we can decide the existence of a path from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ in polynomial time using BFS in D'. This proves that the k-path problem is polynomial for each fixed k. $\square$

Note that we don't actually have to construct D' in advance. It suffices to introduce the vertices and arcs when they become relevant for the search for a path from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$ in D'.

It is not difficult to see that we can also use the approach above to find the cheapest collection of k disjoint paths where the ith path is an (x_i, y_i)-path in a given acyclic digraph with non-negative weights on the arcs. Here the goal is to minimize the total weight of the arcs used by the paths (see Exercise 9.9).

Suppose that D is an acyclic graph and v is a vertex of in-degree 1. Let u be the unique in-neighbour of v. Then the digraph $D' = D//uv$ which we obtain by path-contracting the arc uv is also acyclic. Furthermore, contracting such an arc can have no effect on the existence of a certain linkage in the digraph since only one path in such a linkage may enter the vertex v. This shows that we may assume that all vertices except the terminals have in- and out-degree at least 2 when considering the 2-linkage problem (and more generally the k-linkage problem) for acyclic graphs. Furthermore we may assume that no arc enters x_i and no arc leaves y_i, $i = 1, 2$.

It is also easy to see that given any acyclic digraph D with distinct vertices x_1, x_2, y_1, y_2 we may in polynomial time either decide the existence of disjoint (x_1, y_1)-, (x_2, y_2)-paths, or obtain a new reduced digraph D^* such that $d^-_{D^*}(x_1) = d^-_{D^*}(x_2) = d^+_{D^*}(y_1) = d^+_{D^*}(y_2) = 0$, every other vertex has in- and out-degree at least 2 in D^* and D^* has the desired paths if and only if D has such paths. Hence, from a computational point of view, the following result due to Thomassen completely solves the 2-path problem for acyclic digraphs.

Theorem 9.2.15 *[704] Let D be an acyclic digraph on at least 5 vertices with vertices x_1, x_2, y_1, y_2 such that $d^-(x_1) = d^-(x_2) = d^+(y_1) = d^+(y_2) = 0$ and every other vertex has in- and out-degree at least 2. Suppose D does not contain disjoint (x_1, y_1)-, (x_2, y_2)-paths. Let H denote the digraph one obtains from D by adding two new vertices x_0, y_0 and the arcs $x_0x_1, x_0x_2, y_1y_0, y_2y_0, x_1y_2, x_2y_1$. Then H can be drawn in the plane such that the outer cycle is formed by the two paths $x_0x_1y_2y_0, x_0x_2y_1y_0$ and every other facial cycle[2] is the union of two directed paths in H (see Figure 9.4).*

[2] A cycle C in a plane graph G is **facial** with respect to a planar drawing of G if C is the boundary of some face.

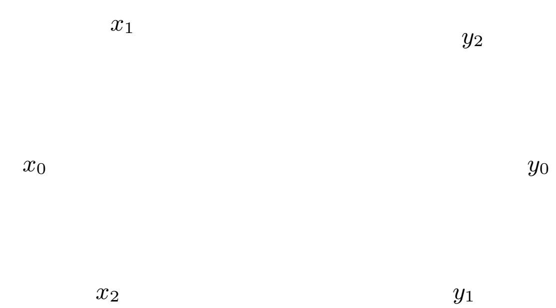


Figure 9.4 The digraph H obtained from D by adding x_0, y_0 and arcs x_0x_1, x_0x_2, y_1y_0, y_2y_0, x_1y_2, x_2y_1 (shown as fat arcs).

Theorem 9.2.15 was generalized by Metzlar [563]. The following interesting connection between the 2-path problem for undirected graphs and the 2-path problem for acyclic digraphs is a corollary of Theorem 9.2.15.

Corollary 9.2.16 *[704] Let $D = (V, A)$ be an acyclic digraph and suppose that the vertices x_1, x_2, y_1, y_2 are all distinct and satisfy that $d^-(x_i) = d^+(y_i) = 0$ for $i = 1, 2$ and that all other vertices of D have in- and out-degree at least 2. Then D contains disjoint (x_1, y_1)-, (x_2, y_2)-paths if and only if $UG(D)$ contains such paths.* □

Thomassen [704] mentioned that it would be interesting to have a direct proof of Corollary 9.2.16. Such a proof was given by Lucchesi and Giglio in [527]. In that paper the connection between the 2-path problem for acyclic digraphs and the 2-path problem for undirected graphs was studied. It was shown that there is a very close connection between the two problems.

The example in Figure 9.5 shows that Corollary 9.2.16 has no analogue when $k > 2$.

9.3 Linkings in Tournaments and Generalizations of Tournaments

We now turn to linking problems for tournaments and their generalizations. It turns out that for semicomplete digraphs enough structure is present to allow a polynomial algorithm for the 2-path problem (Theorem 9.3.12). We show in Subsection 9.3.3 that this algorithm can be used as a subroutine in a polynomial algorithm for the 2-path problem for a large super class of the semicomplete digraphs.

We start out with some sufficient conditions in terms of the degree of (local) strong connectivity.

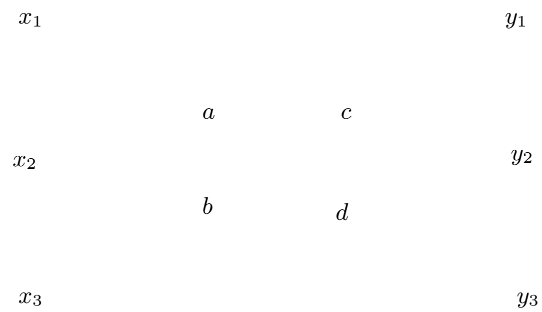


Figure 9.5 An acyclic digraph D in which every non-special vertex has in- and out-degree at least 2. There does not exist disjoint paths P_1, P_2, P_3 such that P_i is an (x_i, y_i)-path, $i = 1, 2, 3$. However $UG(D)$ has such paths.

9.3.1 Sufficient Conditions in Terms of (Local-)Connectivity

The following proposition was proved by Thomassen [701] in the case when D is a tournament. By inspection of the proof in [701] one sees that the only place there where it is used that one is dealing with a tournament, rather than an arbitrary digraph, is to be sure that there is an arc between every successor of x and every predecessor of y on the paths $P_1, \ldots, P_p$ below. Hence we can state and prove Thomassen's result in the following much stronger form:

Proposition 9.3.1 *[52, 701] Let D be a digraph and x, y, u, v distinct vertices of D such that $\kappa(u, v) \geq q + 2$ and $P_1, \ldots, P_p$ are internally disjoint (x, y)-paths such that the subdigraph $D\langle V(P_1) \cup \ldots \cup V(P_p)\rangle$ has no (x, y)-path of length less than or equal to 3 and such that the successor of x on P_i is adjacent to the predecessor of y on P_j for all $i, j \in \{1, 2, \ldots, p\}$. Then D has q internally disjoint (u, v)-paths, the union of which intersects at most $2q$ of the paths $P_1, \ldots, P_p$.*

Proof: We may assume that $p \geq 2q + 1$ since otherwise the claim is trivially true. Let $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_q\}$ be internally disjoint (u, v)-paths in $D - \{x, y\}$. We define two collections of subpaths of the paths in $\mathcal{Q}$ as follows (in Exercise 9.15 the reader is asked to describe an algorithm for constructing such collections starting from $\mathcal{Q}$).

Let $Q'_1, Q'_2, \ldots, Q'_q$ be chosen such that either $Q'_i = Q_i$ or $Q'_i = Q[u, z]$ for some vertex $z \in V(P_j)$ where $j \in \{1, 2, \ldots, p\}$ and $P_j[z, y]$ has only the vertex z in common with $U = V(Q'_1) \cup \ldots \cup V(Q'_q)$. We also assume that $|U|$ is minimum subject to the conditions above. If some path P_r contains a vertex w from U and $P_r[w, y]$ contains no vertices from $U - w$, then the minimality of U implies that one of the paths $Q'_1, Q'_2, \ldots, Q'_q$ terminates in w. This implies that the collection $Q'_1, Q'_2, \ldots, Q'_q$ intersects at most q of the paths $P_1, P_2, \ldots, P_p$.

Analogously we can define a collection $Q''_1, Q''_2, \ldots, Q''_q$ where Q''_i is either Q_i or $Q''_i = Q_i[w, v]$ for a vertex w on some P_k satisfying that $P_k[x, w]$ contains only the vertex w from $V(Q''_1) \cup \ldots \cup V(Q''_q)$ and such that $Q''_1, Q''_2, \ldots, Q''_q$ intersect at most q of the paths $P_1, P_2, \ldots, P_p$.

Now we construct the desired paths as follows. For each $i = 1, 2, \ldots, q$, if $Q'_i = Q_i$ or $Q''_i = Q_i$, then let $R_i := Q_i$. Otherwise let z be the terminal vertex of Q'_i, let w be the initial vertex of Q''_i and let r, j be chosen such that $z \in V(P_j), w \in V(P_r)$. Let x' (y') be the successor (predecessor) of x (y) on P_r (P_j). By the assumption that D contains no (x, y)-path of length 3 and that every successor of x is adjacent to every predecessor of y on the paths $P_1, \ldots, P_p$, we get that $y'x' \in A$. Let $R_i := Q'_i P_j[z, y'] P_r[x', w] Q''_i$ (see Figure 9.6).

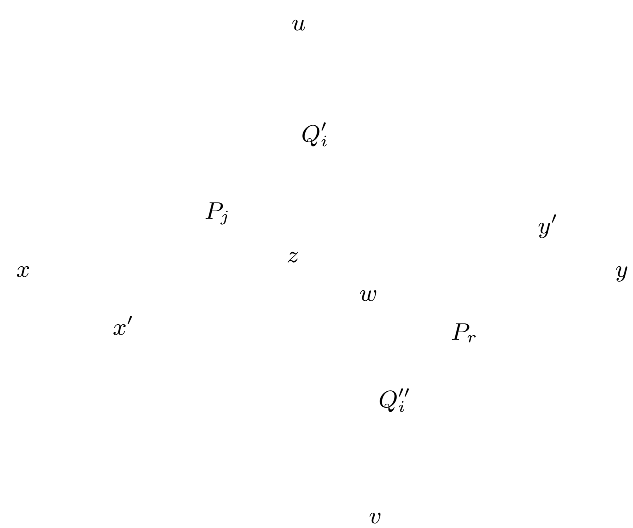


Figure 9.6 How to obtain R_i from Q'_i, Q''_i, P_j and P_r. The fat arcs indicate the resulting (u, v)-path.

Now $R_1, R_2, \ldots, R_q$ are internally disjoint (u, v)-paths and by construction they contain no more than $2q$ vertices from the paths $P_1, P_2, \ldots, P_p$. □

Our proof above is constructive and can easily be turned into a fast algorithm for finding the desired collection of paths (Exercise 9.16). The following result by Thomassen is an easy corollary

Corollary 9.3.2 *[701] Every 5-strong semicomplete digraph is 2-linked.*

Proof: Let D be a 5-strong semicomplete digraph and let x_1, x_2, y_1, y_2 be arbitrary distinct vertices of D. If $D - \{x_{3-i}, y_{3-i}\}$ has an (x_i, y_i)-path P of length at most 3 for $i = 1$ or $i = 2$, then $D - P$ is strong and hence contains an (x_{3-i}, y_{3-i})-path. Hence we may assume that every (x_i, y_i)-path in $D - \{x_{3-i}, y_{3-i}\}$ has length at least 4 for $i = 1, 2$.

Let P_1, P_2, P_3 be internally disjoint (x_1, y_1)-paths in $D - \{x_2, y_2\}$. Then D and these paths satisfy the assumption of Theorem 9.3.1 for $q = 1$ and it follows that D has an (x_2, y_2)-path which intersects at most two of the paths P_1, P_2, P_3. Since x_1, x_2, y_1, y_2 were chosen arbitrarily, it follows from Lemma 9.2.2 that D is 2-linked. □

Bang-Jensen [43] constructed the 4-strong semicomplete digraph in Figure 9.7, hence showing that 5-strong connectivity is best possible for general semicomplete digraphs.

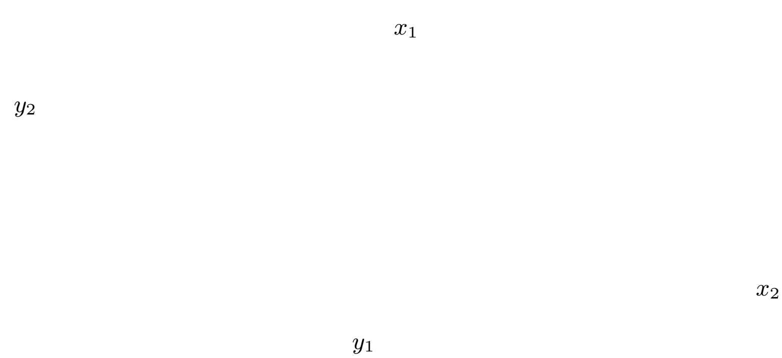


Figure 9.7 A 4-strong non-2-linked semicomplete digraph T. All arcs not shown go from left to right and $x_1y_2x_1, x_2y_1x_2$ are the only 2-cycles in T. There is no pair of disjoint (x_1, y_1)-,(x_2, y_2)-paths in T. The tournament which results from T by deleting the arcs y_2x_1 and y_1x_2 is also 4-strong.

We now turn our attention to special classes of generalizations of tournaments. The first lemma shows that for the class of round decomposable locally semicomplete digraphs one can improve the bound from Corollary 9.3.2. The proof is left as Exercise 9.20.

Lemma 9.3.3 *[52] For each natural number k, every $(3k-2)$-strong round decomposable locally semicomplete digraph is k-linked.* □

In order to get a result on k-linkings for locally semicomplete digraphs that are not round decomposable we use the following lemma which allows us to apply Proposition 9.3.1. Recall that by Exercise 4.33, $\alpha(D) \leq 2$ if D is locally semicomplete but not round decomposable.

Lemma 9.3.4 *[52] Let x and y be distinct vertices in a locally semicomplete digraph D such that $\alpha(D) \leq 2$ and let $P_1, \ldots, P_p$ be internally disjoint (x,y)-paths such that the locally semicomplete digraph $D' = D\langle V(P_1) \cup \ldots \cup V(P_p)\rangle$ has no (x,y)-path of length less than 6. Then for all $1 \leq i, j \leq p$, the predecessor u of y on P_i dominates the successor v of x on P_j.*

Proof: We may assume that each P_i is a minimal (x,y)-path. Suppose there exist i and j such that the predecessor u of y on P_i is not adjacent to the successor v of x on P_j. Note that the assumption of the lemma and Exercise 9.18 implies that $y \rightarrow x$. Therefore D' is strong and we conclude from Exercise 9.18 (applied to u, v) that D' contains an (x,y)-path of length at most 5, contradicting the assumption. Hence $u \rightarrow v$ must hold. $\square$

The following theorem by Bang-Jensen gives a sufficient condition for the existence of a specified k-linking in a locally semicomplete digraph which is not round decomposable in terms of local connectivities. It generalizes a result by Thomassen for tournaments [701]. Bang-Jensen also proved an analogous result for quasi-transitive digraphs, see [52] for details.

Theorem 9.3.5 *[52] There exists, for each natural number k, a natural number $f(k)$ such that the following holds. If D is a locally semicomplete digraph with $\alpha(D) \leq 2$ and $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$ are distinct vertices in D such that $\kappa(x_i, y_i) \geq f(k)$ for all $i = 1, \ldots, k$, then D has disjoint paths $P_1, P_2, \ldots, P_k$ where P_i is an (x_i, y_i)-path, $i = 1, 2, \ldots, k$.*

Proof: Let $f(1) = 1$ and $f(k) = 2(k-1)f(k-1) + 2k + 1$ for $k \geq 2$. We prove by induction on k that this choice works for f. This is clear for $k = 1$, so we proceed to the induction step assuming $k \geq 2$. Suppose that $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$ are distinct vertices in a locally semicomplete digraph D for which $\alpha(D) \leq 2$ and assume that $\kappa(x_i, y_i) \geq 2(k-1)f(k-1) + 2k + 1$ for all $i = 1, \ldots, k$. We prove that $D - \{x_2, \ldots, x_k, y_2, \ldots, y_k\}$ has an (x_1, y_1)-path P_1 such that $\kappa_H(x_i, y_i) \geq f(k-1)$ for $i = 2, \ldots, k$, where $H = D - V(P_1)$. Then the result follows by induction. If $D - \{x_2, \ldots, x_k, y_2, \ldots, y_k\}$ has an (x_1, y_1)-path of length at most 5, then this can play the role of P_1, so assume that no such path exists. Let $Q_1, Q_2, \ldots, Q_{2(k-1)f(k-1)+1}$ be internally disjoint (x_1, y_1)-paths in $D - \{x_2, \ldots, x_k, y_2, \ldots, y_k\}$. We show that one of these can play the role of P_1. First note that by Lemma 9.3.4 and the remark above, we have that for all $1 \leq i, j \leq 2(k-1)f(k-1) + 1$ the predecessor of y_1 on Q_i dominates the successor of x_1 on Q_j. Hence, by Proposition 9.3.1, for each $i = 2, 3, \ldots, k$, there are internally disjoint (x_i, y_i)-paths $P_{1,i}, P_{2,i}, \ldots, P_{f(k-1),i}$ which together intersect at most $2f(k-1)$ of the paths $Q_1, Q_2, \ldots, Q_{2(k-1)f(k-1)+1}$. Hence there is at least one path Q_r which intersects none of $P_{j,i}$, $2 \leq i \leq k$, $1 \leq j \leq f(k-1)$. Thus we can use that Q_r as P_1. $\square$

Combining Lemma 9.3.3, Theorem 9.3.5 and Theorem 4.11.15 we obtain the following result by Bang-Jensen (extending a similar result for semicomplete digraphs by Thomassen [701]):

Theorem 9.3.6 *[52] There exists, for each natural number k, a natural number $f(k)$ such that every $f(k)$-strong locally semicomplete digraph is k-linked.*

Here and below the function $f(k)$ is the function which is defined in the proof of Theorem 9.3.5.

Corollary 9.3.7 *[52] Every $f(k)$-strong locally semicomplete digraph is k-arc-cyclic.*

The function $f(k)$ is probably far from best possible for Theorem 9.3.6 and Corollary 9.3.7. In particular, $f(2) = 7$, but, using Theorem 4.11.15, it should be possible to prove that the following holds.

Conjecture 9.3.8 *[52] Every 5-strong locally semicomplete digraph is 2-linked.*

9.3.2 The 2-Path Problem for Semicomplete Digraphs

In the proof of Corollary 9.3.2 we really only used that $\kappa_{T-\{x_i,y_i\}}(x_{3-i}, y_{3-i})$ was at least 3 for $i = 1, 2$ in order to ensure the existence of three internally disjoint (x_1, y_1)-paths in $D - \{x_2, y_2\}$ and then we applied Proposition 9.3.1. Bang-Jensen strengthened this sufficient condition as follows.

Theorem 9.3.9 *[43] Let T be a semicomplete digraph and let x_1, x_2, y_1, y_2 be distinct vertices of T. Suppose that*

$$\min\{\kappa_{T-\{x_2,y_2\}}(x_1, y_1), \kappa_{T-\{x_1,y_1\}}(x_2, y_2)\} \geq 2 \text{ and}$$

$$\max\{\kappa_{T-\{x_2,y_2\}}(x_1, y_1), \kappa_{T-\{x_1,y_1\}}(x_2, y_2)\} \geq 3,$$

then T has a pair of disjoint (x_1, y_1)-, (x_2, y_2)-paths. □

This is best possible with respect to local connectivities. The semicomplete digraph in Figure 9.7 shows that we cannot replace 3 by 2 above. However, see Theorem 9.3.13 for a special case where we can do this.

Bang-Jensen showed that for cycles through two arcs (the special case when $y_1 \rightarrow x_2$ and $y_2 \rightarrow x_1$), we can strengthen Corollary 9.3.2 in the case of tournaments. For semicomplete digraphs the digraph in Figure 9.7 shows that we cannot always weaken the connectivity requirement.

Theorem 9.3.10 *[43] Every 3-strong tournament and every 5-strong semicomplete digraph is 2-arc-cyclic.* □

It follows from the proof of Theorem 9.3.10 in [43] that for a fixed pair of arcs e, e' we can replace the connectivity requirement that D is 5-strong by $(5-i)$-strong provided that i of the arcs e, e' are not in a 2-cycle ($i = 1, 2$).

Conjecture 9.3.11 *[52] Every 3-strong locally tournament digraph is 2-arc-cyclic.*

The example in Figure 9.7 indicates that finding a complete generalization of those semicomplete digraphs that do not have disjoint (x,y)-, (u,v)-paths for a given set of distinct vertices x, y, u, v may be very difficult. In the special case where we allow u and y to be equal, that is, we are seeking an (x,v)-path which passes through the vertex u (that is, the problem (P5) in Proposition 9.2.1), it is indeed possible to give a characterization. Such a characterization was given by Bang-Jensen in [45].

From the algorithmic point of view, the 2-path problem for semicomplete digraphs was solved by Bang-Jensen and Thomassen who proved the following result:

Theorem 9.3.12 *[89] The 2-path problem is solvable in time $O(n^5)$ for semicomplete digraphs.* □

The proof of this result in [89] is highly non-trivial. The basic approach is divide and conquer. However, several non-trivial results and steps are needed to make the algorithm work. We state the most important of these results below since it is of independent interest.

Recall from Chapter 6 that an (s,t)-separator S is trivial if t has in-degree zero, or s has out-degree zero in $D - S$. The following result which complements Theorem 9.3.9 is very important for the proof of correctness of the algorithm of Bang-Jensen and Thomassen, since it corresponds to a case where no problem reduction is possible (using the approach taken in the algorithm).

Theorem 9.3.13 *[89] Let T be a semicomplete digraph, and let x_1, x_2, y_1, y_2 be distinct vertices of T, such that for each $i = 1, 2$, there are two, but not three, internally disjoint (x_i, y_i)-paths in $T - \{x_{3-i}, y_{3-i}\}$. Suppose that all (x_i, y_i)-separators of size 2 in $T - \{x_{3-i}, y_{3-i}\}$ are trivial, for $i = 1, 2$. Then T has a pair of disjoint (x_1, y_1)-, (x_2, y_2)-paths. Furthermore such a pair of paths can be constructed in time $O(n^3)$.*

Note that the semicomplete digraph in Figure 9.7 does not satisfy the assumption of Theorem 9.3.13 since the two non-labeled vertices in the middle form a non-trivial (x_2, y_2)-separator of size 2 in $T - \{x_1, y_1\}$.

9.3.3 The 2-Path Problem for Generalizations of Tournaments

Now we show that the 2-path problem can be solved in polynomial time for quite large classes of digraphs which can be obtained starting from semicomplete digraphs and then performing certain substitutions. The algorithm we describe uses the polynomial algorithm from Theorem 9.3.12 for the case of semicomplete digraphs as a subroutine. The results in this section are due to Bang-Jensen [52].

Theorem 9.3.14 *[52] Let $D = F[S_1, S_2, \ldots, S_f]$ where F is a strong digraph on $f \geq 2$ vertices and each S_i is a digraph with n_i vertices and let x_1, x_2, y_1, y_2 be distinct vertices of D. There exists semicomplete digraphs $T_1, \ldots, T_f$ such that $V(T_i) = V(S_i)$, $i = 1, 2, \ldots, f$, and the digraph $D' = F[T_1, T_2, \ldots, T_f]$ has vertex-disjoint (x_1, y_1)-, (x_2, y_2)-paths if and only if D has such paths. Furthermore, given D and x_1, x_2, y_1, y_2, D' can be constructed in time $O(n^2)$, where n is the number of vertices of D.*

Proof: If D has the desired paths, then so does any digraph obtained from D by adding arcs. Hence if D has the desired paths, then trivially D' exists and can be constructed in time $O(n^2)$ once we know a pair of disjoint (x_1, y_1)-, (x_2, y_2)-paths.

If no S_i contains both of x_1, y_1 or both of x_2, y_2, then it is easy to see that D has the desired paths if and only if it has such paths which do not use an arc inside any S_j. Thus in this case we can add arcs arbitrarily inside each S_i to obtain a D' which satisfies the requirement.

Suppose next that some S_i contains all of the vertices x_1, x_2, y_1, y_2. If there is an (x_j, y_j)-path P in $S_i - \{x_{3-j}, y_{3-j}\}$, $j = 1$ or 2, then it follows from that fact that F is strong that D has the desired paths and we can find such a pair in time $O(n^2)$. Thus, by our initial remark, we may assume that there is no (x_j, y_j)-path P in $S_i - \{x_{3-j}, y_{3-j}\}$ for $j = 1, 2$. Now it is easy to see that D has the desired paths if and only if it has such paths which do not use an arc inside any S_j. Thus we can replace S_i with a tournament in which x_1 and x_2 both have no out-neighbours in $S_i - \{x_1, x_2\}$ and every other S_k by an arbitrary tournament on the same vertex set. Clearly the digraph D' obtained in this way satisfies the requirement.

Suppose now without loss of generality that $x_1, y_1 \in V(S_j)$ for some j but $x_2 \notin V(S_j)$. Suppose first that $y_2 \in V(S_j)$. If there is no (x_1, y_1)-path in $S_j - y_2$, then D has the desired paths if and only if it has such paths which do not use an arc inside any S_i and we can construct D' by adding arcs in S_j in such a way that no (x_1, y_1)-path avoiding y_2 is created (that is, y_2 will still separate x_1 from y_1 in $D'\langle V(S_j)\rangle$) and arbitrary arcs in every other S_i. On the other hand if $S_j - y_2$ contains an (x_1, y_1)-path avoiding y_2, then it follows from the fact that F is strong that D has the desired paths and hence D' exists as remarked above. Hence we may assume that $y_2 \notin V(S_j)$.

If S_j contains an (x_1, y_1)-path which does not cover all the vertices of S_j, then it follows from the fact that F is strong that D has the desired paths. Thus we may assume that either S_j has no (x_1, y_1)-path, or every (x_1, y_1)-path in S_j contains all the vertices of S_j. In the last case we may assume that $V(S_j)$ separates x_2 from y_2. Now D has the desired paths if and only if it has such a pair which does not use any arcs from S_j. Thus in both cases we can construct D' by replacing S_j by a tournament with no (x_1, y_1)-path and every other S_i by an arbitrary tournament on the same vertex set, except in the case when x_2 and y_2 belong to some S_i, $i \neq j$. In this case we replace that S_i by a tournament with no (x_2, y_2)-path (by the remark above we may assume that S_i has no (x_2, y_2)-path).

It follows from the considerations above that D' can be constructed in time $O(n^2)$. □

Recall that quasi-transitive digraphs can be decomposed according to Theorem 4.8.5. Hence we can apply Theorem 9.3.14 to these digraphs.

Theorem 9.3.15 *[52] There exists a polynomial algorithm for the 2-path problem for quasi-transitive digraphs.*

Proof: Let D be a quasi-transitive digraph and x_1, x_2, y_1, y_2 specified distinct vertices for which we want to determine the existence of vertex-disjoint (x_1, y_1)-,(x_2, y_2)-paths. First check that $D - \{x_i, y_i\}$ contains an (x_{3-i}, y_{3-i})-path for $i = 1, 2$. If not then we stop. Now it follows from Theorem 4.8.5 that either x_1, x_2, y_1, y_2 are all in the same strong component of D, or the paths exist. For example, if D is not strong and y_1, say, is not in the same strong component as x_1 then, by Theorem 4.8.5, x_1 and y_1 belong to different sets W_i, W_j in the canonical decomposition $D = Q[W_1, \ldots, W_{|Q|}]$, where Q is a transitive digraph. Hence $x_1 \rightarrow y_1$ and the desired paths clearly exist.

Thus we may assume that D is strong. Let $D = S[W_1, W_2, \ldots, W_{|S|}]$ be a decomposition of D according to Theorem 4.8.5. Now apply Theorem 9.3.14 and construct the digraph D' which has the desired paths if and only if D does. As remarked in Theorem 9.3.14, D' can be constructed in polynomial time. By the construction of D' (replacing each W_i by a semicomplete digraph) it follows that D' is a semicomplete digraph and hence we can apply the polynomial algorithm of Theorem 9.3.12 to D' in order to decide the existence of the desired paths in D. The algorithm of Theorem 9.3.12 can be used to find vertex-disjoint (x_1, y_1)-, (x_2, y_2)-paths in D' if they exist and given these paths it is easy to construct the corresponding paths in D (it suffices to take minimal paths). □

By inspecting the proof of Theorem 9.3.14 it is not difficult to see that the following much more general result is true. The main point is that in the proof of Theorem 9.3.14 we either find the desired paths or decide that they exist if and only if there are such paths that use no arcs inside any S_i. Hence instead of making each T_i semicomplete, we may just as well make it an independent set, by deleting all arcs inside S_i.

Theorem 9.3.16 *[52] Let Φ be a class of strongly connected digraphs, let Φ_0 denote the class of all extensions of graphs in Φ and let*

$$\Phi^* = \{F[D_1, \ldots, D_{|F|}] : F \in \Phi, \text{ each } D_i \text{ is an arbitrary digraph}\}.$$

There is a polynomial algorithm for the 2-path problem in Φ^* if and only if there is a polynomial algorithm for the 2-path problem for all digraphs in Φ_0. □

This result shows that studying extensions of digraphs can be quite useful.

One example of such a class Φ, for which Theorem 9.3.16 applies, is the class of strong semicomplete digraphs. This follows from the fact that we can reduce the 2-path problem for extended semicomplete digraphs to the case of semicomplete digraphs in the same way as we did for quasi-transitive digraphs in the proof of Theorem 9.3.15. Hence the 2-path problem is polynomially solvable for all digraphs that can be obtained from strong semicomplete digraphs by substituting arbitrary digraphs for vertices. It is important to note here that Φ must consist only of strong digraphs, since it is not difficult to reduce the 2-path problem for arbitrary digraphs (which is $\mathcal{NP}$-complete by Theorem 9.2.3) to the 2-path problem for those digraphs that can be obtained from the digraph H consisting of just an arc uv by substituting arbitrary digraphs for the vertex v.

The proof of the following easy lemma is left to the reader as Exercise 9.21. Note that four is best possible as can be seen from the complete biorientation of the undirected graph consisting of 4-cycle $x_1x_2y_1y_2x_1$ and a vertex z joined to each of the four other vertices.

Lemma 9.3.17 *Let D be a digraph of the form $D = \vec{C}_2[S_1, S_2]$, where S_i is an arbitrary digraph on n_i vertices, $i = 1, 2$. If D is 4-strong then D is 2-linked.* □

The following result generalizes Corollary 9.3.2.

Theorem 9.3.18 *[52] Let $k \geq 4$ be a natural number and let F be a digraph on $f \geq 2$ vertices with the property that every k-strongly connected digraph of the form $F[T_1, T_2, \ldots, T_f]$, where T_i, $i = 1, 2, \ldots, f$, is a semicomplete digraph, is 2-linked. Let $D = F[S_1, S_2, \ldots, S_f]$, where S_i is an arbitrary digraph on n_i vertices, $i = 1, 2, \ldots, f$. If D is k-strongly connected, then D is 2-linked.*

Proof: Let $D = F[S_1, S_2, \ldots, S_f]$, where S_i is an arbitrary digraph on n_i vertices, $i = 1, 2, \ldots, f$, be given. By Lemma 9.3.17 we may assume that D cannot be decomposed as $D = \vec{C}_2[R_1, R_2]$, where R_1 and R_2 are arbitrary digraphs. Construct D' as described in Theorem 9.3.14. Note that by Lemma 7.13.1, $\kappa(D') = \kappa(D)$. Thus D' is k-strong and using Theorem 9.3.14 and the assumption of the theorem we conclude that D is 2-linked. □

Corollary 9.3.19 *[52] Every 5-strong quasi-transitive digraph is 2-linked.*

Proof: By Theorem 4.8.5, every strong quasi-transitive digraph is of the form $D = F[S_1, S_2, \ldots, S_f]$, $f = |F|$, where F is a strong semicomplete digraph and each S_i is a non-strong quasi-transitive digraph on n_i vertices. By Lemma

4.8.4 and the connectivity assumption, $|F| \geq 3$. Note that for any choice of semicomplete digraphs $T_1, \ldots, T_f$ the digraph $D' = F[T_1, T_2, \ldots, T_f]$ is semicomplete. Hence the claim follows from Theorem 9.3.18 and the fact that, by Corollary 9.3.2, every 5-strong semicomplete digraph is 2-linked. (Since F has at least three vertices, it follows from Lemma 7.13.1 that $\kappa(D') = \kappa(D)$.) $\square$

9.4 Linkings in Planar Digraphs

In this section we briefly discuss the k-path problem for planar digraphs (recall the definition of a planar digraph from Section 4.14). The constraint that the digraph in question can be embedded in the plane clearly poses some restrictions to the structure of disjoint paths. This is illustrated by the following result.

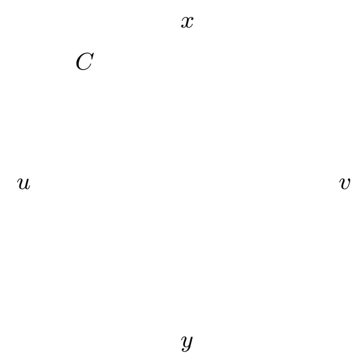


Figure 9.8 A topological obstruction for the existence of disjoint (x, y)- and (u, v)-paths in a planar graph G. The cycle C is the boundary of the outer face of G.

Proposition 9.4.1 *Suppose that $D = (V, A)$ is a planar digraph with distinct vertices $x, y, u, v \in V$ and that D is embedded in the plane in such a way that the vertices x, v, y, u appear on the bounding cycle C of the outer face in that order (see Figure 9.4.1). Then D does not have a pair of disjoint (x, y)-, (u, v)-paths.*

Proof: We first prove that no matter how we connect x and y by a simple (that is, not self intersecting) curve R and u, v by another simple curve R', both inside the bounded disc with boundary C (see Figure 9.8) the two curves must intersect. Suppose we can choose simple curves R, R' so that R connects x and y and R' connects u and v. Then we can add a new point z in the interior of the outer face and join it to each of the vertices x, y, u, v by disjoint simple curves which lie entirely in the closed disc formed by the outer face and its boundary C. This gives us an embedding of K_5 in the plane, contradicting Theorem 4.14.1.

Suppose now that P, Q are disjoint paths in D such that P is an (x, y)-path and Q is a (u, v)-path. In the embedding of D these correspond to simple curves and hence, by the argument above, they must intersect at some point in the plane. Since D is planar, no two arcs intersect in the interior (as curves) and hence we see that P and Q must intersect in some vertex v of D. However this contradicts the assumption that they are disjoint. $\square$

We point out that the first part of the proof above can be established using the Jordan curve theorem directly to establish that R and R' must intersect somewhere in the disc with boundary C (see e.g. the book by Bondy and Murty [127]).

It was shown by Lynch [529] that when k is part of the input, then the k-path problem remains $\mathcal{NP}$-complete even for planar digraphs. For fixed k, Schrijver has developed a polynomial algorithm.

Theorem 9.4.2 *[656, 657] For every fixed* [3] *k the k-path problem is polynomially solvable for planar digraphs.* $\square$

The proof method is based on cohomology over free (non-abelian) groups, a topic which would require too much space to cover in the present book. Schrijver mentions that part of the group theory and topology is mainly used to keep notation fairly simple, but in any case the proof is too complicated to include here even as a (convincing) sketch. For additional discussion on and applications (for digraphs embedded on surfaces) of this very powerful proof technique we refer the reader to Schrijver's papers [656, 657, 658]. We should mention though that arguments like those used in the proof of Proposition 9.4.1 play an important role in Schrijver's approach.

To further illustrate how to use planarity in arguments in disjoint path problems, we consider a special case of the k-path problem for which a good characterization for the existence of a prescribed linking has been found by Ding, Schrijver and Seymour [194].

Suppose that we are given a planar digraph $D = (V, A)$ which is embedded in the plane in such a way that the vertices $s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k$ all belong to the boundary of the outer face F of D. Ding, Schrijver and Seymour [194] proved that in this case there is a simple polynomial algorithm to decide the existence of a collection of disjoint paths $P_1, P_2, \ldots, P_k$, where P_i is an (s_i, t_i)-path, $i = 1, 2, \ldots, k$.

In fact, as we will see below, it turns out to be easier to describe an algorithm for the following slight extension of the problem: in addition to the

[3] That is, k is not part of the input.

vertices $s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k$ we are also given subsets $A_1, A_2, \ldots, A_k$ of A and we demand that P_i can only use[4] arcs from A_i for $i = 1, 2, \ldots, k$.

Motivated by the example in Figure 9.8 we say that two pairs of terminals (s_i, t_i) and (s_j, t_j) on $bd(F)$ **cross** if each simple curve from s_i to t_i in $\mathcal{R}^2 - F$ (considered as a subspace of $\mathcal{R}^2$) crosses each simple curve from s_j to t_j in $\mathcal{R}^2 - F$. By Proposition 9.4.1 a necessary condition for the existence of disjoint $(s_1, t_1), \ldots, (s_k, t_k)$-paths in D is that the following cross-freeness condition is satisfied:

$$\text{for every } i \neq j \ (s_i, t_i) \text{ and } (s_j, t_j) \text{ do not cross.} \tag{9.1}$$

Using the cross-freeness condition we see that there is no solution unless the terminals occur in the order $u_1, v_1, u_2, v_2, \ldots u_k, v_k$ around $bd(F)$, where $\{u_i, v_i\} = \{s_{\pi(i)}, t_{\pi(i)}\}$ for some permutation π of $\{1, 2, \ldots, k\}$. Clearly this condition can be checked in polynomial time if we are given the (polygonal) embedding of D.

We measure **closeness** of two polygonal paths with the same end-points by the area between the two paths. See Figure 9.9 for an illustration. The proof of the following lemma is left as Exercise 9.23.

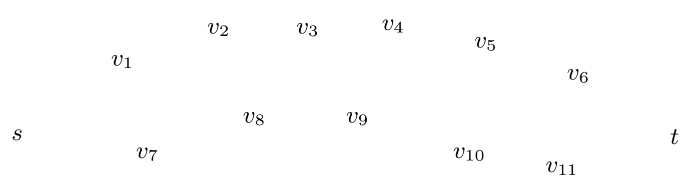


Figure 9.9 Let R be the path $sv_1v_2v_3v_4v_5v_6t$ in the underlying graph of D. The (s,t)-path $sv_7v_2v_9v_5v_6v_{11}t$ is closer to R than the (s,t)-path $sv_7v_8v_9v_5v_6v_{11}t$.

Lemma 9.4.3 *Let R be a path from x to y along the boundary of the outer face (ignoring the orientation of the arcs in D) and let D' be a subdigraph of D which contains the vertices x and y. Then either D' has no (x,y)-path or there exist a unique (x,y)-path Q in D' which is closest to R. Given the embedding of D, we can find Q in polynomial time if it exists. Furthermore, no other (x,y)-path 'crosses over' Q at any point (e.g. in Figure 9.9 the path $v_8v_9v_5$ crosses over the path $v_2v_9v_{10}$ at the vertex v_9).*

Now we are ready to describe a greedy algorithm which either finds the desired paths in D, or a proof that no such paths exist (using only arcs from the sets $A_1, A_2, \ldots, A_k$).

[4] In [194] Ding, Schrijver and Seymour consider an even more general case where not all paths linking different pairs of terminals must be disjoint, but for simplicity we assume that they are all disjoint.

Start with s_k, t_k. Since D satisfies the cross-freeness condition, one of the two paths between s_k and t_k along $bd(F)$ contains no other terminals. Denote this path by P.

If $D\langle A_k\rangle$ contains no (s_k, t_k)-path, then there is no solution, so assume below that such a path exists.

Let P_k be the unique (s_k, t_k)-path in $D\langle A_k\rangle$ which is closest to P. Modify A_i, $i = 1, 2, \ldots, k-1$ by removing from A_i every arc that is incident to a vertex on P_k. Now repeat the steps above for the pair s_{k-1}, t_{k-1} and continue recursively.

After at most k iterations we either find the required linking or conclude that no such linking exists.

To prove the correctness of the algorithm we observe that, if $Q_1, Q_2, \ldots, Q_k$ is a solution, then so is $Q_1, Q_2, \ldots, Q_{k-1}, P_k$. Indeed, if P_k intersects some Q_i, then so does Q_k because P_k is either equal to Q_k or strictly closer to P than Q_k. This shows that the greedy choice is legal and the correctness follows. It also follows from Lemma 9.4.3 that the algorithm above is polynomial in the size of D.

We finish this section with some remarks on the problem (P3) in Proposition 9.2.1 for the case of planar digraphs. By Theorem 9.2.6 there is no degree of vertex-strong connectivity which guarantees that a digraph is 2-cyclic (that is, has a cycle containing x, y for every choice of vertices x, y). For planar digraphs the maximum degree of vertex-strong connectivity is 5 (Exercise 7.8). One may ask whether there is some degree of vertex-strong connectivity which suffices to guarantee that the planar digraph is 2-cyclic. However this is not the case as shown by the 5-strong non-2-cyclic planar digraph D_k ($k = 20$) in Figure 9.10 (Exercise 9.25). This example arose from a personal communication with Böhme and Harant (October 1999). The fact that there exist 5-strong non-2-cyclic planar digraphs was also mentioned by Bermond and Thomassen in the survey paper [115]. Note also that these examples of 5-strong non-2-cyclic planar digraphs show that for directed graphs there is no analogue of Tutte's theorem on hamiltonian planar graphs (every 4-connected planar graph is hamiltonian [721]).

Using the same family of planar undirected graphs G_k, $k \geq 20$, as in Figure 9.10 one can easily construct 5-strong planar graphs which do not contain disjoint $[s_1, t_1]$-, $[s_2, t_2]$-paths, hence providing the proof that the condition of being 6-connected cannot be lowered to being 5-connected for undirected graphs (recall the discussion at the end of Subsection 9.2.1).

9.5 Arc-Disjoint Branchings

This section is devoted to a very important result due to Edmonds [214]. The result can be viewed as just a fairly simple generalization of Menger's theorem. However, as will be clear from the next subsections, it has many important consequences. Recall again that an out-branching is a spanning

x $\overleftrightarrow{G}_{20}$ y

$G4$

(a) (b)

Figure 9.10 Part (a) shows a planar 5-connected graph G_k with $k = 4$; Part (b) shows a 5-strong planar digraph D_k that is obtained from the complete biorientation of G_k (shown for $k = 20$) by adding two new vertices x, y and joining these by the arcs indicated. The digraph has no cycle through x and y.

out-arborescence. In this and the next sections, unless explicitly stated otherwise, we assume that we are dealing with a directed multigraph (that is, we allow parallel arcs, but no loops).

Theorem 9.5.1 (Edmonds' branching theorem) *[214] A directed multigraph $D = (V, A)$ with a special vertex z has k arc-disjoint spanning out-branchings rooted at z if and only*[5] *if*

$$d^-(X) \geq k \qquad \textit{for all } X \subseteq V - z. \tag{9.2}$$

Proof: We give a short proof due to Lovász [521]. The necessity is clear, so we concentrate on sufficiency. The idea is to grow an out-arborescence F from z in such a way that the following condition is satisfied:

$$d^-_{D-A(F)}(U) \geq k - 1 \quad \text{for all } U \subseteq V - z. \tag{9.3}$$

If we can keep on growing F until it becomes spanning while always preserving (9.3), then the theorem follows by induction on k. To show that we can do this, it suffices to prove that we can add one more arc at a time to F until it is spanning. Let us call a set $X \subseteq V - z$ **problematic** if $d^-_{D-A(F)}(X) = k - 1$. It follows from the submodularity of $d^-_{D-A(F)}$ (recall Corollary 7.1.2) that, if X, Y are problematic and $X \cap Y \neq \emptyset$, then so are $X \cap Y, X \cup Y$. Observe also that, if X is problematic, then $X \cap V(F) \neq \emptyset$, because X has in-degree at least k in D. If all problematic sets are contained in $V(F)$, then let $T = V$. Otherwise let T be a minimal (with respect to inclusion) problematic set which is not contained in $V(F)$.

[5] By Menger's theorem (Theorem 7.3.1), (9.2) is equivalent to the existence of k arc-disjoint-paths from z to every other vertex of D.

We claim that there exists an arc uv in D such that $u \in V(F) \cap T$ and $v \in T - V(F)$. Indeed if this was not the case then every arc that enters $T - V(F)$ also enters T and we would have

$$d^-_D(T - V(F)) = d^-_{D-A(F)}(T - V(F)) \leq d^-_{D-A(F)}(T) \leq k - 1, \qquad (9.4)$$

contradicting the assumption of the theorem.

The arc uv cannot enter a problematic set T', since that would contradict the definition of T (recall that $u \in T$). Hence we can add the arc uv to F without violating (9.3) and the claim follows by induction. □

The proof above can be turned into a polynomial algorithm which given a directed multigraph $D = (V, A)$ a vertex $z \in V$ and a natural number k, either finds k arc-disjoint out-branchings from k, or a set $X \subseteq V - z$ with out-degree less than k (Exercise 9.27).

The following possible generalization naturally emerges. In addition to z, we are given a subset $T \subseteq U - z$ so that $d^-(X) \geq k$ for every subset $X \subseteq V - z, X \cap T \neq \emptyset$ (by Menger's theorem this is equivalent to saying that there are k-arc-disjoint (z,t)-paths for every $t \in T$). Is it true that there are k arc-disjoint out-arborescences rooted at z so that each contains every element of T? The answer is yes if $T = V - z$ (by Edmonds' theorem) or if $|T| = 1$ (by Menger's theorem). However, Lovász [519] found the example in Figure 9.11 which shows that such a statement is not true in general. This example can be generalized to directed multigraphs with arbitrarily many vertices (Exercise 9.30).

x

z

Figure 9.11 A digraph with $\lambda(z,t) \geq 2$, $t \in T$ which has no two arc-disjoint out-arborescences rooted at z and both containing every element of T. Here T consists of the three black vertices ([519, Figure 1]).

Observe that in Figure 9.11 $d^-(x) = 1 < 2 = d^+(x)$ holds for the only vertex x not in T and recall that the desired number of arc-disjoint arborescences above was two. Bang-Jensen, Frank and Jackson proved that, if $\lambda(z,x) \geq k$ holds for those vertices $x \in V(D)$ for which $d^+(x) > d^-(x)$ (that is, the value of k is restricted by the local arc-connectivities from z to these vertices), then a generalization is indeed possible.

Theorem 9.5.2 *[53] Let $D = (V, A)$ be a directed multigraph with a special vertex z and let $T' := \{x \in V - z : d^-(x) < d^+(x)\}$. If $\lambda(z,x) \geq k(\geq 1)$ for every $x \in T'$, then there is a family $\mathcal{F}$ of k arc-disjoint out-arborescences rooted at z so that every vertex $x \in V$ belongs to at least $r(x) := \min(k, \lambda(z,x))$ members of $\mathcal{F}$.* □

Clearly, if in Theorem 9.5.2 $\lambda(z,x) \geq k$ holds for every $x \in V$, then we are back at Edmonds' theorem. Another special case is also worth mentioning. Call a directed multigraph $D = (V, A)$ with root z a **preflow directed multigraph** if $d^-(x) \geq d^+(x)$ holds for every $x \in V - z$. (The name arises from a max-flow algorithms of Karzanov [475] and Goldberg and Tarjan [324], see also the definition of a preflow in Chapter 3). The following corollary of Theorem 9.5.2 may be considered as a generalization of Theorem 3.3.1.

Corollary 9.5.3 *[53] In a preflow directed multigraph $D = (V, A)$ for any integer $k(\geq 1)$ there is a family $\mathcal{F}$ of k arc-disjoint out-arborescences with root z so that every vertex x belongs to $\min(k, \lambda(z,x;D))$ members of $\mathcal{F}$. In particular, if $k := \max(\lambda_D(z,x) : x \in V - z)$, then every x belongs to $\lambda_D(z,x)$ members of $\mathcal{F}$.* □

Aharoni and Thomassen have shown that Edmonds' branching theorem cannot be generalized to infinite directed multigraphs [4].

9.5.1 Implications of Edmonds' Branching Theorem

Below we give a number of nice consequences of Theorem 9.5.1 (for yet another consequence see Theorem 9.7.2). The first result, due to Even, may be viewed as a generalization of Menger's theorem for global arc-strong connectivity.

Corollary 9.5.4 *[229, Theorem 6.10] Let $D = (V, A)$ be a k-arc-strong directed multigraph and let x, y be arbitrary distinct vertices of V. Then for every $0 \leq r \leq k$ there exist paths $P_1, P_2, \ldots, P_k$ in D which are arc-disjoint and such that the first r paths are (x,y)-paths and the last $k - r$ paths are (y,x)-paths.*

Proof: Let $[D, x, y]$ be as described above. Add a new vertex s and join it to x by r parallel arcs of the form sx and to y by $k - r$ parallel arcs of the form sy. Let D' denote the new directed multigraph. We claim that D' satisfies (9.2). To see this let $X \subseteq V$ be arbitrary. If $X \neq V$, then we have $d^-_{D'}(X) \geq d^-_D(X) \geq k$, since D is k-arc-strong. If $X = V$, we have

$d^-_{D'}(V) = d^+_{D'}(s) = k$. It follows from Theorem 9.5.1 that D' contains k arc-disjoint out-branchings all rooted at s. By the construction of D', these branchings restricted to D must consist of r out-branchings rooted at x and $k-r$ out-branchings rooted at y. Take the r (x,y)-paths from those rooted at x and the $k-r$ (y,x)-paths from those rooted at y and we obtain the desired paths. □

The next result, due to Nash-Williams, gives a sufficient condition for the existence of k-edge-disjoint spanning trees in an undirected graph. This condition is the best possible in terms of the edge-connectivity (see the remark after Theorem 9.5.6) and hence we see that for undirected graph we may need twice the obvious edge-connectivity requirement to guarantee k edge-disjoint trees. This contrasts with the case for directed graphs where k-arc-strong connectivity suffices by Edmonds' theorem.

Theorem 9.5.5 *[584] Every $2k$-edge-connected undirected graph contains k edge-disjoint spanning trees.*

Proof: Let $G=(V,E)$ be a $2k$-edge-connected undirected graph. By Nash-Williams' orientation theorem (Theorem 8.6.3), G has a k-arc-strong orientation $D=(V,A)$. Let $z \in V$ be arbitrary and note that $d^-(X) \geq k$ holds for each subset $X \subseteq V-z$ of vertices. Hence by Theorem 9.5.1, D contains k-arc disjoint out-branchings rooted at z. Suppressing the orientation of all arcs on the branchings we obtain k edge-disjoint trees in $G = UG(D)$. □

The following characterization, due to Tutte, of undirected graphs which have k edge-disjoint spanning trees can also be derived from Edmonds' branching theorem and Theorem 8.7.6 (see Exercise 9.35). See also Exercise 8.57 for a simpler orientation result which still implies Theorem 9.5.6.

Theorem 9.5.6 *[722] An undirected graph $G=(V,E)$ has k edge-disjoint spanning trees if and only if*

$$\sum_{1\leq i<j\leq p} e(V_i,V_j) \geq k(p-1), \tag{9.5}$$

holds for every partition $V_1, V_2, \ldots, V_p$ of V. Here $e(V_i,V_j)$ denotes the number of edges with one end in V_i and the other in V_j. □

It is easy to derive Theorem 9.5.5 from Theorem 9.5.6. Furthermore, we can use Theorem 9.5.6 to show that the condition in Theorem 9.5.5 is best possible in terms of the edge-connectivity. Let G_k be the graph obtained from the complete graph on $2k+2$ vertices by removing the edges of a hamiltonian cycle. Then it is easy to show that G_k is $(2k-1)$-edge-connected and using Theorem 9.5.6 on the partition corresponding to one vertex per set in the partition we can see that G_k has no k edge-disjoint spanning trees (in fact this partition has precisely one arc less than the required number). In order to get an example with arbitrarily many vertices and no k edge-disjoint trees for each k we let H be an arbitrary $2k$-edge-connected graph and let H_k be the graph consisting of $2k+2$ copies $H_1, H_2, \ldots, H_{2k+2}$ of H and with one edge between H_i and H_j just if the corresponding vertices v_i, v_j are adjacent in G_k (where we have assumed that the vertices of G_k are labelled $v_1, v_2, \ldots, v_{2k+2}$ and H_i corresponds to v_i for $i = 1, 2, \ldots, 2k+2$). It is not difficult to prove that H_k is $(2k-1)$-edge-connected and the partition corresponding to the $2k+2$ copies of H shows that H_k has no k edge-disjoint spanning trees. Note also that G_k above is $(2k-1)$-edge-connected and $(2k-1)$-regular. Furthermore , a simple counting argument shows that all except finitely many $(2k-1)$-edge-connected and $(2k-1)$-regular graphs have no k edge-disjoint spanning trees (simply because they do not have enough edges).

In some applications (e.g. when a number of tasks have to be distributed to different units who can cover part of the jobs or demands) one is interested in covering all edges (arcs) of an undirected (a directed) graph by forests (arborescences).

Theorem 9.5.7 *[585] Let $G = (V, E)$ be an undirected graph. Then E can be covered by k forests if and only if*

$$|E(G\langle X\rangle)| \leq k(|X| - 1) \qquad \textit{for all } X \subseteq V. \tag{9.6}$$

Proof: Since no forest can use more than $|X|-1$ edges with both ends inside any set X, we see that the condition (9.6) is necessary. To prove sufficiency we use Theorem 9.5.1 and the following result which follows easily from Theorem 8.7.3:

Proposition 9.5.8 *A graph $H = (V, E)$ has an orientation $D = (V, A)$ such that $d^-_D(v) \leq k$ for every vertex $v \in V$ if and only if*

$$|E(G\langle X\rangle)| \leq k|X| \qquad \textit{for all } X \subseteq V.$$

□

Suppose now that $G = (V, E)$ satisfies (9.6). By Proposition 9.5.8, G has an orientation D such that $d^-_D(v) \leq k$ for every vertex $v \in V$. Add a new vertex s to D and add $k - d^-_D(v)$ arcs from s to v for each $v \in V$. Denote the new directed multigraph by D'. We claim that

$$d^-_{D'}(X) \geq k \qquad \text{for all } X \subseteq V. \tag{9.7}$$

This follows from the fact that for every $X \subseteq V$ we have

$$\begin{aligned} d^-_{D'}(X) &= \sum_{v \in X} d^-_D(v) - |E(G\langle X\rangle)| \\ &= k|X| - |E(G\langle X\rangle)| \\ &\geq k|X| - k(|X| - 1) = k. \end{aligned}$$

By Theorem 9.5.1, D' has k arc-disjoint out-branchings rooted at s. These branchings must use all arcs of D since every vertex of V has in-degree one in each of these branchings and we have only added $k - d^-_D(v)$ arcs from s to v. Now delete the vertex s from each of the branchings and suppress the orientations of all arcs. The resulting k forests cover E. □

The last part of the proof above also implies the sufficiency part of the following theorem. The necessity of (9.8) follows from the fact that no vertex of an out-branching has in-degree bigger than one. The necessity of (9.9) is seen as in the proof above.

Theorem 9.5.9 *[252] The arc set of a directed graph $D = (V, A)$ can be covered by k out-arborescences if and only if*

$$d^-(v) \leq k \qquad \text{for all } v \in V \text{ and} \tag{9.8}$$

$$|A(D\langle X\rangle)| \leq k(|X| - 1) \qquad \text{for all } X \subseteq V. \tag{9.9}$$

□

9.6 Edge-Disjoint Mixed Branchings

We saw in the proof of Theorem 9.5.5 that we could use Edmonds' branching theorem to prove that every $2k$-edge-connected graph has k-edge-disjoint spanning trees. However, that proof does not imply an algorithm to check whether a given undirected graph has k edge-disjoint spanning trees. In fact this problem is more complicated for undirected graphs than the problem of finding k arc-disjoint out-branchings from a given root in a directed multigraph where the proof of Edmonds' branching theorem provides the answer. For undirected graphs the characterization, given in Theorem 9.5.6,is much more complicated and does not imply a polynomial algorithm for the problem. Note that such an algorithm can be obtained from a formulation of the problem as a matroid partition problem (see Exercise 12.46). See also the remark at the end of the section.

A **mixed multigraph** is the same as a mixed graph, except that we allow parallel arcs and parallel edges as well as arcs that are parallel to edges. Consider the following common generalization of a spanning tree rooted at s in an undirected multigraph and an out-branching with root s in a directed multigraph. A **mixed out-branching** rooted at s is a mixed graph F whose underlying graph is a tree such that $\overleftrightarrow{F}$ contains an out-branching rooted at s. We say that two subgraphs of a mixed multigraph are **edge-disjoint** if they do not share any arcs or edges (they may contain different copies of an arc/edge, but not the same).

Definition 9.6.1 *Let $M = (V, E \cup A)$ be a mixed multigraph with a special vertex s. A mixed out-branching F_s^+ with root s is a spanning tree in the underlying undirected multigraph G of M with the property that there is a path from s to every other vertex v in F_s^+.*

One reason why mixed out-branchings are of interest in relation to undirected graphs can be seen from the following easy lemma (which in particular covers the case when no arc of M is directed).

Lemma 9.6.2 *Let $M = (V, E \cup A)$ be a mixed multigraph with a special vertex s called root. There are k edge-disjoint mixed out-branchings rooted at s if and only if there exist an orientation D of M with k edge-disjoint out-branchings at s.*

Proof: Exercise 9.31. □

The following characterization, due to Frank, generalizes Theorem 9.5.6 and Theorem 9.5.1. This theorem can be derived from the feasibility theorem for intersecting submodular flows (Exercise 9.33).

Theorem 9.6.3 *[252] Let $M = (V, E \cup A)$ be a mixed multigraph with a special vertex s. There are k edge-disjoint mixed out-branchings rooted at s, if and only if the following holds for all subpartitions $\mathcal{F} = \{V_1, V_2, \ldots, V_t\}$ of $V - s$:*

$$a_{\mathcal{F}} \geq kt, \tag{9.10}$$

where $a_{\mathcal{F}}$ denotes the number of edges, oriented or not, which enter some V_i. □

We point out that one can use submodular flows to decide in polynomial time whether a given undirected graph G has k edge-disjoint spanning trees. By Lemma 9.6.2 all we need to check is whether there is some orientation of G which has k arc-disjoint out-branchings from a given vertex. Thus, given G we form an arbitrary orientation D of G and then follow the approach in Exercise 8.65. It is not hard to see that, with a slight modification, the same approach can be used to determine the existence of k edge-disjoint mixed branchings from a given root in a mixed graph (Exercise 9.32).

9.7 Arc-Disjoint Path Problems

Recall that a directed multigraph $D = (V, A)$ is weakly k-linked if for every choice of (not necessarily distinct) vertices $s_1, \ldots, s_k, t_1, \ldots, t_k$, D contains arc-disjoint paths $P_1, \ldots, P_k$ such that P_i is an (s_i, t_i)-path for $i = 1, \ldots, k$. The **arc-disjoint k-path problem** is the following. Given a directed multigraph $D = (V, A)$ and distinct vertices $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$, decide whether D contains k arc-disjoint paths $P_1, \ldots, P_k$ such that P_i is an (x_i, y_i)-path. In view of Theorem 9.7.2 below, the following result by Fortune, Hopcroft and Wyllie may seem slightly surprising.

Theorem 9.7.1 *[247] The arc-disjoint k-path problem is $\mathcal{NP}$-complete already for $k = 2$.*

Proof: Let $[D, x, y, u, v]$ be an instance of the 2-path problem. Transform $D = (V, A)$ into the directed multigraph H by performing the vertex-splitting procedure (see Section 3.2). Then it is easy to show that H has a pair of arc-disjoint (x_t, y_s)-, (u_t, v_s)-paths if and only if D has disjoint (x, y)-, (u, v)-paths (Exercise 9.36). Since H can be constructed from D in polynomial time, the claim now follows from Theorem 9.2.3. □

For planar directed multigraphs it is an open problem whether there exists a polynomial algorithm to decide the existence of two arc-disjoint paths with prescribed end vertices (see e.g. Schrijver's papers [656, page 265] and [658]). Even the complexity of the special case when we are looking for arc-disjoint (x, y)- and (y, x)-paths is open! Hence we see from Theorem 9.4.2 that the arc-disjoint 2-path problem is much more difficult for planar digraphs than the 2-path problem. This is not really surprising since planarity certainly has implications on vertex disjoint paths, whereas the implications on arc-disjoint paths are not so obvious although there clearly are some.

Observe that, if D is weakly k-linked, then D is k-arc-strong. To see this it suffices to take $s_i = x$ and $t_i = y$ for each i, then there are k arc-disjoint (x, y)-paths in D and since x, y may be chosen arbitrarily, it follows that D is k-arc-strong.

Shiloach observed [669] that Edmonds' branching theorem implies that k-arc-strong connectivity is also sufficient for the existence of k arc-disjoint paths with specified initial and terminal vertices:

Theorem 9.7.2 *A directed multigraph D is weakly k-linked if and only if D is k-arc-strong.*

Proof: Above we have argued on the necessity. To see the sufficiency, let $x_1, x_2, \ldots, x_k$, $y_1, \ldots, y_k$ be given. Construct a new directed multigraph D' by adding a new vertex s and arcs sx_i, $i = 1, 2, \ldots, k$ to D. Since D is k-arc-strong, it is not difficult to check that $d^-_{D'}(X) \geq k$ for every subset X of V. Hence by Edmonds' branching theorem, D' has arc-disjoint out-branchings $F^+_{s,1}, \ldots, F^+_{s,k}$ all rooted at s. Since s has out-degree k in D', each $F^+_{s,i}$ must use precisely one arc out of s and without loss of generality $F^+_{s,i}$ uses the arc sx_i. Now it is clear that $F^+_{s,i}$ contains an (x_i, y_i)-path P_i and the paths $P_1, \ldots, P_k$ form the desired linking. □

Using Theorem 9.5.2 we can obtain, in an analogous way, the following sufficient condition, due to Bang-Jensen, Frank and Jackson, for the existence of k arc-disjoint paths with prescribed initial and terminal vertices (Exercise 9.37).

Theorem 9.7.3 *[53] Let $(s_1, t_1), \ldots, (s_k, t_k)$ be k pairs of vertices in a directed multigraph $D = (V, A)$ so that for every vertex x with $d^-(x) < d^+(x)$ or $x = t_j$ there are arc-disjoint paths from s_i to x $(i = 1, \ldots, k)$. Then there are arc-disjoint paths from s_i to t_i $(i = 1, \ldots, k)$.* □

Note that, if we only impose the condition in Theorem 9.7.3 on the vertices t_i, $i = 1, 2, \ldots, k$, then D may not have arc-disjoint paths from s_i to t_i $(i = 1, 2, \ldots, k)$. This can be seen from the example in Figure 9.12. The example can easily be generalized to arbitrary local strong connectivities from s_i to t_i, $i = 1, 2$ while preserving planarity. We formulate this as a theorem below.

Theorem 9.7.4 *For every natural number k there exists a planar digraph D with distinct vertices s_1, s_2, t_1, t_2 such that D has $\kappa_D(s_i, t_i) \geq k$ for $i = 1, 2$, but D has no arc-disjoint (s_1, t_1)-, (s_2, t_2)-paths.* □

This shows that there is no sufficient condition for the existence of arc-disjoint paths connecting the vertices of a prescribed set in terms of local vertex-strong connectivities from s_i to t_i, $i = 1, 2 \ldots, r$.

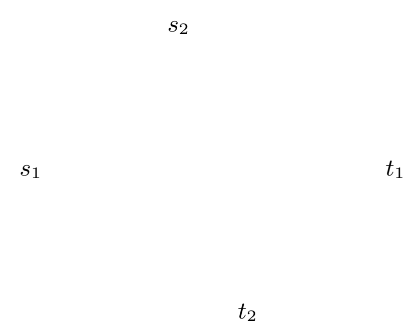


Figure 9.12 An example of a planar digraph with $\kappa(s_i, t_i) = 2$, $i = 1, 2$ and no arc-disjoint (s_1, t_1)-, (s_2, t_2)-paths.

As yet another example of the usefulness of Edmonds branching theorem, we consider the following problem called the **arc-disjoint (t_1, t_2)-linking problem**: given a directed multigraph D and two specified vertices t_1 and t_2. Do there exist arc-disjoint (s_1, t_1)-, (s_2, t_2)-paths for every choice of vertices s_1, s_2 in D, except possibly in the case when $s_i = t_j \neq t_i$ and there are no arcs out of s_i? The **(t_1, t_2)-cut condition** is satisfied by D if

$$|(S, \bar{S})| \geq |\{i = 1, 2 : t_i \notin S\}| \tag{9.11}$$

for each (t_1, t_2)-cut $(S, \bar{S})$. The cut condition is obviously a necessary condition for the directed multigraph to have the arc-disjoint linking property. Below we give a very simple proof due to Frank (private communication, April 1994) that it is also sufficient.

Theorem 9.7.5 *D has the arc-disjoint linking property with respect to the set $\{t_1, t_2\}$ if and only if D satisfies the (t_1, t_2)-cut-condition.*

Proof: By the remark above, it suffices to consider the case when D satisfies the (t_1, t_2)-cut-condition. Add an extra vertex t and the following new arcs: $t_1t, t_2t, t_1s_2, t_2s_1$. Now it follows from Theorem 9.5.1 and the fact that D satisfies the (t_1, t_2)-cut condition that in the extended graph there are 2 arc-disjoint in-branchings $F^-_{t,1}, F^-_{t,2}$ rooted at t. These contain the two required paths in the original graph since the new arcs t_1s_2 and t_2s_1 cannot play a role. □

9.7.1 Arc-Disjoint Paths in Acyclic Directed Multigraphs

The following easy observation, due to Fortune, Hopcroft and Wyllie, can be used to reduce the arc-disjoint k-path problem to the k-path problem in the case of acyclic directed multigraphs. We need the following lemma whose proof is left as Exercise 9.38.

Lemma 9.7.6 *If D is acyclic, then so is its line digraph $L(D)$.* □

Theorem 9.7.7 *[247] For each k, there exists a polynomial algorithm for the arc-disjoint k-path problem for the class of acyclic directed multigraphs.*

Proof: Let $[D, x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k]$ be an instance of the arc-disjoint k-path problem where D is an acyclic directed multigraph. If some x_i has out-degree zero or some y_j has in-degree zero, then trivially the desired paths do not exist. Hence we may assume that this is not the case.

Transform the instance $[D, x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k]$ into a new instance $[D', x'_1, x'_2, \ldots, x'_k, y'_1, y'_2, \ldots, y'_k]$ as follows. If x_i has out-degree two or more we add a new vertex x'_i and the arc x'_ix_i to D; otherwise let $x'_i := x_i$, $i = 1, 2, \ldots, k$. Similarly, if y_j has in-degree more than one, we add a new vertex y'_j and the arc $y_jy'_j$; otherwise let $y'_j := y_j$, $j = 1, 2, \ldots, k$. Clearly, D' has arc-disjoint paths $P'_1, \ldots, P'_k$ such that P'_i is an (x'_i, y'_i)-path, $i = 1, 2, \ldots, k$, if and only if D has arc-disjoint paths $P_1, \ldots, P_k$, where P_i is an (x_i, y_i)-path, $i = 1, 2, \ldots, k$.

Now consider $D^* := L(D')$ and let s_i (t_i) be the vertex of D^* which corresponds the unique arc with tail (head) x'_i (y'_i). Then it is easy to show that D^* has a collection $Q_1, Q_2, \ldots, Q_k$ of disjoint paths so that Q_i is an (s_i, t_i)-path, $i = 1, 2, \ldots, k$ if and only if D' has arc-disjoint paths $P'_1, \ldots, P'_k$ such that P'_i is an (x'_i, y'_i)-path, $i = 1, 2, \ldots, k$.

Since there is a polynomial algorithm for transforming the instance $[D, x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k]$ into $[D^*, s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k]$, the theorem now follows from Theorem 9.2.14. □

In [656], Schrijver shows how to apply a polynomial algorithm for the arc-disjoint k-path problem in acyclic digraphs to solve a scheduling problem in the airline industry.

9.7.2 Arc-Disjoint Paths in Eulerian Directed Multigraphs

As we will see below, questions about arc-disjoint paths are slightly easier for eulerian directed multigraphs than for arbitrary directed multigraphs. However, the arc-disjoint 2-path problem seems difficult and is still open. As we mentioned in Chapter 7, eulerian directed multigraphs often have properties similar to those of undirected multigraphs. This is also illustrated by their properties with respect to arc-disjoint paths as can be seen from some of the results mentioned in this subsection (see, e.g., Figure 9.14).

We start with a very simple, yet quite important observation. As mentioned earlier the complexity version of the corresponding problem for planar digraphs is still open.

Lemma 9.7.8 *Let D be a eulerian directed multigraph and let s, t be distinct vertices of D. Then D has arc-disjoint (s,t)-, (t,s)-paths if and only if D has an (s,t)-path.*

Proof: Let P be an arbitrary (s,t)-path. Let D' be obtained from D by removing the arcs of P. In D', every vertex distinct from s, t has in-degree equal out-degree and we have $d^-_{D'}(s) = d^+_{D'}(s) + 1$, $d^+_{D'}(t) = d^-_{D'}(t) + 1$. Let $\mathcal{N}(D')$ be the network representation of D' (recall Definition 7.1.4) and let x be the flow that has value equal to the capacity on every arc. By the flow decomposition theorem (Theorem 3.3.1), x can be decomposed into a (t,s)-flow of value one and some cycle flows. Since the (t,s)-path in $\mathcal{N}(D')$ is also a path in D', D' contains a (t,s)-path as claimed. □

Let $x_1, \ldots, x_k$ be a k-tuple of (not necessarily distinct) vertices, which will be called **terminals**. We say that a trail $T = (v_0 v_1 v_2 \ldots v_{t-1} v_t)$ **visits the terminals in the order** $x_1, \ldots, x_k$ if $x_1 = v_{i_1}, x_2 = v_{i_2}, \ldots, x_k = v_{i_k}$ for some $0 \leq i_1 \leq \ldots \leq i_k \leq t$. (We do not exclude some additional occurrences of terminals in a trail. In general, a trail may visit given terminals in several different orders.) Based on the following lemma (whose proof is left as Exercise 9.42), we could restrict ourselves only to eulerian trails. However, it is sometimes convenient to work also with non-eulerian trails.

Lemma 9.7.9 *Let D be an eulerian directed multigraph. Assume that there is a trail visiting some terminals in the order $x_1, \ldots, x_k$. Then there exists an eulerian trail visiting the terminals in the same order.* □

Given an eulerian directed multigraph and terminals $x_1, x_2, \ldots, x_k$ there are at least three different problems one may consider [440]:

Specific Trail Problem (ST-problem).

Instance: An eulerian directed multigraph G and an ordered k-tuple of terminals $x_1, \ldots, x_k$.

Question: Does there exist a trail visiting the terminals in the order $x_1, \dots, x_k$?

Unique Trail Problem (UT-problem).

Instance: An eulerian directed multigraph G and an unordered k-tuple of terminals $x_1, \dots, x_k$.
Question: Do all eulerian trails visit the terminals in the same cyclical order?

All Trail Problem (AT-problem).

Instance: An eulerian directed multigraph G and an unordered k-tuple of terminals $x_1, \dots, x_k$.
Question: Does there exist, for every permutation π of $\{1, \dots, k\}$, a trail visiting the terminals in the order $x_{\pi(1)}, \dots, x_{\pi(k)}$?

We will denote by k-ST, k-UT and k-AT the corresponding problems when the number of terminals is exactly k. The ST-problem seems to be the most important among these three problems, since it is equivalent to the eulerian arc-disjoint linking problem (see Lemma 9.7.10). However, the remaining two problems occur naturally in the study of the ST-problem.

As we show below, results on these three problems for eulerian directed multigraphs are, in fact, strongly related to arc-disjoint linkings in directed multigraphs which are not eulerian, but become eulerian if we add the so-called demand arcs. Let $[D, s_1, s_2, \dots, s_k, t_1, t_2, \dots, t_k]$ be an instance of the arc-disjoint k-path problem. The **demand directed multigraph** H associated with this instance is the directed multigraph consisting of the arcs[6] $t_1s_1, t_2s_2, \dots, t_ks_k$. The special case of the arc-disjoint k-path problem when $D+H$ is eulerian, (here H is the demand directed multigraph of D) is called the **eulerian arc-disjoint k-linking problem**. When, instead of being a fixed number k, the number of demand arcs is part of the input, we call the above problem the **eulerian arc-disjoint linking problem**.

Lemma 9.7.10 *The k-ST-problem is equivalent to the eulerian arc-disjoint k-linking problem.*

Proof: We show that the k-ST problem is a special case of the eulerian arc-disjoint k-linking problem using the following reduction. Let $[D, x_1, \dots, x_k]$ be an instance of the k-ST-problem. Define $s_1, t_1, \dots, s_k, t_k$ by $s_i = x_i$ and $t_i = x_{i+1}, i = 1, \dots, k$, $(x_{k+1} = x_1)$ and let H consist of the arcs t_is_i, $i = 1, \dots, k$. Then $D + H$ is eulerian and it is easy to see that $D + H$ has arc-disjoint paths $P_1, \dots, P_k$, where P_i is and (s_i, t_i)-path, $i = 1, 2, \dots, k$, if and only if D has a trail visiting the terminals in the order $x_1, x_2, \dots, x_k$.

Conversely, given an instance $[D, s_1 \dots, s_k, t_1, \dots, t_k]$ of the eulerian arc-disjoint k-linking problem (thus $D+H$ is eulerian), we construct an instance

[6] Hence, if $s_1 = s_2 = \dots = s_k$ and $t_1 = t_2 = \dots = t_k$, the demand directed multigraph consists of k parallel arcs from t_1 to s_1.

of the k-ST-problem as follows. Let $\tilde{D}$ be the directed multigraph obtained from D by adding new vertices $x_1, \ldots, x_k$, and arcs $x_i s_i, t_i x_{i+1}, i = 1, \ldots, k$. Clearly, $\tilde{D}$ is an eulerian directed multigraph, and it admits a closed trail visiting the terminals in the order $x_1, \ldots, x_k$ if and only if D admits an arc-disjoint k-linking for the prescribed pairs (s_i, t_i) $i = 1, \ldots, k$, of terminals. □

Now we see from Lemma 9.7.8 that the arc-disjoint 2-path problem is easy in the case when the directed multigraph in question becomes eulerian if we add the two demand arcs $t_1 s_1, t_2 s_2$. This was also observed by Frank in [257]. The eulerian arc-disjoint 3-linking problem is already considerably harder. It was solved by Ibaraki and Poljak [440]. We describe their main result in Theorem 9.7.11.

It is easy to see that for $k = 3$, the problems 3-ST, 3-UT, and 3-AT are mutually equivalent from a complexity point of view. The reason is that for $k = 3$ there are only two distinct cyclical orders of terminals, (x_1, x_2, x_3) and (x_1, x_3, x_2). Moreover, we may assume that one eulerian trail T of G is already given (since it may be constructed by a polynomial time algorithm according to Exercise 1.72). The trail T visits the terminals in one of the possible orders, say (x_1, x_2, x_3). Hence it only remains to decide whether there is a trail visiting the terminals in the other order.

We recall the solution, due to Ibaraki and Poljak [440], of the UT-problem, since it suggests a possible approach to the remaining two problems. Recall that, for an arc a of D, D/a denotes the directed multigraph obtained from D by (set-)contracting the arc a. We allow terminals to be identified by the contraction. Below we denote the set of terminals by X and an instance of the UT-problem by $[D, X]$. Clearly, if $[D, X]$ admits several orders of visiting terminals, then $[D/a, X]$ admits several orders as well, but the converse need not be true. We say that $[D, X]$ is **UT-minimal**, if $[D, X]$ admits unique cyclical order of visiting terminals by an eulerian trail, but $[D/a, X]$ admits several orders whenever any arc a is contracted. Ibaraki and Poljak characterized UT-minimal instances.

Theorem 9.7.11 *[440] Let $[D, X]$ be a UT-minimal instance. Then*

(a) $d^+(x) = d^-(x) = 1$ for every terminal x, and $d^+(u) = d^-(u) = 2$ for every non-terminal u,

(b) D can be embedded in the plane such that every face is a directed cycle, and all terminals lie on one common face. □

Observe that first part of the condition (b) is equivalent with the property that the four edges incident to a non-terminal vertex u are oriented alternatively out of and in to the vertex u (in the planar representation). See Figure 9.13.

Theorem 9.7.12 *[440] Both the UT-problem and the 3-ST-problem are polynomially solvable.* □

x_3

x_1 x_2

Figure 9.13 An eulerian digraph with no (eulerian) trail visiting x_1, x_2, x_3 in that order.

Furthermore, Ibaraki and Poljak proved that the eulerian arc-disjoint linking problem and hence the ST problem are $\mathcal{NP}$-complete.

Theorem 9.7.13 *[440] The eulerian arc-disjoint linking problem is $\mathcal{NP}$-complete.*

Proof: We sketch the construction used in [440]. The reduction is from the arc-disjoint 2-path problem, which is $\mathcal{NP}$-complete by Theorem 9.7.1. Let $[D = (V, A), s_1, s_2, t_1, t_2]$ be an instance of the arc-disjoint 2-path problem. Let $D^* = D + H$ be the directed multigraph we obtain from D by adding the two demand arcs t_1s_1 and t_2s_2.

Form a directed multigraph D' from D by adding two new vertices s, t and, for every $v \in V$, appending $\max\{0, d^+_{D^*}(v) - d^-_{D^*}(v)\}$ arcs of the form sv as well as $\max\{0, d^-_{D^*}(v) - d^+_{D^*}(v)\}$ arcs of the form vt. Let p be the sum of $d^+_{D^*}(v) - d^-_{D^*}(v)$ taken over those vertices for which this number is positive. Now let $s_i = s$ and $t_i = t$, $i = 3, 4, \ldots, p+2$, be new terminals. Then $[D', s_1, s_2, \ldots, s_{p+2}, t_1, t_2, \ldots, t_{p+2}]$ is an instance of the eulerian arc-disjoint linking problem and it is not difficult to show that D has arc-disjoint (s_1, t_1)-, (s_2, t_2)-paths if and only if D' has arc-disjoint (s_i, t_i)-paths, $i = 1, 2, \ldots, p+2$, (Exercise 9.43). $\square$

Ibaraki and Poljak posed the following conjecture:

Conjecture 9.7.14 *[440] The k-ST-problem is polynomial for any fixed k.*

The condition of minimality which was used in Theorem 9.7.11 can be replaced by a more technical notion of irreducibility. Let us say that an instance $[D, X]$ is **2-irreducible** if there is no set $S, |S| > 1$, of vertices such that one of the following holds:

(a) $|(S, \bar{S})| = |(\bar{S}, S)| \leq 2$, $D\langle S\rangle$ is connected and $S \cap X = \emptyset$,
(b) $|(S, \bar{S})| = |(\bar{S}, S)| = 1$, and $|S \cap X| = 1$.

Note that D/S (the directed multigraph obtained by contracting S) is eulerian whenever D is eulerian. It is not difficult to see the following:

Lemma 9.7.15 *Let $[D, X]$ be an instance of the UT-problem which admits a unique order, and let S satisfy one of the conditions (a) and (b). Then $[D/S, X]$ admits a unique order as well.* □

It is also easy to see that D/S can be realized by a series of arc contractions, and hence every minimal UT-instance is 2-irreducible. Thus, the following theorem is a generalization of Theorem 9.7.11.

Theorem 9.7.16 *[440] Let $[D, X]$ be a UT-instance which is 2-irreducible and admits eulerian trail with unique order of terminals. Then the conditions (a) and (b) of Theorem 9.7.11 hold.*

The polynomial time algorithm for the UT-problem is a consequence of Theorem 9.7.16. The algorithm proposed in [440] consists of the following steps:

1. Reduce an instance $[D, X]$ to a 2-irreducible one. This can be done by applying network flow techniques.
2. Check the degree conditions.
3. Using a planarity test, decide whether D has a planar drawing, and if yes, then test the remaining conditions of Theorem 9.7.16.

The notion of 2-irreducibility formulated here is weaker than the notion of irreducibility used in [440] where it was required, in addition, that $[D, X]$ does not contain any non-terminal vertex of in- and out-degree one. However, using the general definition of irreducibility given in [88, Section 3], it can be seen that this additional condition is automatically satisfied by any AT-infeasible and irreducible instance.

Let $[D, X]$ be an instance of AT-problem. Let us say that $[D, X]$ is **AT-minimal**, if $[D, X]$ does not admit an eulerian trail visiting the terminals for every given order, but $[D/a, X]$ does whenever any arc a is contracted. The following result by Bang-Jensen and Poljak shows that there are also degree restrictions on AT-minimal instances.

Theorem 9.7.17 *[88] Let $[D, X]$ be k-AT-minimal. Then $d^+(u) \leq k-1$ for every non-terminal u, and $d^+(x) \leq k-2$ for every terminal x.* □

The edge-disjoint 2-path problem for undirected graphs is polynomially solvable and a complete characterization of undirected graphs having no edge-disjoint s_1t_1 and s_2t_2-paths is available (Dinic and Karzanov [196, 197], Seymour [662] and Thomassen [697]). Such a graph G can be reduced to a graph G' that has a planar representation with the following properties (see Figure 9.14(a)):

(a) Each of the four terminals has degree 2 and all other vertices have degree 3, and
(b) the terminals are located on the outer face in the order s_1, s_2, t_1, t_2.

x u s p

s t p q

v y q t

(a) (b) (c)

Figure 9.14 Part (a) shows an infeasible instance for the edge-disjoint 2-path problem for undirected graphs. The graph shown has no edge-disjoint xy-path and uv-path; Parts (b) and (c) show infeasible instances of the arc-disjoint $[s,t]$-, $[p,q]$-paths problem for eulerian directed multigraphs.

The complete biorientation $\overleftrightarrow{G}$ of an undirected graph G is eulerian and it contains arc-disjoint (s_1,t_1)-, (s_2,t_2)-paths if and only if G contains edge-disjoint s_1t_1, s_2t_2-paths. Hence, the arc-disjoint 2-path problem for eulerian digraphs generalizes the edge-disjoint 2-paths problem. So far the arc-disjoint 2-path problem for eulerian digraphs remains unsolved. However, even the simpler version in which we just require arc-disjoint $[s_1,t_1]$, $[s_2,t_2]$-paths (that is, the order of s_i, t_i is not fixed in the ith path, $i = 1, 2$) still generalizes the edge-disjoint 2-path problem. This problem was recently solved by Frank, Ibaraki and Nagamochi in [270]. They proved that the problem is solvable in polynomial time. Furthermore they showed the following result. By a **reduction** below we mean a series of transformations such that the desired paths exist in the new digraph if and only if they exist in the previous digraph (for details see [270]).

Theorem 9.7.18 *[270] Let D be an eulerian directed multigraph and let s_1, s_2, t_1, t_2 be not necessarily distinct vertices of D. Then D contains arc-disjoint $[s_1,t_1]$, $[s_2,t_2]$-paths, unless it can be reduced to an eulerian directed multigraph D' such that either D' has 6 vertices and is isomorphic to the digraph in Figure 9.14(c), or each of (a),(b) and (c) below hold.*

(a) Each of s_1, s_2, t_1, t_2 have in- and out-degree one and all other vertices have in- and out-degree two in D'.
(b) There is at most one cut vertex[7] in $UG(D')$.
(c) D has a planar embedding such that every face is a directed cycle and all terminals are located on the outer face in the order s, p, t, q where $\{s,t\} = \{s_1,t_1\}$ and $\{p,q\} = \{s_2,t_2\}$. □

[7] Recall that a vertex x in a connected undirected graph G is a cut vertex if $G - x$ is not connected.

We finish this section with a remark on very recent work by Seymour and Johnson (Seymour and Johnson, personal communication on work in progress, February 2000) which may have far reaching consequences. There seems to be a theory for eulerian directed multigraphs which is similar to the graph minors theory for undirected graphs by Seymour and Robertson (see e.g. [642]). Instead of 'minor' the natural containment relation when studying eulerian directed multigraphs is immersion which we define below for 2-regular directed multigraphs.

A 2-regular directed multigraph H is **immersed** in another 2-regular directed multigraph D if we can obtain H from D by repeatedly choosing a vertex v with in-neighbours u_1, u_2 and out-neighbours w_1, w_2, deleting v and adding two new arcs u_1w_1, u_2w_2. See Figure 9.15.

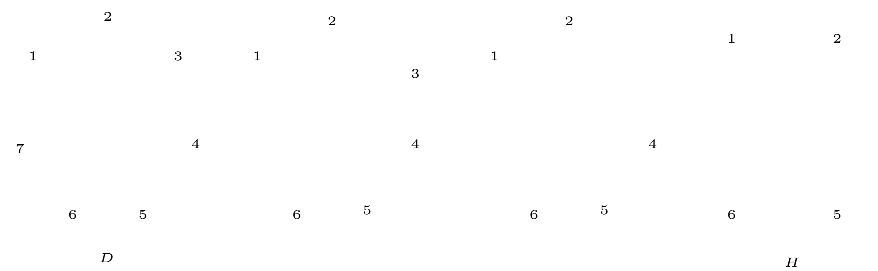


Figure 9.15 Immersing the directed multigraph H in the directed multigraph D by suppressing the vertices $7, 3$ and 4 in that order.

It appears that an analogue of the structure theorem of graph minors[8] holds for 2-regular directed multigraphs. The potential applications of such a theorem include the well-quasi ordering of 2-regular directed multigraphs under immersion and a polynomial time algorithm for the arc-disjoint k-path problem. The polynomial solvability of the arc-disjoint k-path problem even appears to hold for general eulerian directed multigraphs (Seymour, private communication, February 2000).

9.7.3 Arc-Disjoint Paths in Tournaments and Generalizations of Tournaments

We now consider the arc-disjoint 2-path problem for some generalizations of tournaments. We prove that this problem and a related special case (the arc version of problem (P5) from Proposition 9.2.1) are polynomially solvable for semicomplete digraphs. As we will see in Section 9.9, the corresponding algorithms are used as subroutines in a much more complicated algorithm for a problem concerning arc-disjoint in- and out-branchings in tournaments.

[8] For a nice introduction to this deep result, see Diestel's book [191].

We prove the first results for the class of extended locally in-semicomplete digraphs instead of just for semicomplete digraphs. We do this to show that not much extra effort is needed to obtain the result (which also has the same statement as for semicomplete digraphs alone) for this much larger class of digraphs. The results in this subsection are due to Bang-Jensen [46, 51]

Recall that two vertices are similar if and only if they are non-adjacent and have the same in- and out-neighbours. Note that, if x, y are non-adjacent vertices with a common out-neighbour w in an extended locally in-semicomplete, digraph, then x and y are similar vertices, by the definition of an extension and the definition of a locally in-semicomplete digraph.

The following lemma can be proved along the same lines as Lemma 9.7.20. The proof is left to the reader as Exercise 9.39.

Lemma 9.7.19 *Let D be a strong extended locally in-semicomplete digraph and let x, y be distinct vertices of D. Then D has arc-disjoint (x, y)-,(y, x)-paths if and only if there is no arc a such that $D - a$ contains no (x, y)-path and no (y, x)-path.* □

Lemma 9.7.20 *[51] Let D be an extended locally in-semicomplete digraph and x, y, z vertices of D such that $x \neq z$ and D contains a path from y to z. If D has arc-disjoint (x, y)-, (x, z)-paths, then D contains arc-disjoint (x, y)-, (y, z)-paths. Similarly, if an extended locally out-semicomplete digraph D' has a path from x to y and arc-disjoint (x, z)-, (y, z)-paths, then D' has arc-disjoint (x, y)-, and (y, z)-paths.*

Proof: Let P_1 and P_2 be arc-disjoint paths such that P_2 is an (x, z)-path and P_1 is a minimal (x, y)-path. If $y \in V(P_2)$, or $y \rightarrow x$ then the claim is trivial so we assume that none of these hold. We can also assume that x and y are not similar vertices, because if they are, then y dominates the successor of x on P_2 and again the claim is trivial.

If D has a (y, z)-path whose first intersection with $V(P_1) \cup V(P_2)$ (starting from y) is on P_2, then the desired paths clearly exist. Hence we may assume that D contains a path from y to $V(P_1) \cup V(P_2) - y$ whose only vertex w from $V(P_1) \cup V(P_2) - y$ is in $V(P_1) - V(P_2)$. Now choose P among all such paths so that w is as close as possible to x on P_1. By the assumption above $w \neq x$. Let u (v) denote the predecessor of w on P_1 (P), i.e $u = w^-_{P_1}$ and $v = w^-_P$.

Suppose first that u and v are not adjacent. Then, by the remark just before Lemma 9.7.19, u and v are similar. Now the choice of P implies that $v = y$ (otherwise the predecessor of v on P dominates u, contradicting the choice of P). By the assumption that x and y are not similar we conclude that $u \neq x$, but then $u^-_{P_1} \rightarrow y$, contradicting the minimality of P_1.

Thus we may assume that u and v are adjacent. By the choice of P, this implies that $u \rightarrow v$. Choose r as the first vertex on P which is dominated by u. By the minimality of P_1, $r \neq y$. Let s be the predecessor of r on P. The choice of r and P implies that u and s are similar. Thus as above, we must have $s = y$ and, since $u \neq x$ we reach a contradiction as before.

The second half of the lemma follows from the first by considering the converse and interchanging the names of x and z. $\square$

Using Lemma 9.7.20 we can now characterize those extended locally in-semicomplete digraphs which do not have arc-disjoint (x,y)-,(y,z)-paths.

Theorem 9.7.21 *[51] An extended locally in-semicomplete digraph D has arc-disjoint (x,y)-, (y,z)-paths if and only if it has an (x,y)-path and a (y,z)-path and D has no arc e such that $D-e$ has no (x,y)-path and no (y,z)-path.*

Proof: Clearly if D has such an arc e, then the paths cannot exist. Now assume that D has no such arc and that D has an (x,y)-path and a (y,z)-path. We prove that D has the desired paths. By Lemma 9.7.19 we may assume $x \neq z$.

By Lemma 9.7.20, we may assume that D contains no pair of arc-disjoint (x,y)-,(x,z)-paths. Thus, by Menger's theorem, there exists an arc $e = uv$ such that $D - e$ has no path from x to $\{y,z\}$. Let $A = \{w : \exists(x,w) -$ path in $D - e\}$ and $B = V(D) - A$. Then $x \in A$, $y,z \in B$ and the only arc from A to B is e.

Since D contains an (x,y)-path, $D\langle A\rangle$ has an (x,u)-path and $D\langle B\rangle$ has a (v,y)-path. $D\langle B\rangle$ also has a (y,z)-path, since e does not destroy all paths from y to z.

If $v = y$ the desired paths clearly exist (and can in fact be chosen vertex disjoint). If $v = z$, then it follows from our assumption that there is no arc a in $D\langle B\rangle$ which separates y from z and also z from y. Now it follows from Lemma 9.7.19 that $D\langle B\rangle$ contains arc-disjoint (z,y)-, (y,z)-paths and hence D contains the desired paths. Thus we may assume $v \neq y,z$.

Now it is clear that the desired paths exist if and only if $D\langle B\rangle$ has arc-disjoint (v,y)-, (y,z)-paths. By induction this is the case unless there exists an arc $e' = ab$ in $D\langle B\rangle$ such that $D\langle B\rangle - e'$ has no path from v to y and no path from y to z, but then e' separates x from y and y from z in D, contradicting the assumption that D has no such arc. $\square$

Our proof above is constructive and hence we have the following (see also Exercise 9.40):

Corollary 9.7.22 *[46] There exists a polynomial algorithm which given an extended in-semicomplete digraph D and distinct vertices x,y,z decides whether D has arc-disjoint (x,y)-, (y,z)-paths (or equivalently, an (x,z)-trail through y).* $\square$

We can now prove the main result of this subsection.

Theorem 9.7.23 *[46] The arc-disjoint 2-path problem is polynomially solvable for semicomplete digraphs.*

Proof: (Sketch) Let $[D, x_1, x_2, y_1, y_2]$ be an instance of the arc-disjoint 2-path problem for semicomplete digraphs. By relabelling if necessary, we can assume that $x_1 \rightarrow x_2$. Below it is understood that we stop as soon as the existence of the desired paths have been decided.

It is easy to check whether D has (x_i, y_i)-paths for $i = 1, 2$. If not, then D does not have the desired paths and we stop. Next check whether there is any arc e such that $D - e$ has no (x_i, y_i)-path for $i = 1, 2$. If such an arc exists, then D does not have the desired paths and we stop. Now check whether D contains arc-disjoint (x_2, y_1)-, (x_2, y_2)-paths P, P'. If this is the case then either x_1P or $P[x_1, y_1]$ (if $x_1 \in V(P)$) and P' are the desired paths and we stop.

Hence, by Menger's theorem, there is an arc e such that $D - e$ has no path from x_2 to $\{y_1, y_2\}$. Let

$$Y := \{v : v \text{ has a path to } \{y_1, y_2\} \text{ in } D - e\}; \qquad X := V(D) - Y.$$

Then $x_2 \in X$ and $x_1 \in Y$, because the arc e does not separate x_1 from $\{y_1, y_2\}$. Furthermore, e is the only arc from X to Y. Let z be the head of e and let w be its tail. Note that $D\langle X \rangle$ contains an (x_2, w)-path Q since D contains an (x_2, y_2)-path.

If $z = x_1$, then the desired paths exist: We cannot have another arc e' which separates x_1 from $\{y_1, y_2\}$ in $D' = D\langle Y \rangle$ because then e' separates $\{x_1, x_2\}$ from $\{y_1, y_2\}$ and we would have stopped earlier. Thus by Menger's theorem D' contains arc-disjoint (x_1, y_1)-, (x_1, y_2)-paths P_1, P_2. Now P_1 and QP_2 are the desired paths.

If $z = y_2$, then the desired paths exist since any (x_1, y_1)-path in D' and Qy_2 will work.

If $z = y_1$, then the desired paths exist if and only if D' contains arc-disjoint (x_1, y_1)-, (y_1, y_2)-paths. This can be decided in polynomial time by the algorithm whose existence follows from Corollary 9.7.22.

Finally, if $z \notin \{x_1, y_1, y_2\}$, then the desired paths exist if and only if D' contains arc-disjoint (x_1, y_1),- (z, y_2)-paths. Hence we have reduced the problem to a smaller one of the same kind.

We leave it to the reader to verify that our steps above can be performed in polynomial time and to estimate the time complexity of the algorithm (Exercise 9.41). $\square$

9.8 Integer Multicommodity Flows

Recall the definition of a network and a flow from Chapter 3. In this section we consider briefly the following common generalization of flows and arc-disjoint paths called the **integer multicommodity flow problem** (if k is fixed in advance we call it the **integer *k*-commodity flow problem**):

Given a natural number $k \geq 1$, a network $\mathcal{N} = (V, A, \ell \equiv 0, u)$, $2k$ not necessarily distinct vertices $s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k$ and integers $r_1, r_2, \ldots, r_k$, decide whether there exist integer valued flows $f^1, f^2, \ldots, f^k$ such that each of the following holds (recall that $|f^i|$ is the value of the flow f^i):

(i) f^i is an (s_i, t_i)-flow in $\mathcal{N}$,
(ii) $|f^i| \geq r_i$, $i = 1, 2, \ldots, k$,
(iii) $f^p_{ij} \geq 0$ for every $ij \in A$, $p = 1, 2, \ldots, k$,
(iv) For every $ij \in A$: $\sum_{p=1}^{k} f^p_{ij} \leq u_{ij}$.

A collection of flows $f^1, f^2, \ldots, f^k$ which satisfies (i)-(iv) is called a **feasible k-commodity flow with respect to (s_i, t_i), $i = 1, 2, \ldots, k$**. We can also consider the **maximization version** where no demands $r_1, r_2, \ldots, r_k$ are specified (or they are to be considered as lower bounds) and the goal is to maximize the sum of the values of the flows.

If we take $k = 1$ we have the standard (maximum) (s, t)-flow problem which was studied in Chapter 3, where several polynomial algorithms were described for the problem. However, Even showed that already when $k = 2$ the problem becomes very hard.

Theorem 9.8.1 *[230] The integer 2-commodity problem is $\mathcal{NP}$-complete.*

Proof: The problem clearly belongs to $\mathcal{NP}$ since given a feasible instance we can take specifications of 2 feasible flows, one from s_1 to t_1 and the other from s_2 to t_2, as a valid certificate.

Now let $[D = (V, A), x_1, x_2, y_1, y_2]$ be an instance of the arc-disjoint 2-path problem. Let $\mathcal{N} = (V, A, \ell \equiv 0, u \equiv 1)$, take $s_i = x_i, t_i = y_i$, $i = 1, 2$ and let $r_1 = r_2 = 1$. Then it is easy to see that D has arc-disjoint (x_1, y_1)-, (x_2, y_2)-paths if and only if $\mathcal{N}$ has a feasible integer 2-commodity flow with respect to the pairs (s_i, t_i), $i = 1, 2$. Now the claim follows from Theorem 9.7.1. $\square$

What we really observed above was simply that the arc-disjoint 2-path problem is nothing but a very special case of the 2-commodity flow problem. This is not surprising since if we concentrate on one of the two flows f^i in a feasible integer 2-commodity flow (with respect to the values r_1, r_2 and the capacities of the given network), then f^i is just a normal (s_i, t_i)-flow and hence can be decomposed into r_i (s_i, t_i)-paths and some cycle flows by Theorem 3.3.1. Hence the integer multicommodity flow problem is nothing but a generalization of arc-disjoint path problems.

The name **multicommodity flow** comes from the interpretation of each flow as representing a different commodity that has to be shipped from the source of that commodity to its sink while respecting the total capacity of the network. Problems of this type are of importance in practical applications such as telecommunications and routing problems. For a number of results on how to solve multicommodity flow problems in practice see the book by Gondran and Minoux [332]. See also the survey [31] by Assad.

9.9 Arc-Disjoint In- and Out-Branchings

We saw in Section 9.5 that the problem of deciding the existence of k arc-disjoint out-branchings all with the same root could be solved efficiently and in Subsection 9.5.1 we saw that many problems can be reformulated and solved using an algorithm for the k arc-disjoint out-branchings problem. In this section we consider the following much harder problem concerning arc-disjoint in- and out-branchings.

Problem 9.9.1 *Given a digraph D and vertices u, v (not necessarily distinct). Decide whether D has a pair of arc-disjoint branchings F_u^+, F_v^- such that F_u^+ is an out-branching rooted at u and F_v^- is an in-branching rooted at v.*

Theorem 9.9.2 *[46] Problem 9.9.1 is $\mathcal{NP}$-complete for arbitrary digraphs.*

Proof: We give a proof due to Thomassen (see [46]). The problem belongs to $\mathcal{NP}$, since if the desired branchings exist, then such a pair forms a certificate that the given instance is a 'yes' instance. We show how to reduce the arc-disjoint 2-path problem to Problem 9.9.1 in polynomial time.

Let $(D = (V, A), x_1, x_2, y_1, y_2)$ be an instance of the arc-disjoint 2-path problem. Construct a new digraph D' by adding 4 new vertices x'_1, x'_2, y'_1, y'_2 and the following arcs (see Figure 9.16):
$\{x'_1x_1, x'_2x_2, y_1y'_1, y_2y'_2, x'_2x'_1, y'_1x'_1, y'_2y'_1, y'_2x'_2, y'_2x'_1\} \cup \{vx'_1 : v \in V(D) - x_1\} \cup \{y'_2v : v \in V(D) - y_2\}$.

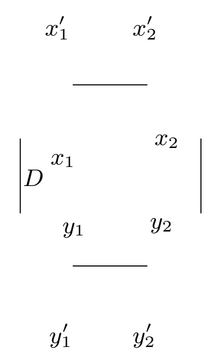


Figure 9.16 The construction of D' in the proof of Theorem 9.9.2. The fat arcs indicate that all the arcs have that direction, except the arcs $x'_1x_1, y_2y'_2$.

The reader can easily verify (Exercise 9.48) that there exists arc-disjoint branchings $F^+_{x'_2}$, $F^-_{y'_1}$ in D' if and only if D contains a pair of arc-disjoint (x_1, y_1)-, (x_2, y_2)-paths. Since we can construct D' in polynomial time from D, it follows that Problem 9.9.1 is $\mathcal{NP}$-complete. □

It is easy to reduce (in polynomial time) Problem 9.9.1 for the case when $u \neq v$ to the case when $u = v$ for arbitrary digraphs (Exercise 9.49). Hence the problem remains $\mathcal{NP}$-complete when we ask for an out-branching and an in-branching that are arc-disjoint and have the same root. However, Bang-Jensen and Huang showed that, if the vertex that is to be the root is adjacent to all other vertices in the digraph and is not in any 2-cycle, then the problem becomes polynomially solvable.

Theorem 9.9.3 *[79] Let $D = (V, A)$ be a strongly connected digraph and v a vertex of D such that v is not on any 2-cycle and $V(D) = \{v\} \cup N^-(v) \cup N^+(v)$. Let $\mathcal{A} = \{A_1, A_2, \dots A_k\}$ ($\mathcal{B} = \{B_1, B_2, \dots, B_r\}$) denote the set of terminal (initial) components in $D\langle N^+(v)\rangle$ ($D\langle N^-(v)\rangle$). Then D contains a pair of arc-disjoint branchings F_v^+, F_v^- such that F_v^+ is an out-branching rooted at v and F_v^- is an in-branching rooted at v if and only if there exist two disjoint arc sets $E_{\mathcal{A}}, E_{\mathcal{B}} \subset A$ such that all arcs in $E_{\mathcal{A}} \cup E_{\mathcal{B}}$ go from $N^+(v)$ to $N^-(v)$ and every $A_i \in \mathcal{A}$ ($B_j \in \mathcal{B}$) is incident with an arc from $E_{\mathcal{A}}$ ($E_{\mathcal{B}}$). Furthermore, there exists a polynomial algorithm to find the desired branchings, or demonstrate the non-existence of such branchings.*

Proof: We prove the characterization and refer the reader to [79] and Exercise 9.51 for the algorithmic part.

First we note that, if the branchings exist, then the arc sets $E_{\mathcal{A}}$ and $E_{\mathcal{B}}$ exist. Indeed, if F_v^+, F_v^- are such branchings, then there must be an arc from F_v^- (F_v^+) leaving (entering) every terminal (initial) component of $D\langle N^+(v)\rangle$ ($D\langle N^-(v)\rangle$) and since v is not on any 2-cycle, all these arcs go from $N^+(v)$ to $N^-(v)$.

Suppose that there exist sets $E_{\mathcal{A}}$ and $E_{\mathcal{B}}$ as above. Every vertex $x \in N^+(v)$ has a path to one of the terminal components in $\mathcal{A}$ and every vertex in $N^-(v)$ can be reached by a path from one of the initial components in $\mathcal{B}$. Hence, we can choose a family of vertex disjoint arborescences $F_1^-, F_2^-, \dots, F_k^-$, $F_1^+, F_2^+, \dots, F_r^+$ such that F_i^- (F_j^+) is an in-arborescence (out-arborescence) rooted at a vertex in A_i (B_j) and $\bigcup_{i=1}^k V(F_i^-) = N^+(v)$, $\bigcup_{j=1}^r V(F_j^+) = N^-(v)$. Let F_v^+ be the out-branching induced by the arcs $\{vw : w \in N^+(v)\} \cup E_{\mathcal{B}} \cup \bigcup_{j=1}^r E(F_j^+)$ and F_v^- be the in-branching induced by the arcs $\{uv : u \in N^-(v)\} \cup E_{\mathcal{A}} \cup \bigcup_{i=1}^k E(F_i^-)$. Then F_v^+ and F_v^- are the desired branchings. $\square$

The following is an easy corollary of Theorem 9.9.3.

Corollary 9.9.4 *[46] A tournament $D = (V, A)$ has arc-disjoint branchings F_v^+, F_v^- rooted at a specified vertex $v \in V$ if and only if D is strong and for every arc $a \in A$ the digraph $D - a$ contains either an out-branching or an in-branching with root v.* $\square$

There is a small inconsistency in the statement (and the proof) of Theorem 9.9.3 in [79] as it was not mentioned that v is not on a 2-cycle and the statement (the part involving the ends to the arcs in $E_{\mathcal{A}}$, $E_{\mathcal{B}}$) becomes slightly different when v is on 2-cycles. However, as the reader is asked to prove in Exercise 9.51, one can still describe a nice characterization and prove that it can be checked in polynomial time whether the desired branchings exist and to find such branchings if they exist. Since the discussion above takes care of the semicomplete case, a possible next step is to consider the following problem posed by Bang-Jensen.

Problem 9.9.5 *[65] Characterize those locally semicomplete digraphs D that have arc-disjoint branchings F_v^+, F_v^- for a given vertex $v \in V(D)$.*

When $u \neq v$, Problem 9.9.1 becomes much harder even for tournaments. The following complete characterization for the case of tournaments was found by Bang-Jensen. Note that the characterization is only valid for tournaments and not general semicomplete digraphs (in which case X, Y, Z, W is not a partition of $V - \{u, v\}$).

Theorem 9.9.6 *[46] Let $T = (V, A)$ be a tournament and let u, v be distinct vertices of T. Define the sets X, Y, Z, W as follows:*

$$X = \{x \in V : ux, vx \in A\}, \qquad Y = \{x \in V : ux, xv \in A\},$$
$$Z = \{x \in V : xu, vx \in A\}, \qquad W = \{x \in V : xu, xv \in A\}.$$

Then T has an out-branching F_u^+ and an in-branching F_v^- such that $A(F_u^+) \cap A(F_v^-) = \emptyset$ if and only if none of the following holds.

(1) $|V| \leq 3$ or $|V| = 4$ and $vu \in A$.
(2) T is not strong and either u is not in the initial strong component of T, or v is not in the terminal strong component of T.
(3) T is strong and there exists an arc e such that u is not in the initial strong component of $T - e$ and v is not in the terminal strong component of $T - e$.
(4) T is strong, $uv \in A$, $Y = \emptyset$, $X, W \neq \emptyset$ and (I) below holds

$$(I) \begin{cases} \text{There is exactly one arc } e_1 \text{ leaving the terminal strong component} \\ \text{of } T\langle X\rangle \text{ and there is exactly one arc } e_2 \text{ entering the initial strong} \\ \text{component of } T\langle W\rangle \text{ and } e_1 \neq e_2 \end{cases}$$

and finally every (X, W)-path in $T - \{u, v\}$ contains both e_1 and e_2.
(5) T is strong, $vu \in A$, $Y = \{y\}$, $X, W \neq \emptyset$, $[T, u, v]$ satisfies (I), there is no (X, W)-trail in $T - \{u, v\}$ which contains y and every (X, W)-path in $T - \{u, v\}$ contains both e_1 and e_2.
(6) T is strong, $vu \in A$, $Y = \emptyset$, $X, W \neq \emptyset$, $[T, u, v]$ satisfies (I), there exist a pair of arc-disjoint (u, v)-paths and for every choice of arc-disjoint (u, v)-paths P_1, P_2 either $e_1, e_2 \in A(P_1)$, or $e_1, e_2 \in A(P_2)$. □

By inspecting each of the exceptions above one easily derives the following sufficient condition for the existence of arc-disjoint in- and out-branchings in a tournament.

Corollary 9.9.7 *[46] Every 2-arc-strong tournament $T = (V, A)$ contains arc-disjoint in- and out-branchings F_r^-, F_s^+ for every choice of vertices $r, s \in V$.* □

Some of the conditions in Theorem 9.9.6 are quite complicated and even to prove the necessity requires some work (Exercise 9.50). We now show how to check the conditions in Theorem 9.9.6 efficiently. This together with the fact that the proof in [46] is constructive implies a polynomial algorithm for the arc-disjoint in and out-branching problem in tournaments.

Theorem 9.9.8 *[46] There is a polynomial algorithm for checking whether a given tournament with specified distinct vertices u, v has arc-disjoint branchings F_u^+, F_v^- and finding such branchings if they exist.*

Proof: The construction part of this proof relies on the fact that the proof of Theorem 9.9.6 in [46] is constructive and that proof is very long and technical. Hence we will only show how to check each of the conditions (1)-(6) in polynomial time (and hence checking whether or not the desired branchings exist). Conditions (1)-(4) are easy to check in polynomial time, so we concentrate on checking conditions (5) and (6). Let $[T, u, v]$ be an instance of the problem for which we wish to check conditions (5) and (6).

First we show how to check condition (5) using the polynomial algorithm from Corollary 9.7.22. Since every (X, W)-trail contains an (X, W)-path and every (X, W)-path contains e_1 and e_2 we conclude that every (X, W)-trail contains e_1 and e_2. That is, every (X, W)-trail must start at the tail x of e_1 and terminate at the head w of e_2. It is easy to show that there exists an (x, w)-trail that contains y if and only if there exist arc-disjoint (x, y)-, (y, w)-paths. Now we use the algorithm from Corollary 9.7.22 to check whether or not there exist arc-disjoint (x, y)-, (y, w)-paths. Condition (5) is satisfied if and only if there do not exist such paths.

Here is how to check condition (6) using the polynomial algorithm $\mathcal{A}$ from Theorem 9.7.23. It is easy to verify the existence of two arc-disjoint (u, v)-paths (use Lemma 7.1.5). In fact, if such paths do not exist then $[T, u, v]$ satisfies (3). Let X_l denote the terminal strong component of $T\langle X\rangle$ and W_1 the initial strong component of $T\langle W\rangle$ and let s be the number of strong components of $T\langle W\rangle$. Since $Y = \emptyset$ and there is only one arc leaving X_l and only one arc entering W_1, the existence of two arc-disjoint (u, v)-paths implies that $l, s \geq 2$, i.e., $X - X_l \neq \emptyset$ and $W - W_1 \neq \emptyset$. Let $T'' = T - X_l$ and check whether there exist two arc-disjoint (u, v)-paths in T''. If such paths exist then $[T, u, v]$ does not satisfy (6) and we stop. Let $T''' = T - W_1$ and check whether there exist two arc-disjoint (u, v)-paths in T'''. If such paths exist we stop because then $[T, u, v]$ does not satisfy (6).

By now we know (since we have not stopped yet) that for every pair P_1, P_2 of arc-disjoint (u,v)-paths e_1 and e_2 belong to $A(P_1) \cap A(P_2)$. That is $[T,u,v]$ satisfies (6) if and only if there do not exist arc-disjoint (u,v)-paths P_1, P_2 with $e_i \in P_i$, $i = 1,2$. We use $\mathcal{A}$ to check that possibility in the following way.

Since $[T,u,v]$ satisfies (I) we know that for every pair of arc-disjoint (u,v)-paths exactly one of these paths contains a vertex from X_l and exactly one contains a vertex from W_1. Moreover, if there exist arc-disjoint (u,v)-paths P_1, P_2 with $e_i \in A(P_i)$, $i = 1,2$ then we may assume that $P_1[u, X_l] = ux$ and $P_2[W_1, v] = wv$, where x is the tail of e_1 and w is the head of e_2.

Let T' be the tournament obtained from T as follows. Contract X_l into one vertex x_2 and W_1 into one vertex y_1. Furthermore, if there are arcs in both directions between x_2 and some $z \in T' - x_2$ we remove the arc zx_2. The arcs incident with y_1 are similarly modified. Let $x_1 = u$ and $y_2 = v$.

Now it is easy to see that there exist arc-disjoint (u,v)-paths P_1, P_2 in T satisfying $e_i \in A(P_i)$, $i = 1,2$, if and only if there exist arc-disjoint (x_1,y_1)-(x_2,y_2)-paths $P_{(x_1,y_1)}, P_{(x_2,y_2)}$ in T'. Now we use $\mathcal{A}$ to check whether or not such paths exist in T'.

It is easy to see that the above methods provide polynomial algorithms to verify conditions (5) and (6). $\square$

Bang-Jensen posed the following conjecture. This conjecture was verified by Bang-Jensen and Huang [79] for the special case when D is quasi-transitive and $u = v$.

Conjecture 9.9.9 *[65] Problem 9.9.1 is polynomially solvable for locally semicomplete digraphs and quasi-transitive digraphs.*

For the case when v is adjacent to all other vertices one can prove the following using Theorem 9.9.3 and the extension in Exercise 9.51 (see Exercise 9.52).

Theorem 9.9.10 *Let D be a 2-arc-strong digraph with a vertex v that is adjacent to all other vertices of D. Then D has arc-disjoint in- and out-branchings rooted at v.*

Thomassen conjectured that there is some sufficient condition, in terms of arc-strong connectivity, for the existence of arc-disjoint in- and out-branchings rooted at the same vertex in a digraph.

Conjecture 9.9.11 *[708] There exists a natural number N such that every digraph D which is N-arc-strong has arc-disjoint branchings F_v^+, F_v^- for every choice of $v \in V(D)$.*

For tournaments the following much stronger property has been conjectured by Bang-Jensen and Gutin:

Conjecture 9.9.12 *[65] There exists a function $f : \mathcal{Z}_+ \rightarrow \mathcal{Z}_+$ such that for every natural number k every $f(k)$-strongly arc-connected tournament T has $2k$ arc-disjoint branchings $F^+_{v,1}, \dots, F^+_{v,k}, F^-_{v,1}, \dots, F^-_{v,k}$ such that $F^+_{v,1}, \dots, F^+_{v,k}$ are out-branchings rooted at v and $F^-_{v,1}, \dots, F^-_{v,k}$ are in-branchings rooted at v, for every vertex $v \in V(T)$.*

It follows from Corollary 9.9.7 that $f(1) = 2$.

9.10 Minimum Cost Branchings

Given a directed multigraph $D = (V, A)$ a special vertex s and a non-negative cost function w on the arcs. What is the minimum cost of an out-branching F^+_v rooted at s in D? This problem, which is a natural generalization of the minimum spanning tree problem for undirected graphs (Exercise 9.58), is called **the minimum cost branching problem**. The problem arises naturally in applications where one is seeking a minimum cost subnetwork which allows communication from a given source to all other vertices in the network (see the discussion at the end of the section).

The minimum cost branching problem was first shown to be polynomially solvable by Edmonds [211]. Later Fulkerson [283] gave a two phase greedy algorithm which solves the problem very elegantly. The fastest algorithm for the problem is due to Tarjan [689]. Tarjan's algorithm solves the problem in time $O(m \log n)$, that is, with the same time complexity as Kruskal's algorithm for undirected graphs [169]. The purpose of this section is to describe a generalization of Fulkerson's algorithm (due to Frank [250]) which can be used to solve a more general problem.

9.10.1 Matroid Intersection Formulation

To illustrate the generality of matroids, let us show how to formulate the minimum cost branching problem as a weighted matroid intersection problem. We refer to Section 12.7 for relevant definitions on matroids.

Let $D = (V, A)$ be a directed multigraph and let $r \in V$ be a vertex which can reach all other vertices by directed paths. We define $M_1 = (A, \mathcal{I}_1)$ and $M_2 = (A, \mathcal{I}_2)$ as follows (here $\mathcal{I}_1, \mathcal{I}_2 \subseteq 2^A$):

- $A' \in \mathcal{I}_1$ if and only if no two arcs in A' have a common head and no arc has head r,
- $A'' \in \mathcal{I}_2$ if and only if $UG(D\langle A'' \rangle)$ has no cycle.

It follows from the definition of M_2 that M_2 is the circuit matroid of $UG(D)$ (see Section 12.7). It is easy to show that M_1 satisfies the axioms (I1)-(I3) and hence is a matroid. In particular, all maximal members of $\mathcal{I}_1$ have the same size $n - 1$ (by our assumption, every vertex in $V - r$ has at least one in-neighbour) and thus the rank of M_1 is $n - 1$.

Since r can reach all other vertices, $UG(D)$ is connected and hence the rank of M_2 is also $n-1$. We claim that every common base of M_1 and M_2 is an out-branching with root r. This follows easily from the definition of an out-branching and the fact that any common base corresponds to a spanning tree in $UG(D)$, since M_2 has rank $n-1$.

Thus we can find an out-branching with root r by applying the algorithm for matroid intersection of Theorem 12.7.11 to M_1, M_2. Of course such an out-branching can be found much easier by using e.g. DFS starting from r. However the point is that using the weighted matroid intersection algorithm, we can find a minimum cost out-branching F_r^+ in D. It is easy to see that the required oracles for testing independence in M_1 and M_2 can be implemented very efficiently (Exercise 9.55). In fact, and much more importantly (in the light of the existence of other and more efficient algorithms for minimum cost branchings), using matroid intersection algorithms we can even find a minimum cost subdigraph which has k-out branchings with a specified root s in a directed multigraph with non-negative weights on the arcs (Exercise 9.56). Furthermore, it is shown in Exercise 9.57 that using matroid intersection we can also solve the augmentation problem where one is given a directed multigraph $D=(V,A)$, a root $s \in V$ and a natural number k and the goal is to find a cheapest set of new arcs to add to D from an arc-weighted directed multigraph $D'=(V,A')$ on the same vertex set in order to ensure the existence of k-arc-disjoint out-branchings from s in the resulting directed multigraph. Hence using matroid intersection formulations one can in fact solve problems which are much more general than the minimum cost branching problem.

9.10.2 An Algorithm for a Generalization of the Min Cost Branching Problem

In this subsection we will give a generalization due to Frank [250] of Fulkerson's algorithm [283] for finding a minimum cost out-branching with a given root. This generalization, also allows one to determine the minimum cost of new arcs to add to a directed multigraph which has k-arc-disjoint out-branchings rooted at a vertex s, so as to have $k+1$ arc-disjoint out-branchings rooted at s.

To motivate the generalization below, we start with the augmentation problem above. We are given a directed multigraph $D=(V,A)$ a vertex $s \in V$ and a natural number k such that D has k, but not $(k+1)$ arc-disjoint out-branchings rooted at s (by Edmonds' branching theorem and Lemma 7.1.5, this condition can be checked efficiently using flows). Furthermore, we are given another directed multigraph $H=(V,A')$ on the same vertex set and a non-negative weight function $w: A' \rightarrow \mathcal{R}_0$ on A'. The goal is to find a minimum cost set of arcs F from A' so that the directed multigraph $D^*=(V, A \cup F)$ has $(k+1)$ arc-disjoint out-branchings rooted at s. In order to make sure that the problem has a solution we assume that $D''=(V, A \cup A')$ does have $(k+1)$ arc-disjoint out-branchings rooted at s. Note that if we take $H := D$ and then let $D := (V, \emptyset)$ and $k := 0$, we obtain the minimum cost branching problem. Hence the augmentation problem generalizes the minimum cost branching problem.

By (9.2) we have $d^-_D(X) \geq k$ for all $X \subseteq V - s$ and since D does not have $k+1$ arc-disjoint out-branchings from s, there must be some sets for which equality holds. Call such sets X (with $d^-_D(X) = k$) **tight**. Using submodularity of d^-_D it is easy to see that the family $\mathcal{F}_k = \{X \subset V - s : d^-_D(X) = k\}$ is an intersecting family (recall that this means that if $X, Y \in \mathcal{F}_k$ and $X \cap Y \neq \emptyset$, then $X \cap Y, X \cup Y \in \mathcal{F}_k$). In view of Edmonds' branching theorem our goal is to find a minimum cost subset F of A' such that $d^-_F(X) \geq 1$ for each $X \in \mathcal{F}_k$ (after doing this we will have $d^-(X) \geq k+1$ for every $X \subseteq V - s$ in the resulting directed multigraph).

We now see that our problem is a special case of the following more general problem (we obtain the problem above if we take $\mathcal{F} = \mathcal{F}_k$):

Problem 9.10.1 *Given a set V, an intersecting family $\mathcal{F} \subseteq 2^V$ and a directed multigraph $H = (V, A')$ together with a weight function $w : A' \rightarrow \mathcal{R}_0$. Find a minimum cost subset $F \subseteq A'$ such that $d^-_F(X) \geq 1$ for each*[9] $X \in \mathcal{F}$.

To ensure the existence of a solution, we must assume that $d^-_{A'}(X) \geq 1$ for each $X \in \mathcal{F}$.

We solve this generalization instead of just the minimum cost branching problem. The motivation for this is to show the reader that often considering an abstraction of a problem will allow one to solve a more general problem (see the next subsection). Furthermore the solution for the abstraction can often be simpler (or at least not more difficult), since we have gotten rid of the special requirements of the original problems (of course these are still inherent in the abstraction, but we have more freedom here).

In order to describe the two phase greedy algorithm below for solving Problem 9.10.1 we let M be a matrix whose rows are indexed by the members of $\mathcal{F} = \{X_1, X_2, \ldots, X_q\}$ and whose columns are indexed by the arcs $e_1, e_2, \ldots, e_m$ in H. We let $M_{X_i, e_j} = 1$ precisely if the arc e_j enters the set X_i.

Now we can formulate Problem 9.10.1 as the following linear programming problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in A'} w(e)x(e) \\ \text{subject to} \quad & Mx \geq 1 \qquad \text{for all } X \in \mathcal{F} \\ & x \geq 0. \end{aligned} \tag{9.12}$$

We are only interested in integer solutions, but as we are going to see, provided all weights are integers, there are integer valued optimum solutions

[9] That is, we want to **cover** every member of $\mathcal{F}$ by an arc of H.

to the system below (this also follows from the fact that the system is a TDI system (see Schrijver's book [659]), but we prove the integrality of the solution directly). Note that, if x is an optimal solution to (9.12), then $x \leq 1$. This follows from the fact that the costs are non-negative. Hence if x is a function on the arcs of A' we can say that $x(e_i) = 1$ ($x(e_i) = 0$) corresponds to including (excluding) e_i in the solution.

The dual of (9.12) is

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{q} y(X_i) \\ \text{such that} \quad & y^T M \leq w(e) \qquad\qquad \text{for all } e \in A' \\ & y \geq 0. \end{aligned} \tag{9.13}$$

Here $y(X_i)$ denotes the dual variable associated with the set X_i of $\mathcal{F} = \{X_1, X_2, \dots, X_q\}$. Note that, if we let

$$m(y, e) = \sum_{\{X \in \mathcal{F}: e \text{ enters } X\}} y(X), \tag{9.14}$$

then the first constraint in the dual problem says that we must have $m(y, e) \leq w(e)$ for every arc e.

The two phase greedy algorithm below works by first finding a feasible solution to the dual in a greedy way and then solving (greedily) the primal problem using the dual solution that we obtained in the first phase.

A pair (x, y) of solutions to (9.12) and (9.13), respectively, are optimal (for the primal, respectively, the dual) if and only if it satisfies the complementary slackness conditions, see e.g. [160]:

(I) For every arc $e \in A'$: $x(e) > 0$ implies $m(y, e) = w(e)$ and
(II) For every $X \in \mathcal{F}$: $y(X) > 0$ implies[10] $x^-(X) = 1$.

We now describe the algorithm and show that at termination, the final vectors x, y are integral and satisfy (1), respectively (2), and hence they are optimal solutions to the primal, respectively the dual, problems. The description given here is based on notes by the first author from a lecture given by Frank in Grenoble, June 1996. See also Frank's paper [250].

The Frank-Fulkerson algorithm

Phase 1: Start with $y \equiv 0$. In the initial step we choose $A_1 \in \mathcal{F}$ to be a minimal member of $\mathcal{F}$, that is, no proper subset of A_1 belongs to $\mathcal{F}$ (see

[10] Here $x^-(U)$ denotes the sum of the x values on all arcs entering the set U. See also Chapter 8.

Exercise 9.59 for an algorithm to find such a minimal member when $\mathcal{F}$ is the family of tight sets avoiding a fixed vertex s in a directed multigraph). Choose an arc e_1 which enters A_1 (i.e. $d^-_{e_1}(A_1) = 1$) such that $w(e_1)$ is minimum among all arcs of A' which enter A_1. Set $y(A_1) := w(e_1)$.

In the general step, we assume that $(A_1, e_1, y(A_1)), (A_2, e_2, y(A_2)), \dots, (A_{i-1}, e_{i-1}, y(A_{i-1}))$ have been determined. Let $A'' = \{e_1, e_2, \dots, e_{i-1}\}$. If $d^-_{A''}(X) \geq 1$ for all $X \in \mathcal{F}$, then Phase 1 has been completed and we go to Phase 2. Assume this is not the case and choose $A_i \in \mathcal{F}$ as a minimal member of $\mathcal{F} - \{A_1, A_2, \dots, A_{i-1}\}$ which has $d^-_{A''}(A_i) = 0$.

By the assumption that $d^-_{A'}(X) \geq 1$ for every $X \in \mathcal{F}$, there is at least one arc from A' which enters A_i. Choose among all such arcs one, e_i, which minimizes $w(e_i) - m(y, e_i)$. Let $y(A_i) := w(e_i) - m(y, e_i)$ (possibly $y(A_i) := 0$). Note that it is easy to find e_i since there are currently (at most) $i-1$ sets for which y is non-zero and hence $m(y, e)$ can be calculated easily for every arc $e \in A'$ which enters A_i. Let $i := i + 1$ and continue the general step. This completes the description of Phase 1. See Figure 9.17 for an example (for an instance of the minimum cost branching problem) of an execution of Phase 1.

Before we go on to describe the second phase of the algorithm, we make some useful observations. Let $\mathcal{L} = \{A_1, A_2, \dots, A_t\}$ be the family of sets chosen when Phase 1 terminates (there are only finitely many sets in $\mathcal{F}$, so termination follows from the assumption that A' has at least one arc entering every member of $\mathcal{F}$). Recall that a family $\mathcal{H}$ of sets is laminar if $A, B \in \mathcal{H}$ and $A \cap B \neq \emptyset$ implies $A \subseteq B$ or $B \subseteq A$.

Claim A: $\mathcal{L}$ is a laminar family.

Proof: Suppose $A_i - A_j, A_i \cap A_j, A_j - A_i$ are all non-empty for some $1 \leq i < j \leq t$. Since $\mathcal{F}$ is intersecting we have $A_i \cap A_j \in \mathcal{F}$. Recall that A_i is minimal in $\mathcal{F} - \{A_1, A_2, \dots, A_{i-1}\}$ at the time it is chosen and none of the arcs $e_1, e_2, \dots, e_{i-1}$ chosen so far enters A_i. Hence there must exist a $p < i$ such that e_p enters $A_i \cap A_j$, but not A_i (which means that the tail of e_p is in $A_i - A_j$). But now the arc e_p enters A_j, contradicting the fact that at the (later) time when we consider A_j no previously chosen arc enters that set. This contradiction implies the claim. □

Claim B: After Phase 1 the vector y is a feasible solution to (9.13).

Proof: It follows from the way we assign values to y that we will always have $y(X) \geq 0$ for every $X \in \mathcal{F}$. Hence it suffices to prove that $m(y, e) \leq w(e)$ for all arcs $e \in A'$. Note that y is zero on $\mathcal{F} - \mathcal{L}$, so we only have to consider the contribution from y on the sets in $\mathcal{L}$. Since $\mathcal{L}$ is laminar, those sets from $\mathcal{L}$ which are entered by a given arc e form a chain $A_{i_1} \subset A_{i_2} \subset \ldots \subset A_{i_r}$. Furthermore, it follows from the way we choose the A_i's in Phase 1 that $1 \leq i_1 < i_2 < \ldots < i_r \leq t$.

Figure 9.17 An execution of Phase 1 (for the minimum cost branching problem) on the digraph shown in the upper left corner with root r. Fat arcs show the arc chosen in that step. Normal arcs are arcs that have already been chosen at this point in time.

Consider now a fixed arc e. We wish to show that $m(y,e) \leq w(e)$ remains true during all of Phase 1. Clearly the initial choice $y \equiv 0$ satisfies this. When we consider the first set A_{i_1} from $\mathcal{L}$ which contributes to $m(y,e)$, we choose $y(A_{i_1})$ so as to maintain the inequality $m(y,e) \leq w(e)$ (because $y(A_{i_1})$ is assigned the minimum of $w(e') - m(y,e')$ over all e' (including e) which enter A_{i_1}. Then when we choose $y(A_{i_2})$, the value assigned to $y(A_{i_1})$ now contributes to $w(e) - m(y,e)$ and again we can argue that we will have $m(y,e) \leq w(e)$ after assigning the value to $y(A_{i_2})$. Now it is easy to prove by induction on r (the number of sets in $\mathcal{L}$ which are entered by e) that $m(y,e) \leq w(e)$ remains true throughout Phase 1. Since e was an arbitrary arc, we have proved that y is feasible for the dual (9.13). $\square$

Let $e_1, e_2, \ldots, e_t$ be the arcs chosen in Phase 1. Call an arc $e \in A'$ **tight** if $m(y,e) = w(e)$. Note that each of $e_1, e_2, \ldots, e_t$ are tight by the way we choose the dual variables (recall how we assign the value to $y(A_i)$).

Hence the complementary slackness condition (I) will be trivially satisfied if we can find a subset $F \subseteq \{e_1, e_2, \ldots, e_t\}$ such that x is non-zero only on arcs in F and x is a feasible solution to the primal. In order to ensure optimality, we must choose x and F so that (II) also holds, that is, we must have that $y(X) > 0$ implies $x^-(X) = 1$ for every $X \in \mathcal{F}$. We are now ready to describe the second and last phase of the algorithm.

Phase 2:
Let $F := \emptyset$.
Consider the arcs $e_t, e_{t-1}, \ldots, e_2, e_1$ in that order. After having considered $e_t, e_{t-1}, \ldots, e_{i+1}$ we add e_i to F if and only if $d^-_F(A_i) = 0$ (that is, the arc e_i is only added to F if no arc with a higher index enters A_i). See Figure 9.18 for an illustration of Phase 2 (corresponding to the same example as Figure 9.17).

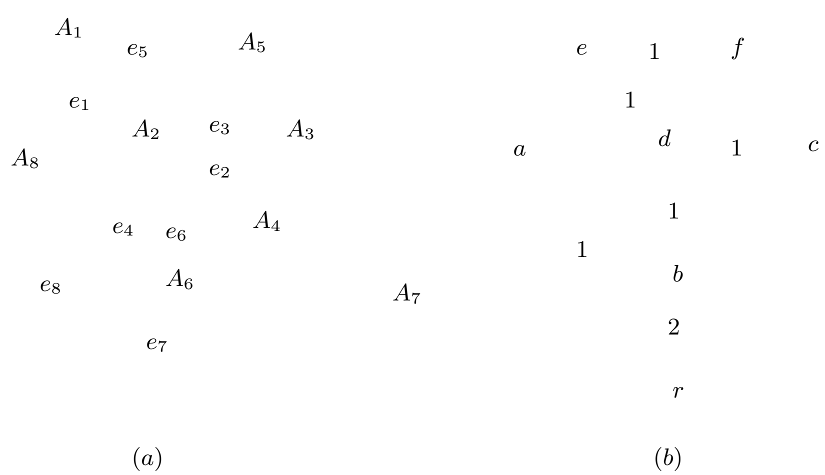


Figure 9.18 Phase 2 on the digraph from Figure 9.17. Part (a) shows the input to phase 2. Part (b) shows the output from Phase 2.

Now set $x(e) := 1$ if $e \in F$ and $x(e) := 0$ otherwise. This concludes Phase 2.

It is easy to see that the choice for x and F made above satisfies (II) since y is non-zero only on set which belong to $\mathcal{L}$.

Claim C: x is a feasible solution to (9.12).

By the definition of x our claim is equivalent to saying that $d^-_F(Z) \geq 1$ for all $Z \in \mathcal{F}$. If no arc e_i was disposed of (i.e. not chosen) in Phase 2, then this follows from the fact that at the termination of Phase 1, every member of $\mathcal{F}$ is entered by at least one of the arcs $e_1, e_2, \ldots, e_t$. Hence we may assume that at least one arc e_j was disposed of in Phase 2 and that some member Z of $\mathcal{F}$ has $d_F^-(Z) = 0$. We show that this leads to a contradiction.

Let Z be a maximal member of $\mathcal{F}$ such that $d_F^-(Z) = 0$ holds.

We first prove that there exits an index i such that $A_i \subset Z$ and e_i enters Z. Choose i as small as possible so that $e_i \notin F$ and e_i enters Z. Suppose that $A_i \not\subset Z$. Then $A_i - Z, A_i \cap Z, Z - A_i$ are all non-empty (the last set is non-empty by the minimality of i and the definition of A_i). Since $\mathcal{F}$ is intersecting, the set $A_i \cap Z$ belongs to $\mathcal{F}$. Now the minimality of A_i (at the time it was chosen in Phase 1) implies that there is some $j < i$ such that e_j enters $A_i \cap Z$, but not A_i. This means that the tail of e_j belongs to $A_i - Z$ (and hence e_j enters Z). However, since $d_F^-(Z) = 0$ we have $e_j \notin F$, contradicting the choice of i. Thus we have shown that there exists an index i so that $A_i \subset Z$ and e_i enters Z.

Now choose among all pairs (A_i, e_i) such that $A_i \subset Z$ and e_i enters Z, the one which has the highest index p. Since e_p was not added to F in Phase 2, there must exist an index $j > p$ such that $e_j \in F$ and e_j enters A_p. Using that $\mathcal{L}$ is laminar and $j > p$ we get that $A_p \subset A_j$ (we cannot have $A_j \subset A_p$, by the way we choose A_p in Phase 1).

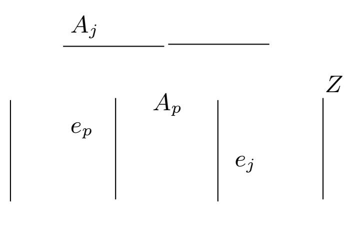


Figure 9.19 The positions of the sets A_j, A_p and Z and the arcs e_j, e_p.

Note that $A_j \not\subset Z$ since otherwise e_p would enter A_j, contradicting the fact that $j > p$ and A_j had in-degree zero when we choose it in Phase 1. Furthermore, e_j does not enter Z since $d_F^-(Z) = 0$. Thus we must have the picture in Figure 9.19. Now it follows that the member $Z \cup A_j \in \mathcal{F}$ is not entered by any arc from F (recall that e_j is the unique arc from F which enters A_j). This contradicts the maximality of Z.

Hence we have shown that the set Z cannot exist and thus x is indeed a feasible solution to (9.12) and Claim C is proved. $\square$

It follows from Claims B and C that the pair (x, y) of primal, respectively dual, solutions satisfies the complementary slackness conditions (I) and (II) and hence are optimal solutions to the problems (9.12) and (9.13) respectively. This proves the correctness of the algorithm.

Let D be a directed multigraph with a special vertex s. By an ***s*-cut** we mean an arc set of the form $(\overline{U}, U)$ where $U \subset V - s$ (that is, an s-cut is the set of all the arcs entering U for some U not containing s). The following min-max result due to Fulkerson [283] is a consequence of our arguments above where we proved the existence of optimal integer valued solutions for both the primal and the dual. It is instructive to check the statement of the theorem on the example in Figures 9.17 and 9.18.

Theorem 9.10.2 *[283] Let $D = (V, A)$ be a directed multigraph with a special vertex $s \in V$ which can reach all other vertices of V and a non-negative integer weight function $w : A \rightarrow \mathcal{Z}_0$ on the arcs. The minimum weight of an out-branching with root s is equal to the maximum number of s-cuts (with repetition allowed) so that no arc a is in more than $w(a)$ of these cuts.* □

9.10.3 The Minimum Covering Arborescence Problem

As we can see from Exercise 9.60 (and Tarjan's algorithm in [689]), we can find an optimal branching quite efficiently. It is also easy to decide if a digraph has some arborescence rooted at a prescribed vertex s which covers (that is, contains the vertices of) a certain specified subset X of the vertex set (Exercise 9.54). This makes it natural to consider the following problem which we call the **minimum covering arborescence problem**. Given a digraph $D = (V, A)$ with a non-negative integer valued weight function w on the arcs, some vertex $s \in V$ and a subset $X \subseteq V$. What is the cost of a minimum out-arborescence F_s^+ rooted in s such that $X \subseteq V(F_s^+)$?

Theorem 9.10.3 *The minimum covering arborescence problem is $\mathcal{NP}$-hard even when $w \equiv 1$.*

Proof: We show how to reduce the **graph Steiner problem** to the special case $w \equiv 1$ of the minimum covering arborescence problem in polynomial time. The graph Steiner problem is as follows (this is a special case, but already this is $\mathcal{NP}$-complete). Given an undirected graph $G = (V, E)$ and a subset $X \subset V$, find a subtree of G which contains all vertices of X and as few other vertices as possible.

Let $[G, X]$ be an instance of the graph Steiner problem and construct an instance $[D, X, s]$ of the minimum covering arborescence problem by letting D be the complete biorientation of G, taking s as some vertex from X and using the same X. Every tree T which covers X in G corresponds in the obvious way to an out-arborescence F_s^+ in D which covers X and vice versa. This completes the construction which can obviously be performed in polynomial time. Since the graph Steiner problem is $\mathcal{NP}$-hard [474] we conclude that so is the minimum covering arborescence problem. □

It follows from Frank's results in [265] that, if the cost of all arcs whose head do not belong to X is zero, then the problem can be solved in polynomial time. In fact, the model in [265] shows that even the generalization where one is seeking k arc-disjoint arborescences with a common root all of which cover a prescribed subset X can be solved in polynomial time, provided the cost of all arcs whose head do not belong to X is zero.

In real-life applications such as telecommunications, one is often interested in serving only a subset of the customers from a given source and furthermore not all customers have the same demand. This gives rise to the following more general problem which is called **the directed Steiner problem with connectivity constraints** (DSCC) in [171]. Given a directed graph $D = (V, A)$ with weights on the arcs, a special vertex s and a number k_v associated with each vertex $v \in V - s$, find a minimum cost subset $A' \subseteq A$ such that $D\langle A' \rangle$ contains k_v arc-disjoint (s, v)-paths for all $v \in V - s$. It follows from our remarks above that this problem is $\mathcal{NP}$-complete even if we only allow $k_v \in \{0, 1\}$ for each $v \in V - s$. In [171] Dahl discusses a cutting plane approach to solving the DSCC problem. It is also shown in [171] how to formulate another classical problem from operations research, the uncapacitated facility location problem, as an instance of the DSCC problem (see Exercise 9.71).

Let us conclude this section with a few remarks on the directed Steiner problem. The **directed Steiner problem** is as follows. Given a directed multigraph $D = (V, A)$ and a subset S of its vertices, find a minimum subset A' of A such that $D' = (V, A')$ contains an (s, t)-path for every choice of $s, t \in S$. The vertices in S are called **terminals**. Clearly this problem is $\mathcal{NP}$-hard as it contains the graph Steiner problem as a special case. In Exercise 9.69 the reader is asked to describe a polynomial algorithm for the case when $|S| = 2$. Recently Feldman and Ruhl [233] proved that for every fixed k the directed Steiner problem with k terminals is solvable in polynomial time. In fact they proved that the following more general problem is polynomially solvable for every fixed p. Given a directed multigraph $D = (V, A)$ and p pairs $\{(s_1, t_1), \ldots, (s_p, t_p)\}$ find a smallest set of arcs A' in A such that $D' = (V, A')$ contains an (s_i, t_i)-path for $i = 1, 2, \ldots, p$. Feldman and Ruhl also showed that the weighted version is still polynomial (provided p is fixed).

9.11 Increasing Rooted Arc-Strong Connectivity by Adding New Arcs

The approach in the last section does not allow us to solve the augmentation problem where one starts with an arbitrary digraph with a special vertex s and the goal is to add arcs so that the new digraph has k-arc-disjoint out-branchings rooted at s. Only the case when there are already $k-1$ arc-disjoint out-branchings from s in D is covered above.

The following theorem, answering the general case, can be derived from Theorem 7.6.3 (Exercise 9.73). We give a direct proof below since it is quite simple and illustrates once again the usefulness of submodularity in proofs.

Theorem 9.11.1 *Let $D = (V, A)$ be a digraph with a special vertex s. Let k be a natural number. The minimum number of new arcs $\gamma_{s,k}(D)$ one has to add to D in order to obtain a new digraph $D' = (V, A \cup F)$ which has k arc-disjoint out-branchings rooted at s satisfies $\gamma_{s,k}(D) = \gamma$, where*

$$\gamma = \max\{\sum_{X \in \mathcal{F}} \max\{0, k - d_D^-(X)\} : \mathcal{F} \text{ is a subpartition of } V - s\}. \quad (9.15)$$

Furthermore, an optimal augmenting set F can always be chosen such that all new arcs have tails at s.

Proof: Let $[D, k, s]$ be given. By Edmonds' branching theorem, we must have $d_{D'}^-(X) \geq k$ for all $X \subseteq V - s$. Hence $\gamma_{s,k}(D) \geq \gamma$ must hold. We prove below that there exists a good augmenting set with no more than γ arcs. It is instructive to compare this proof with the proof of Theorem 7.6.3.

Let $v_1, v_2, \ldots, v_{n-1}$ be a fixed labeling of $V - s$. Add k parallel arcs from s to every other vertex. Clearly the digraph obtained in this way satisfies (9.2). To distinguish the added arcs from arcs in A we refer to them below as **new arcs**. Starting with $i = 1$ we delete as many new arcs of the kind sv_i as possible while preserving (9.2) in the current digraph. If $i < n - 1$ let $i := i + 1$ and repeat the deleting step; otherwise stop. Let F be the final set of new arcs after the deletion phase and let $D^* = (V, A \cup F)$ denote the current digraph when this process stops. We will show that $|F| \leq \gamma$. This will complete the proof and also imply the second claim since all arcs in F have tail s.

Since no remaining new arc sv can be removed without violating (9.2), it must enter a set X such that $d_{D^*}^-(X) = k$. Call a set $X \subseteq V - s$ **critical** if $d_{D^*}^-(X) = k$. Let $S := \{v : sv \in F\}$, that is, S is the set of all vertices that are entered by an arc from F. Choose a family of critical sets $\mathcal{F} = \{X_1, X_2, \ldots, X_t\}$ such that $\mathcal{F}$ covers S[11] and t is minimum with respect to this condition.

We claim that $\mathcal{F}$ is a subpartition of $V - s$. Suppose that this is not the case. Then $\mathcal{F}$ contains two sets X_i, X_j, $i \neq j$ such that $X_i \cap X_j \neq \emptyset$. However using the submodularity of $d_{D^*}^-$ we obtain

$$\begin{aligned} k + k = d_{D^*}^-(X_i) + d_{D^*}^-(X_j) &\geq d_{D^*}^-(X_i \cap X_j) + d_{D^*}^-(X_i \cup X_j) \\ &\geq k + k. \end{aligned}$$

Hence $d_{D^*}^-(X_i \cup X_j) = k$ and we can replace X_i, X_j by the set $X_i \cup X_j$, contradicting the choice of $\mathcal{F}$ (note that $X_i \cup X_j \subseteq V - s$ and hence $d_{D^*}^-(X_i \cup X_j) \geq k$ must hold). Thus $\mathcal{F}$ is indeed a subpartition of $V - s$.

Now we have

[11] A collection of sets $\mathcal{F}$ **covers** a set S if every $s \in S$ belongs to some member X of $\mathcal{F}$.

$$kt = \sum_{i=1}^{t} d^-_{D^*}(X_i) = |F| + \sum_{i=1}^{t} d^-_D(X_i),$$

since every edge from F enters precisely one set $X_i \in \mathcal{F}$ and each X_i has $d^-_{D^*}(X_i) = k$. Thus

$$|F| = \sum_{i=1}^{t}(k - d^-_D(X_i)) \leq \gamma.$$

and the proof is complete. □

The method used to prove Theorem 9.11.1 cannot be extended to the case when the new arcs have costs and hence we cannot solve the cost version case of the problem in this way. As we remarked at the end of Subsection 9.10.1 this problem can be solved using an algorithm for weighted matroid intersection (Exercise 9.57). Hence weighted matroid intersection algorithms are a quite powerful tool.

Frank [265] has shown that, using a similar (but more complicated) approach to that used in Section 9.10, one can also solve the problem in which the goal is to add a minimum cardinality set of new arcs to a digraph $D = (V, A)$ with a special vertex s with $\kappa(s, v) \geq k$ for all $v \in V - s$, so as to increase $\kappa(s, v)$ to at least $k + 1$ for every $v \in V - s$. As we mentioned in Chapter 7 this problem can be solved with the help of submodular flows [275], but the approach in [265] is simpler, since it does not require the (rather complicated) algorithms for submodular flows.

9.12 Exercises

9.1. Prove Proposition 9.2.1.

9.2. Prove that problem (P5) of Proposition 9.2.1 for semicomplete digraphs can be reduced to the 2-path problem for semicomplete digraphs in polynomial time.

9.3. Prove Proposition 9.2.2.

9.4. Prove Lemma 9.2.4.

9.5. Prove Theorem 9.2.7. Hint: use Lemma 9.2.8.

9.6. Prove Theorem 9.2.9 without using Theorem 9.2.10.

9.7. Let D be the acyclic digraph in Figure 9.20. Show that the digraph D' defined as in the proof of Theorem 9.2.14 has a directed path from (x_1, x_2, x_3) to (y_1, y_2, y_3).

9.8. (+) Argue that we do not really need to construct D' when searching for a path from $(x_1, x_2, \ldots, x_k)$ to $(y_1, y_2, \ldots, y_k)$. Does that lead to an improvement in the complexity estimate?

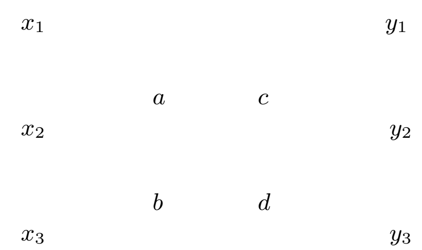


Figure 9.20 An instance of the 3-path problem for acyclic digraphs.

9.9. **Finding a cheapest collection of k disjoint paths with prescribed ends in weighted acyclic digraphs.** Show that the approach used in the proof of Theorem 9.2.14 can be modified so that one can find the cheapest collection of disjoint paths joining x_i to y_i for $i = 1, 2, \ldots, k$.

9.10. (+) Prove that under the assumption of Corollary 9.2.16, for every non-special vertex v, the digraph D contains directed (x_1, v)-, (x_2, v)-, (v, y_1)-, (v, y_2)-paths such that the only common vertex of any two of these paths is v (Lucchesi and Giglio [527]). Hint: use Menger's theorem and the fact that D is acyclic.

9.11. **A sufficient condition for digraph to be 2-linked.** Let $D = (V, A)$ satisfy $d^+(x) + d^-(y) \geq n + 2$ whenever D does not contain the arc xy. Prove that D is 2-linked. Hint: first show that, if $xy \notin A$, then there are three internally disjoint (x, y)-paths of length 2 in D (Heydemann and Sotteau [426]).

9.12. Prove that every k-linked digraph is also k-strong.

9.13. Prove that,if a digraph $D = (V, A)$ is 2-linked, then for every choice of distinct vertices x, y, D contains disjoint cycles C_x, C_y such that $x \in V(C_x), y \in V(C_y)$. Generalize this to k-linked digraphs and k vertices.

9.14. (−) **Disjoint cycles containing prescribed vertices in tournaments.** Prove that a tournament T contains disjoint cycles C_x, C_y such that $x \in V(C_x), y \in V(C_y)$ if and only if T contains disjoint 3-cycles such that one contains x and the other contains y.

9.15. Describe how to construct the collection $Q'_1, Q'_2, \ldots, Q'_q$ of subpaths in the proof of Proposition 9.3.1. What is the complexity of your algorithm?

9.16. Show how to turn the proof of Proposition 9.3.1 into an algorithm which takes as input a collection $P_1, P_2, \ldots, P_p$ of internally disjoint (x, y)-paths and a collection $Q_1, Q_2, \ldots, Q_q$ of internally disjoint (u, v)-paths in $D - \{x, y\}$ and finds a collection of q (u, v)-paths which intersect no more than $2q$ vertices of $P_1, P_2, \ldots, P_p$.

9.17. Let D be a locally semicomplete digraph and let x, y be distinct non-adjacent vertices. Prove that every minimal (x, y)-path is an induced path (Bang-Jensen [44]).

9.18. **(−)** Let D be a locally semicomplete digraph such that $\alpha(D) = 2$. Prove that if x and y are non-adjacent vertices of D and D has an (x, y)-path, then there exists an (x, y)-path P of length at most 3.

9.19. **(+)** Prove the following statement. Let $k \geq 3$, let D be a k-strong locally semicomplete digraph which is round decomposable and let $D = R[S_1, \ldots, S_r]$ be the round decomposition of D. Let x and y be vertices such that $x \in V(S_i)$ and $y \in V(S_j)$, where $i \neq j$ and let P be a minimal (x, y)-path. Then $D - V(P)$ is $(k-2)$-strong (Bang-Jensen [52]). Hint: use Exercise 9.18

9.20. **(+)** Prove Lemma 9.3.3. Hint: use Exercise 9.19.

9.21. Prove Lemma 9.3.17.

9.22. **(++)** Prove Theorem 9.3.13.

9.23. Prove Lemma 9.4.3. Hint: show how to modify a given (x, y)-path which is not closest to R into one which is closer by a stepwise (but finite and polynomially bounded) improvement. For the algorithmic part you can use that the embedding is with polygonal curves.

9.24. Prove that the graph G_4 in Figure 9.10(a) is 5-connected.

9.25. Prove that the digraph D_k is Figure 9.10(b) is 5-strong and has no cycle through x, y. Hint: use Exercise 7.26 and Proposition 9.4.1.

9.26. Show how to derive Menger's theorem (Theorem 7.3.1) from Edmonds' branching theorem (Theorem 9.5.1).

9.27. **(+) A polynomial algorithm for finding k-arc-disjoint out-branchings from a specified root.** Show how to turn the proof of Theorem 9.5.1 into a polynomial algorithm which either finds a collection of k arc-disjoint branchings with root z, or a proof that no such collection of branchings exists. Hint: use flows.

9.28. **Greedy branching algorithm.** Instead of applying the algorithmic version of Theorem 9.5.1 to find k arc-disjoint out-branchings with a given root, one may try a greedy approach: find an out-branching F_z^+ from z. Delete all arcs of F_z^+. Find a new out-branching, delete its arcs and so on. Give an example of a digraph D which has 2-arc-disjoint out-branchings with root z, but not every out-branching F_z^+ can be deleted while leaving another with root z.

9.29. **(+) Tutte's theorem on edge-disjoint trees in undirected graphs.** Derive Theorem 9.5.6 from Theorem 8.7.6.

9.30. Generalize the example in Figure 9.11 to digraphs with arbitrarily many vertices.

9.31. Prove Lemma 9.6.2.

9.32. Show how to use submodular flows to decide in polynomial time whether a mixed graph M has k edge-disjoint mixed branchings from a given root. Hint: see Exercise 8.65 and adjust the upper/lower bounds on arcs appropriately.

9.33. Give a proof of Theorem 9.6.3 using the reduction you found in the previous exercise and the feasibility theorem for intersecting submodular flows.

9.34. **(+) Arc disjoint out-branchings with possibly different roots.** Prove the following result due to Frank [252]: In a directed graph $D = (V, A)$ there are k arc-disjoint out-branchings (possibly with different roots) if and only if

$$\sum_{i=1}^{t} d^-(X_i) \geq k(t-1) \tag{9.16}$$

holds for every subpartition $\{X_1, X_2, \ldots, X_t\}$ of V. Hint: add a new vertex s and a minimal set of new arcs from s to V so that s is the root of k out-branchings in the new graph. Prove that this minimal set of arcs has precisely k arcs.

9.35. Prove Theorem 9.5.6. Hint: use Edmonds' branching theorem and Theorem 8.7.6.

9.36. Supply the missing details in the proof of Theorem 9.7.1.

9.37. Prove Theorem 9.7.3.

9.38. (−) Prove Lemma 9.7.6.

9.39. Prove Lemma 9.7.19.

9.40. Determine the complexity of the algorithm of Corollary 9.7.22

9.41. Fill in the missing details of the proof of Theorem 9.7.23. What is the complexity of this recursive algorithm?

9.42. Prove Lemma 9.7.9.

9.43. Prove the last Claim in the proof of Theorem 9.7.13. Hint: use the same approach as in the proof of Lemma 9.7.8.

9.44. **Fan-in, fan-out in eulerian directed multigraphs.** Let D be an eulerian directed multigraph and suppose D has arc-disjoint paths $P_1, P_2, \ldots, P_k$ such that P_i starts at x_i and ends at u for $i = 1, 2, \ldots, k$. Prove that D contains arc-disjoint paths $P'_1, P'_2, \ldots, P'_k$ such that P'_i is a (u, x_i)-path and P'_i is arc-disjoint from P'_j for all $1 \leq i, j \leq k$.

9.45. **(+) Arc-disjoint (x, y)-, (y, z)-paths in quasi-transitive digraphs.** Prove that the characterization in Theorem 9.7.21 can be extended to quasi-transitive digraphs.

9.46. Show that the 3-ST-problem for eulerian digraphs can be reduced in polynomial time to the problem of deciding the existence of arc-disjoint $[s_1, t_1]$-, $[s_2, t_2]$-paths in an eulerian digraph with specified vertices s_1, t_1, s_2, t_2. Hint: use Exercise 9.44.

9.47. Prove that the arc-version of Problem (P5) of Proposition 9.2.1 is $\mathcal{NP}$-complete.

9.48. Supply the missing details in the proof of Theorem 9.9.2.

9.49. Show how to reduce Problem 9.9.1 for the case $u \neq v$ to the case $u = v$.

9.50. (+) Prove that if any of the conditions (1)-(6) in Theorem 9.9.6 are satisfied, the T has no pair of disjoint branchings F_u^+, F_v^-.

9.51. **(+)** Extend Theorem 9.9.3 to the case when v is on some 2-cycle. Hint: how should the sets $E_{\mathcal{A}}, E_{\mathcal{B}}$ and the branchings described be modified?

9.52. Prove Theorem 9.9.10. Hint: use Theorem 9.9.3 and Exercise 9.51.

9.53. Apply Fulkerson's minimum cost branching algorithm to the digraph in Figure 9.21 to find a minimum cost out-branching from r.

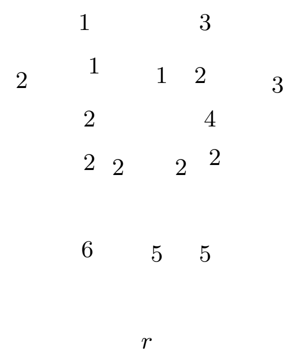


Figure 9.21 An instance of the minimum cost out-branching problem

9.54. **Finding an arborescence which covers a prescribed vertex set.** Show how to decide in polynomial time if a digraph $D = (V, A)$ has an arborescence with root s which contains all vertices of a prescribed subset $X \subseteq V$ (and possibly other vertices).

9.55. **Efficient implementation of independence oracles for the matroid intersection formulation of the minimum cost branching problem.** Show how to implement the necessary oracles for testing independence in the two matroids M_1, M_2 which were used in Subsection 9.10.1. Your algorithms should have complexity around $O(m)$, where m is the number of arcs in the directed multigraph.

9.56. **(+) Finding a minimum cost subdigraph which has k-arc-disjoint out-branchings rooted at s in a directed multigraph**. Show how to formulate this as a matroid intersection problem. Then sketch an algorithm to find the desired branchings. Hint: modify the matroids M_1, M_2 from Subsection 9.10.1.

9.57. **(+) Finding a minimum cost set of new arcs to add to a directed multigraph in order to ensure the existence of k-arc-disjoint out-branchings with a specified root**. Show how to solve this problem using an algorithm for weighted matroid intersection. Hint: use a similar approach as that in Exercise 9.56. Compare also with Exercise 8.65.

9.58. **Formulating the minimum spanning tree problem as a minimum cost branching problem.** Show that the minimum spanning tree problem (given a connected undirected graph with non-negative weights on the edges, find a spanning tree of minimum weight) can be formulated and solved as a minimum cost branching problem.

9.59. Suppose D is a digraph which has k but not $k+1$ arc-disjoint out-branchings rooted at s and let $\mathcal{F} = \{X \subset V - s : d^-_D(X) = k\}$. Explain how to find a minimal member of $\mathcal{F}$ (that is, no $Y \subset X$ belongs to $\mathcal{F}$). Hint: first show how to find a member X of $\mathcal{F}$ using flows and then show how to find a minimal member inside X. For the later, see the result of Exercise 3.35

9.60. **(+) Efficient implementation of the Frank-Fulkerson algorithm**. Try to determine how efficient the Frank-Fulkerson algorithm can be implemented. I.e. identify places in the algorithm where a seemingly time consuming step can be done efficiently.

9.61. **Reducing the shortest path problem to the minimum cost branching problem.** Show how to reduce the shortest (s,t)-path problem for digraphs with non-negative weights on the edges to the min-cost out-branching problem.

9.62. Use your reduction from the previous problem to devise an algorithm for the shortest (s,t)-path problem in a digraph with non-negative weights on the edges. That is, specialize the min-cost branching algorithm to the case where we only want to find a min-cost (s,t)-path. Hint: how many minimal sets are there to choose from in each step of the algorithm?

9.63. Compare your algorithm above to Dijkstra's algorithm (see Chapter 2) and other classical shortest path algorithms from Chapter 2.

9.64. **Simplicity preserving augmentations for rooted arc-connectivity.** Give an argument for the following claim. The Frank-Fulkerson algorithm can be used to find the cheapest set of new edges to add to a digraph to increase the maximum number of edge-disjoint out-branchings rooted at a fixed vertex from k to $k+1$, even when we are not allowed to add arcs that are parallel to already existing ones. Hint: what intersecting family $\mathcal{F}$ and what digraph with the new possible arcs should we consider?

9.65. **Increasing capacity of arcs to increase rooted arc-connectivity** Prove that the Frank-Fulkerson algorithm also works if all new arcs have to be parallel to existing ones.

9.66. Show that if $\mathcal{L}$ is a laminar family (i.e. $X, Y \in \mathcal{L}$ implies $X \cap Y = \emptyset$, or $X \subset Y$, or $Y \subset X$) on a ground set of size n, then the number of sets in $\mathcal{L}$ is at most $2n-1$. Then show that indeed there are digraphs for which the Frank-Fulkerson algorithm may be run (legally) so that it will find $2n-1$ sets before terminating phase 1.

9.67. **Comparing the Frank-Fulkerson algorithm with classical minimum spanning tree algorithms.** Suppose $D = (V, A)$ is a symmetric digraph (i.e. $xy \in E$ if and only if $yx \in E$) and that $c : A \rightarrow \mathcal{R}_+$ satisfies $c(xy) = c(yx)$. Compare the actions of the min-cost branching algorithm to well-known algorithms for finding a minimum spanning tree in a weighted (undirected) graph G. Such algorithms can be found in the book by Cormen, Leiserson and Rivest [169].

9.68. **A min-max formula for the minimum weight of new arcs to add to a digraph in order to increase the number of arc-disjoint out-branchings rooted at a fixed root by one**. Use the description and proof of correctness in Section 9.10 of the Frank-Fulkerson algorithm to derive a min-max formula for the minimum weight of such an augmenting set. Hint: the statement is similar to that of Theorem 9.10.2.

9.69. **(++)** Describe a polynomial algorithm for finding in a given digraph $D = (V, A)$ with specified vertices s, t, a minimum size subset $A' \subseteq A$ such that $D' = (V, A')$ has s, t in the same strong component (Natu and Fang [589]).

9.70. **(+) A min-max characterization of shortest paths.** Prove the following theorem due to Fulkerson:

Theorem 9.12.1 *[282] Let $D = (V, A)$ be a digraph that contains an (s, t)-path. Then the length of a shortest (s, t)-path in D equals the maximum number of arc-disjoint (s, t)-cuts.*

Extend this result to the weigthed case and give a characterization of the length of a shortest (s, t)-path in terms of (s, t)-cuts. Hint: reduce to a minimum cost branching problem and apply Theorem 9.10.2.

9.71. **The uncapacitated facility location problem**. This is the following problem. Given a set $L = \{l_1, l_2, \dots, l_p\}$ of possible locations of facilities (each of unbounded capacity) that shall serve a set $C = \{c_1, c_2, \dots, c_q\}$ of customers. There is a fixed cost w_i of locating a facility at location l_i and the cost of satisfying the demand of customer c_j from location l_i is given by d_{ij}. The problem is to decide which facilities to open and which facilities shall satisfy the demand of a customer such that the total costs are minimized. Show how to formulate this problem as an instance of the DSCC problem (see Subsection 9.10.3). Hint: since there is unbounded capacity at each facility, no client needs more than one facility to serve it. How can you model the cost of opening a facility by the cost of an arc?

9.72. Show that if the cost of opening a facility is zero, then there is a very simple greedy algorithm for solving the uncapacitated facility location problem (defined in Exercise 9.71).

9.73. Show how to derive Theorem 9.11.1 from Theorem 7.6.3.

10. Cycle Structure of Digraphs

In the previous chapters, especially in Chapters 5 and 6, we considered various properties of cycles in digraphs. The study of cycle structure of digraphs is one of the most important areas in the theory of digraphs, and since several very interesting topics in this area have remained uncovered in the previous chapters, we discuss these topics in this chapter. We will mostly consider (directed) cycles; in most cases the adjective 'directed' is omitted. Sometimes we will use oriented cycles, i.e. orientations of undirected cycles.

Section 10.1 is devoted to the cycle space of digraphs. We show how properties of the cycle space imply certain structural results on digraphs. In Section 10.2, we consider polynomial algorithms by Alon, Yuster and Zwick to find paths and cycles of length $\Theta(\log n)$ in a digraph of order n. In Section 10.3, we study how many vertex- or arc-disjoint cycles a digraph can have as well as the problems to find the minimum number of vertices or arcs to destroy all cycles in a graph. In Section 10.4 we will see that the maximum number of vertex-disjoint cycles in a digraph D is related to the minimum number of vertices in D needed to eliminate all cycles of D and the same is true for the corresponding arc version; Younger's conjecture formally states this mutual dependence. We give an overview of the proof of Younger's conjecture by Reed, Robertson, Seymour and Thomas.

The investigation of cycles of length equal k modulo p is started in Section 10.5, where we show that cycles of length 0 modulo p are very useful in the study of Markov chains. A number of remarkable results related to the even cycle problem in digraphs are given in Section 10.6; these include the theorem by McCuaig, Robertson, Seymour and Thomas that the even cycle problem is polynomial time solvable, and the theorem by Thomassen that every strong digraph with minimum out-degree and in-degree at least 3 contains an even cycle. In Section 10.6, we describe some extensions of these and other results to cycles of length equal k modulo p. A brief overview of results on short cycles in semicomplete multipartite digraphs can be found in Section 10.7. An interesting inequality between the length of a longest path and the length of a longest cycle in a strong semicomplete multipartite digraph, conjectured by Volkmann and proved by Gutin and Yeo, is shown in Section 10.8. Results on the well-known Caccetta-Häggkvist conjecture on the girth of a digraph, including the one by Chvátal and Szemerédi, are given in Section 10.9. Section

10.10 is devoted to a pair of additional topics: Ádám's conjecture on the number of cycles in a digraph, and Marcus' theorem on chords of cycles and its applications.

10.1 Vector Spaces of Digraphs

In this section we consider the cycle and cocycle spaces of a connected digraph; we will prove some basic properties of these vector spaces and an interesting result on 2-arc-coloured digraphs whose proof uses the notion and properties of the cycle space. We will use only the most basic notions and results on (general) vector spaces (see e.g., Morris [573]).

Let $D = (V, A)$ be a directed graph with arcs $e_1, \dots, e_m$. The **arc space** $\mathcal{A}(D)$ of D is the vector space over the 2-element field $GF(2) = \{0, 1\}$ of all functions $A \rightarrow GF(2)$. Every vector of $\mathcal{A}(D)$ corresponds naturally to a subset of A, the set of those arcs to which it assigns one. We may think of $\mathcal{A}(D)$ as the set of all subsets of A made into a vector space: the sum of $B, C \subseteq A$, denoted $B \triangle C$ is their symmetric difference, i.e., $B \triangle C = B \cup C - B \cap C$. Observe that the zero vector of $\mathcal{A}(D)$ is $\emptyset$; the sets $\{e_i\}$, $i = 1, 2, \dots, m$ are linearly independent and every vector of $\mathcal{A}(G)$ is the sum of the corresponding singletons, thus, $\{e_i\}$, $i = 1, 2, \dots, m$ form a basis of $\mathcal{A}(D)$ and $\dim \mathcal{A}(D) = m$.

Let B, C be a pair of vectors in $\mathcal{A}(D)$, where $B = \beta_1 e_1 \triangle \beta_2 e_2 \triangle \dots \triangle \beta_m e_m$ and $C = \gamma_1 e_1 \triangle \gamma_2 e_2 \triangle \dots \triangle \gamma_m e_m$, $\beta_i, \gamma_i \in \{0, 1\}$. We write the **scalar product**

$$\langle B, C \rangle := \sum_{i=1}^{m} \beta_i \gamma_i \qquad \pmod 2.$$

We say that B and C are **orthogonal** if $\langle B, C \rangle = 0$. Observe that B and C are orthogonal if and only if $|B \cap C|$ is even. For a pair of distinct subspaces $\mathcal{S}$ and $\mathcal{F}$ of $\mathcal{A}(D)$, we say that $\mathcal{S}$ and $\mathcal{F}$ are **orthogonal** if every vector of $\mathcal{S}$ is orthogonal to every vector of $\mathcal{F}$. It follows from well-known results in linear algebra that

$$\dim \mathcal{S} + \dim \mathcal{F} \leq m \tag{10.1}$$

for orthogonal subspaces $\mathcal{S}$ and $\mathcal{F}$.

In graph theory, some subspaces of $\mathcal{A}(D)$ are of special interest: the cycle space and the cocycle space. The **cycle space** of a digraph $D = (V, A)$ is a subspace of $\mathcal{A}(D)$ consisting of arc sets B such that the degrees of all vertices in the subdigraph $D\langle B \rangle$ are even[1]. The cycle space is indeed a subspace of $\mathcal{A}(D)$: the sum of two vectors in the cycle space as well as the product of a coefficient in $\{0, 1\}$ with a vector in the cycle space belongs to the cycle

[1] Recall that the degree of a vertex x in a digraph D is the sum $d_D^+(x) + d_D^-(x)$, i.e. the degree of x in $UG(D)$.

space. The cycle space is denoted by $\mathcal{C}(D)$; its name is justified by the fact that $\mathcal{C}(G)$ is generated by the oriented cycles of G. Indeed, if B is a vector in $\mathcal{C}(D)$, then $D\langle B\rangle$ contains an oriented cycle Z; $B - A(Z)$ is also a vector of $\mathcal{C}(D)$; it remains to apply the induction on the cardinality of B. Later (see Theorem 10.1.4) we will show that, if D is strong, then $\mathcal{C}(D)$ is generated, in fact, only by directed cycles of D.

For a connected digraph $D = (V, A)$, a set B of arcs is a **cocycle** if $D - B$ is not connected. The **cocycle space** $\mathcal{C}^*(D)$ consists of all cocycles of G and the empty set. We leave as Exercise 10.2 to prove the following proposition (see Bondy and Murty [127] and Diestel [191]):

Proposition 10.1.1 *For a connected digraph $D = (V, A)$, $\mathcal{C}^*(D)$ is a subspace of $\mathcal{A}(D)$. The cocycle space is generated by the cocycles of the form*

$$C(x) = \{xy \in A : y \neq x\} \cup \{zx \in A : z \neq x\}.$$

□

Using the facts that the cycle space of D is generated by oriented cycles of D and the cocycle space of D is generated by the cocycles $C(x)$ together with linearity of the scalar product, it is easy to prove the following proposition.

Proposition 10.1.2 *For a connected digraph D, the cycle space and the cocycle space are orthogonal.* □

Now we are ready to prove the following important theorem on the cycle space and the cocycle space.

Theorem 10.1.3 *For a connected directed graph D, we have $\dim \mathcal{C}(D) = m - n + 1$ and $\dim \mathcal{C}^*(D) = n - 1$.*

Proof: Let T be a spanning oriented tree of D. Recall that $|A(T)| = n - 1$. For an arc e in T, the set $C_e = A(D) - A(T) + e$ is a cocycle. Clearly, the cocycles C_e are linearly independent. Hence, $\dim \mathcal{C}^*(D) \geq n - 1$. If we add an arc e not in T to T, we obtain a digraph $T + e$ with a unique oriented cycle Z_e. Since the set of oriented cycles Z_e is a linearly independent set, we have $\dim \mathcal{C}(D) \geq m - n + 1$. Hence, $\dim \mathcal{C}(D) + \dim \mathcal{C}^*(D) \geq m$.

On the other hand, by Proposition 10.1.2 and Formula (10.1), we have $\dim \mathcal{C}(D) + \dim \mathcal{C}^*(D) \leq m$. Thus, $\dim \mathcal{C}(D) + \dim \mathcal{C}^*(D) = m$ and the formulae of this theorem are proved. □

Interestingly enough, for strong digraphs some bases consist entirely of directed cycles as can be seen from the following easy result:

Theorem 10.1.4 *For a strong digraph D, the cycle space is generated by (directed) cycles of D.*

Proof: By Theorem 7.2.2 and Corollary 7.2.3, D has an ear decomposition $P_1, P_2, \ldots, P_{m-n+1}$, where P_1 is a (directed) cycle, and every P_i, $i > 1$, is either a (directed) path which intersects with $\cup_{k=1}^{i-1} V(P_k)$ only at its end-vertices, or a (directed) cycle having only one vertex in common with $\cup_{k=1}^{i-1} V(P_k)$. Clearly, the subdigraph of D induced by $\cup_{k=1}^{i} V(P_k)$ has a cycle C_i containing P_i. Observe that the cycles $C_1, \ldots, C_{m-n+1}$ are linearly independent. Since $\dim \mathcal{C}(D) = m-n+1$, the cycles $C_1, \ldots, C_{m-n+1}$ form a basis of the cycle space. □

Recall that a transitive triple in a digraph D is a subdigraph of D, which is the non-strong tournament of order 3. For special classes of digraphs, one can find other bases. For example, Thomassen [709] proved the following two results:

Proposition 10.1.5 *If T is a tournament, then $\mathcal{C}(T)$ is generated by the transitive triples together with the hamiltonian (directed) cycles of T.*

Proof: Clearly, the cycle space of T is generated by oriented cycles of length 3 (one may apply the result of Exercise 10.1). If T is not strong, but has a directed 3-cycle, then consider such a 3-cycle $C = xyzx$. Clearly, there is a vertex v such that either v dominates each of x, y, z or is dominated by each of x, y, z. In either case, $C = A(T_{xy}) \triangle A(T_{yz}) \triangle A(T_{zx})$, where T_{uw} is the transitive triple containing v and the arc uw. Hence, $\mathcal{C}(T)$ is generated by the transitive triples.

Therefore, we may assume that T is strong. By Theorem 10.1.4 it suffices to prove that every (directed) cycle $C = x_1 x_2 \ldots x_{n-t} x_1$ of T is a sum of some transitive triples and hamiltonian cycles in T. We prove it by induction on $t = |V(T)| - |V(C)|$. If $t = 0$, then the claim is trivial. Suppose that $t > 0$ and $v \notin V(C)$. If $v \mapsto C$ or $C \mapsto v$, then as above we can see that C is a sum of transitive triples. Otherwise, without loss of generality, we may assume that $x_1 \rightarrow v \rightarrow x_2$. Hence, $A(C) = A(C') \triangle (A(C') \triangle A(C))$, where $C' = C[x_2, x_1] v x_2$. Note that $A(C') \triangle A(C)$ is a transitive triple. Our proposition now follows by induction. □

Theorem 10.1.6 *If T is a 4-strongly connected tournament of order n, then $\mathcal{C}(T)$ is generated by (directed) cycles of lengths n and $n-1$.*

Proof: Let arcs xy, yz, xz form a transitive triple R of T. By Exercise 6.16, the 3-strong tournament $T - y$ has a hamiltonian cycle H through xz. We have

$$A(R) = A(H) \triangle [A(H) \triangle A(R)].$$

Observe that the term in the brackets is the arc set of a hamiltonian cycle of T. Hence, every transitive triple is in the space generated by cycles of length $n-1$ and n and the theorem now follows from Proposition 10.1.5. □

We consider the following nice result, also due to Thomassen. This result is not directly on the cycle space but its proof exploits properties of the cycle space. First we need the definition of a monochromatic subdigraph of a 2-arc-coloured digraph. Let $D = (V, A)$ be a digraph and let $f : A \rightarrow \{1, 2\}$. A subdigraph $D' = D\langle A' \rangle$ of D is **monochromatic** if $f(a) = i$ for all $a \in A'$, where $i = 1$ or 2.

Theorem 10.1.7 *[709] Let D be a strong digraph whose underlying undirected graph is 2-connected. Let arcs of D be coloured into two colours 1 and 2 such that D has an arc of each colour. Then D has a non-monochromatic (directed) cycle.*

Theorem 10.1.7 follows from Theorem 10.1.4 and the next lemma.

Lemma 10.1.8 *[709] Let D be a 2-arc-coloured and non-monochromatic digraph such that $UG(D)$ is 2-connected. If a set of oriented cycles G_D generates $\mathcal{C}(D)$, then G_D has a non-monochromatic oriented cycle.*

Proof: Suppose that every cycle of G_D is monochromatic. We show that this leads to a contradiction. Let x be a vertex of D incident to two arcs, say xy and zx, of different colours. Since $UG(D)$ is 2-connected, $D - x$ has an oriented (y, z)-path P. Clearly, P together with xy and zx forms a non-monochromatic oriented cycle C of D. Since G_D generates $\mathcal{C}(D)$, we have

$$C = C_1 \triangle C_2 \triangle \ldots \triangle C_k,$$

where each C_i is in G_D. Without loss of generality we may assume that each C_i is monochromatic and only the oriented cycles $C_1, \ldots, C_p$, $0 < p < k$, are of colour 1. Hence the two sets of cycles $C_1, \ldots, C_p$ and $C_{p+1}, \ldots, C_k$ have no arc in common. Therefore the fact that $C = C_1 \triangle C_2 \triangle \ldots \triangle C_k$ implies that $Q = C_1 \triangle \ldots \triangle C_p$ must be a proper non-empty subdigraph of C. So, Q is a non-trivial collection of oriented paths and $Q \in \mathcal{C}(D)$, contradicting the fact that $\mathcal{C}(D)$ is a cycle space (some vertex in Q has odd degree). $\square$

Applying Theorem 10.1.7, one can easily conclude that the problem to verify whether a 2-arc-coloured digraph has a non-monochromatic directed cycle is polynomial time solvable. It is interesting to compare this result with Theorem 11.2.2 asserting that the problem to verify whether a 2-arc-coloured digraph has a directed cycle, which alternates in colour, is $\mathcal{NP}$-complete. One may speculate that being non-monochromatic is more vague and thus a weaker property than being alternating.

Several interesting results on tournaments whose proofs are based on the properties of the cycle space can be found in the paper [709] by Thomassen (see also [714]).

10.2 Polynomial Algorithms for Paths and Cycles

While it is $\mathcal{NP}$-complete to decide whether a digraph D_n of order n has a path or cycle with n vertices, it is not trivial to see for what functions $l_p(n)$ and

$l_c(n)$, one can verify in polynomial time whether D_n contains a path (cycle, respectively) of length $l_p(n)$ ($l_c(n)$, respectively). In particular, Papadimitriou and Yannakakis [141] conjectured that one can determine in polynomial time the existence of a path of length $p_l(n) = \Theta(\log n)$. Alon, Yuster and Zwick [16, 17] resolved this conjecture in affirmative. They also proved that one can check whether a digraph of order n has a $\vec{C}_k$ in polynomial time as long as $k = O(\log n)$. In this section we will briefly consider certain elegant ideas behind algorithms designed in [16, 17]. Further developments on the topic can be found in [18] and in the references therein. Various algorithmic aspects on enumeration of short cycles are also discussed there.

We start with a simple technical result on the expectation of a geometric random variable. This result can be found in many books on probability theory; we include its short proof for the sake of completeness. We use $\mathrm{Prob}(E)$ to denote the probability of the event E.

Lemma 10.2.1 *Let $0 < p \leq 1$ and let $x_1, x_2, \ldots$ be a sequence of random boolean variables such that $x_j = 1$ with probability p for each $j \geq 1$. A random variable ν is defined as follows: for $j \geq 1$, $\nu = j$ if and only if $x_j = 1$ and $x_1 = x_2 = \ldots = x_{j-1} = 0$. Then, the expectation of ν is $1/p$.*

Proof: The expectation of ν equals

$$\sum_{i=1}^{\infty} i\cdot\mathrm{Prob}(\nu = i) = \sum_{i=1}^{\infty}\mathrm{Prob}(\nu \geq i) = \sum_{i=1}^{\infty}(1-p)^{i-1} = 1/p.$$

□

To design algorithms verifying the existence of paths and cycles, Alon, Yuster and Zwick [16, 17] introduced two methods: the random acyclic subdigraph method and the colour-coding method. We consider first the random acyclic subdigraph method and then the method of colour-coding. In the rest of this section, we will follow [17].

Let $D = (V, A)$ be a digraph with $V = \{u_1, u_2, \ldots, u_n\}$. Let $M = [m_{ij}]$ be the adjacency matrix of D, i.e. $m_{ij} = 1$ if $u_i \rightarrow u_j$ and $m_{ij} = 0$, otherwise. It is well known (see Exercise 2.20) that the (i, j)th entry of the kth power of M is non-zero if and only if there is a (u_i, u_j)-walk of length k. However, many of (u_i, u_j)-walks of length k can be with repeated vertices (and even arcs). Thus, one naturally asks how we can get rid of walks that are not paths or cycles. One such method is **the random acyclic subdigraph method**: we choose randomly a permutation π on $\{1, 2, \ldots, n\}$ and construct the corresponding acyclic spanning subdigraph H of D by taking the following arcs: $u_{\pi(i)}u_{\pi(j)} \in A(H)$ if and only if $u_{\pi(i)}u_{\pi(j)} \in A$ and $\pi(i) < \pi(j)$. Clearly, every walk of H is a path in D (no vertices can be repeated as H is acyclic). On the other hand, every path P with k arcs in D has a $1/(k+1)!$ chance to be a path in H as well (Exercise 10.5).

Let $O(n^{\omega})$ be the complexity of boolean matrix multiplication (i.e. of the multiplication of two boolean $n \times n$ matrices). Due to Coppersmith and

Winograd [168], $\omega < 2.376$. Using random acyclic subdigraphs, one can prove the following:

Theorem 10.2.2 *[16, 17] Let $D = (V, A)$ be a digraph that contains a path (a cycle, respectively) of length k. A path (a cycle, respectively) of length k in D can be found in expected time $O((k+1)!{\cdot}m)$ ($O(k! \log k{\cdot}n^{\omega})$, respectively).*

Proof: To find a path of length k in D one can apply the following algorithm. Choose randomly a permutation π of $\{1, 2, \ldots, n\}$ and construct the corresponding acyclic spanning subdigraph H of D as described above. Using the $O(m)$-algorithm of Subsection 2.3.2, find a longest path P in H. If the length of P is less than k, then repeat the above procedure. Otherwise return a subpath of P whose length is k.

Since D contains $\vec{P}_{k+1}$, H has a path of length at least k with probability at least $1/(k+1)!$. Hence, by Lemma 10.2.1, the expected number of iterations in the above algorithm is at most $(k+1)!$. Thus, the expected running time is $O((k+1)!m)$ as claimed.

To find a cycle of length k in D one can apply the following algorithm. Choose randomly a permutation π on $\{1, 2, \ldots, n\}$ and construct the corresponding acyclic spanning subdigraph H of D as above. By computing (in time $O(\log k{\cdot}n^{\omega})$, see Exercise 2.21) the $(k-1)$th power of the adjacency matrix of H, we find all pairs of vertices which are end-vertices of $(k-1)$-paths in H (see Exercise 10.6). If the terminal vertex of one of the paths dominates the initial vertex of the path in D, we construct the corresponding k-cycle and stop. If no k-cycle is found we repeat the above procedure.

Clearly, the expected number of iterations in the above algorithm is at most $k!$. This implies the expected running time of $O(k! \log k{\cdot}n^{\omega})$. □

Now we turn our attention to a more powerful approach, the method of colour-coding. Let $c : V \rightarrow \{1, 2, \ldots, k\}$ be a colouring of the vertices of D. A path P in D is **colourful** if no pair of vertices of P are of the same colour.

Lemma 10.2.3 *Let $D = (V, A)$ be a digraph and let $c : V \rightarrow \{1, 2, \ldots, k+1\}$ be a colouring of the vertices of D. A colourful path of length k in D, if one exists, can be found in time $2^{O(k)}{\cdot}m$.*

Proof: Add to D a new vertex s of colour 0 that dominates all vertices of D and is dominated by no vertex. As a result, we obtain a digraph D', which has a $(k+1)$-path starting at s if and only if D contains a path of length k. To find a path of length $k+1$ in D' starting at s we use dynamic programming. Suppose that we have already found for each vertex $v \in V$ the possible sets of colours on colourful (s, v)-paths of length i as well as the corresponding paths (just one path for every possible set). We call such sets also **colourful**. Observe that for every v we have at most $\binom{k+1}{i}$ colourful sets and (s, v)-paths, respectively. We inspect every colourful set C that belongs to the collection of

v and every arc vu. Let $P(C)$ be the corresponding colourful path. If $c(u) \notin C$, then we add $C \cup c(u)$ ($P(C)u$, respectively) to the collection of colourful sets (paths, respectively) of u of cardinality (length, respectively) $i+1$. Clearly, D' contains a colourful $(k+1)$-path with respect to the colouring c if and only if the collection of colourful paths of length $k+1$ for some vertex is not empty. The number of operations of this algorithm is at most

$$O\left(\sum_{i=0}^{k+1} i\binom{k+1}{i}m\right) = O((k+1)2^{k+1}m).$$

$\square$

The next lemma follows from Lemma 10.2.3 and is left as Exercise 10.8.

Lemma 10.2.4 *Let $D=(V,A)$ be a digraph and let $c : V \rightarrow \{1,2,\ldots,k\}$ be a colouring of the vertices of D. For all ordered pairs x,y of distinct vertices colourful (x,y)-paths of length $k-1$ in D, if they exist, can be found in total time $2^{O(k)} \cdot nm$.* $\square$

Actually, for dense digraphs the complexity of this lemma can be improved to $2^{O(k)} \cdot n^{\omega}$ [17]. Clearly, Lemma 10.2.4 implies an $2^{O(k)} \cdot nm$-algorithm to find a k-cycle in D.

If P is a path of order k in D whose vertices are randomly coloured from set of k colours, then P has a chance of $k!/k^k > e^{-k}$ to become colourful. Thus, by Lemma 10.2.1, the expected number of times to randomly generate k-colouring to detect P is at most $\lfloor e^k \rfloor$. This fact and Lemmas 10.2.3 and 10.2.4 imply the following:

Theorem 10.2.5 (Alon, Yuster and Zwick) *[16, 17] If a digraph D has a path of length k (k-cycle, respectively), then a path of length k (k-cycle, respectively) can be found in $2^{O(k)} \cdot m$ ($2^{O(k)} \cdot nm$, respectively) expected time.*

The algorithms mentioned in this theorem are quite simple, but unfortunately not deterministic. Fortunately, one can derandomize these algorithms to obtain deterministic algorithms with time complexity still linear in m. Observe that for a path P of order k in $D=(V,A)$ many k-colourings of V are equally good or bad depending on P being colourful or not. This means that we do not need to consider all k^n k-colourings of V to detect a path of order k in D; a subset S of colourings such that every path of order k is colourful for at least one colouring of S is sufficient. In other words, we wish that for every k-set W of vertices there is a colouring from S that assigns vertices of W different colours.

This is captured in the notion of a *k-perfect family of hash functions* from $\{1,2,\ldots,n\}$ to $\{1,2,\ldots,k\}$. Schmidt and Siegel [653] following Fredman, Komlós and Szemerédi [277] gave an explicit construction of a k-perfect family from $\{1,2,\ldots,n\}$ to $\{1,2,\ldots,k\}$ in which each function is specified by $b =$

$O(k) + 2\log_2 \log_2 n$ bits. Thus, the size of the family is $2^b = 2^{O(k)} \log_2^2 n$. The value of each of these functions on each specified element of $\{1, 2, \dots, n\}$ can be computed in $O(1)$ time. Using this family, the algorithms of Theorem 10.2.5 can be derandomized to obtain deterministic algorithms running in time $O(2^{O(k)} m \log^2 n)$ and $O(2^{O(k)} mn \log^2 n)$, respectively. Alon, Yuster and Zwick [16, 17] pointed out how to decrease each of the above complexities by the multiplicative factor of $\log n$. They also showed how to derandomize some versions of algorithms mentioned in Theorem 10.2.2.

10.3 Disjoint Cycles and Feedback Sets

In this section we discuss several interesting non-trivial results on vertex-disjoint and arc-disjoint cycles. Actually, these results deal with some aspects of the following problem: given a digraph D, find the maximum number of vertex-disjoint (arc-disjoint) cycles in D. This problem itself is $\mathcal{NP}$-hard in both vertex and arc versions (see below). However, some very interesting sufficient conditions have been obtained for the existence of a large number of vertex-disjoint (arc-disjoint) cycles.

We will use some additional notation and terminology. For a digraph D, the maximum number of vertex-disjoint (arc-disjoint) cycles is denoted by $\nu_0(D)$ ($\nu_1(D)$). In a digraph D, a set S of vertices (arcs) is a **feedback vertex set** (an **feedback arc set**) if $D - S$ is acyclic. The minimum number of elements in a feedback vertex (arc) set of D is denoted by $\tau_0(D)$ ($\tau_1(D)$). Notice that the parameters $\tau_0(D)$ and $\tau_1(D)$ have several practical applications, one of the most important is testing electronic circuits (see Leiserson and Saxe [512]). An electronic circuit can be modeled by a directed graph by letting each (boolean) gate correspond to a vertex and the wires into each gate be modeled by arcs into the vertex corresponding to that gate. Finding a small set of arcs whose removal makes the resulting digraph acyclic can help reduce the hardware overhead needed for testing the circuit using so-called scan registers (see Kunzmann and Wunderlich [506]).

10.3.1 Complexity of the Disjoint Cycle and Feedback Set Problems

We start from the following simple, but quite useful result.

Proposition 10.3.1 *For every digraph D there exist digraphs D' and D'' such that $\nu_0(D) = \nu_1(D')$, $\tau_0(D) = \tau_1(D')$, $\nu_1(D) = \nu_0(D'')$ and $\tau_1(D) = \tau_0(D'')$. The digraphs D' and D'' can be constructed from D in polynomial time.*

Proof: The digraph D'' can be defined as $D'' := L(D)$. To construct D' simply apply the vertex splitting procedure (see Subsection 3.2.4) to all vertices of D. The reader is advised to verify that the equalities of this proposition indeed hold. □

This proposition implies that the following problems are of the same complexity (up to a polynomial factor).

The FVS problem: Given an integer k and a digraph D, verify whether $\tau_0(D) \leq k$.
The FAS problem: Given an integer k and a digraph D, verify whether $\tau_1(D) \leq k$.

Similarly the problem of deciding whether $\nu_0(D) \geq k$ is polynomially equivalent to the problem of deciding whether $\nu_1(D) \geq k$.

Karp [474] was the first to prove the following theorem:

Theorem 10.3.2 *The FAS problem is $\mathcal{NP}$-complete.* □

Gavril [306] proved that the FAS problem remains $\mathcal{NP}$-complete even for digraphs D with $\delta^0(D) \leq 3$ or line digraphs.

Proposition 10.3.1 and Theorem 10.3.2 imply immediately that the FVS problem is $\mathcal{NP}$-complete. Using Theorem 12.6.1 due to Bang-Jensen and Thomassen, we obtain the following stronger result:

Theorem 10.3.3 *[89] The FVS problem is $\mathcal{NP}$-complete even for tournaments.* □

The FVS problem remains $\mathcal{NP}$-complete for digraphs D with $\delta^0(D) \leq 2$, planar digraphs D with $\delta^0(D) \leq 3$ (see Garey and Johnson [303]) and for line digraphs (see Gavril [306]). This problem, unlike the FAS problem, is $\mathcal{NP}$-complete even for undirected graphs [303].

It is not surprising that the above mentioned decision problems for the parameters ν_0 and ν_1 are also $\mathcal{NP}$-complete.

Theorem 10.3.4 *Given a digraph D and an integer k, it is $\mathcal{NP}$-complete to decide whether $\nu_0(D) \geq k$ ($\nu_1(D) \geq k$).*

Proof: By Proposition 10.3.1 it is sufficient to show this claim only for ν_0. A scheme of the proof of the assertion for ν_0 is given in Exercise 10.9. □

10.3.2 Disjoint Cycles in Digraphs with Minimum Out-Degree at Least *k*

It turns out that one of the sufficient conditions to guarantee the existence of a large number of vertex-disjoint cycles in a digraph D is that $\delta^+(D)$ is large enough. Let $f(k)$ be the least integer such that every digraph of minimum out-degree at least $f(k)$ contains k vertex-disjoint cycles. The very existence of $f(k)$ for every $k \geq 1$ is not obvious. Thomassen [700] was the first to show this fact. He proved that $f(k) \leq (k+1)!$. Bermond and Thomassen [115] conjectured that, in fact, $f(k) = 2k-1$.This holds for $k=1$ as every acyclic digraph has a vertex of out-degree zero. This holds also for $k=2$ (see Exercises 10.21 and 10.22). Alon [10] was the first to prove that the function $f(k)$ is linear. He obtained the following result.

Theorem 10.3.5 *There exists an absolute constant C so that $f(k) \leq Ck$ for all k. In particular, $C = 64$ will do.* □

We will not give a proof of Theorem 10.3.5 as it is somewhat tedious. However, we will prove a slightly weaker result, Theorem 10.3.8. This proof shows basic ideas involved in the proof of Theorem 10.3.5 in [10]. We leave as Exercise 10.23 the proof of the following corollary.

Corollary 10.3.6 *[10] Every digraph with minimum out-degree k has at least $k^2/128$ arc-disjoint cycles.* □

For k-regular digraphs, the result of this corollary seems far from being sharp. Alon, McDiarmid and Molloy [13] conjectured the following:

Conjecture 10.3.7 *Every k-regular digraph contains $\binom{k+1}{2}$ arc-disjoint cycles.*

This conjecture was verified for $k \leq 3$ in [13]. Now we formulate Theorem 10.3.8.

Theorem 10.3.8 *[10] For k large enough, $f(k) \leq (3+o(1))k\log_e k$.*

For technical reasons, we prove this theorem not only for digraphs, but for directed pseudographs without parallel arcs. However, for shortness we will still use the term 'digraphs' in the rest of this subsection for digraphs with possible loops.

Clearly, Theorem 10.3.8 holds for $k=1$. Assume that Theorem 10.3.8 is true for all values up to some k and $k+1$ is the minimum integer violating the inequality. Then, $f(k+1) > f(k)+4$. Let $D=(V,A)$ be a digraph of minimum out-degree r, $r = f(k+1)-1$, such that D does not have $k+1$ vertex-disjoint cycles. We also assume that D has the minimum possible number of vertices and, subject to this property, the minimum size. By the definition of D, the out-degree of every vertex of D is exactly r and $\delta^-(D) > 0$. Moreover, D has no loop, since otherwise the digraph obtained from D by deleting a vertex with a loop cannot contain k vertex-disjoint cycles, showing that $f(k+1)-2 = r-1 \leq f(k)-1$, which is impossible as we saw above that $f(k+1) > f(k)+4$.

We proceed by proving certain properties of D formulated as lemmas. The proof of Lemma 10.3.11 exploits a probabilistic argument. The first lemma is due to Thomassen [700] and the next two to Alon [10].

Lemma 10.3.9 *[700] For every $v \in V$, the subdigraph $D\langle N^-(v)\rangle$ contains a cycle.*

Proof: Fix an arbitrary vertex $v \in V$. Put $H = D\langle N^-(v)\rangle$. It suffices to show that $\delta^-(H) > 0$. Assume that $u \in V(H)$ and $d^-_H(u) = 0$. Then, there is no vertex in D that dominates both u and v. This implies that the digraph D', obtained from D by first deleting the arcs with tail u except for uv and then contracting uv, has minimum out-degree r. (Notice that D' may have a loop.) By the minimality of D, the digraph D' has $k+1$ vertex-disjoint cycles. These cycles can easily be transformed into vertex-disjoint cycles of D, a contradiction. □

Lemma 10.3.10 *[10] We have $|V| \leq k(r^2 - r + 1)$.*

Proof: Put $n = |V|$ and let G be the undirected graph with vertex set V in which a pair u and v of distinct vertices is adjacent if and only if there is a vertex in D that dominates both. Define $m = n\binom{r}{2}$ and observe that the size of G is at most m (since every vertex of D has out-degree r). Therefore, as it is well known (see, e.g., Berge [105, page 282]) G has an independent set of cardinality at least $\frac{n^2}{2m+n}$. If this number is at least $k+1$, then there is a set $x_1, \ldots, x_{k+1}$ of independent vertices of G. This means that the sets $N^-(x_1)$,...,$N^-(x_{k+1})$ are pairwise disjoint. It now follows from Lemma 10.3.9 that D has $k+1$ vertex-disjoint cycles, a contradiction. Hence, $\frac{n^2}{2m+n} \leq k$. This implies the inequality of Lemma 10.3.10. □

Lemma 10.3.11 *[10] We have $k(r^2 - r + 2)(1 - \frac{1}{k+1})^r \geq 1$.*

Proof: Assume that the inequality of this lemma is false and

$$k(r^2 - r + 2)(1 - \frac{1}{k+1})^r < 1.$$

Assign independently to every vertex $v \in V$ a colour $i \in \{1, 2, \ldots, k+1\}$ with probability $p = \frac{1}{k+1}$. Let V_i be the set of vertices coloured i. For each vertex $v \in V$, let E_v denote the event that all out-neighbours of v are of colours different than that of v. Since every vertex of D has out-degree r we have $\mathrm{Prob}(E_v) = (1-p)^r$. For $i = 1, 2, \ldots, k+1$, let F_i denote the event that $V_i = \emptyset$. Then $\mathrm{Prob}(F_i) = (1-p)^n \leq (1-p)^{r+1}$. Hence, by Lemma 10.3.10,

$$\begin{aligned}\sum_{v \in V} \mathrm{Prob}(E_v) + \sum_{i=1}^{k+1} \mathrm{Prob}(F_i) &\leq n(1-p)^r + (k+1)(1-p)^{r+1} \\ &\leq k(r^2 - r + 1)(1-p)^r + k(1-p)^r \\ &= k(r^2 - r + 2)(1-p)^r \\ &< 1.\end{aligned}$$

This implies that with positive probability each $D\langle V_i\rangle$ is non-empty and has a positive minimum out-degree, and hence possesses a cycle. Thus, there is a choice of $V_1, \ldots, V_{k+1}$ giving $k+1$ disjoint cycles in D, a contradiction. □

Conclusion of the proof of Theorem 10.3.8: Lemma 10.3.11 implies that

$$k(r^2 - r + 2) \geq e^{r/(k+1)}.$$

Hence, for k large enough, $f(k) \leq f(k+1) - 1 = r \leq (3 + o(1))k \log_e k$. Thus, Theorem 10.3.8 is proved. □

10.3.3 Feedback Sets and Linear Orderings in Digraphs

We mentioned above that in many applications one wishes to find a minimum (cardinality) feedback arc set. Observe that, if A' is an arbitrary feedback arc set, then by definition $D - A'$ is acyclic and hence has an acyclic ordering $v_1, v_2, \ldots, v_n$. With respect to this ordering every arc $v_iv_j \in A - A'$ satisfies $i < j$. Hence, from the algorithmic point of view, finding a minimum feedback arc set in D is equivalent to finding an ordering $u_1, u_2, \ldots, u_n$ of V which maximizes (minimizes) the number of forward arcs (backward arcs); an arc u_iu_j is **forward** with respect to the above ordering if $i < j$, otherwise u_iu_j is **backward**[2]. This again is easily seen to be (algorithmically) equivalent to finding an acyclic subdigraph with the maximum number of arcs in D (Exercise 10.14). The latter problem is known as the **acyclic subdigraph problem**.

To illustrate the definitions above and to gain some understanding of difficulties in studying the problems above, let us consider the class of tournaments.

For a tournament T, let $\gamma(T)$ be the size of an acyclic subdigraph of T of maximum size. Fixing an arbitrary ordering $u_1, \ldots, u_n$ of vertices in T, we see that the number of forward arcs plus the number of backward arcs equals $\binom{n}{2}$. By replacing the ordering $u_1, u_2, \ldots, u_n$ by $u_n, u_{n-1}, \ldots, u_1$ if needed, we obtain that $\gamma(T) \geq n(n-1)/4$. One may guess that we can always find an acyclic subdigraph of T of size exceeding $n(n-1)/4$ by a significant number, say, $\epsilon n(n-1)/4$, where ϵ is an absolute positive constant not depending on n. However, this is not true due the following:

Theorem 10.3.12 *For every $n \geq 3$, there exists a tournament T of order n such that $\gamma(T) \leq n(n-1)/4 + \sqrt{n^3 \log_e n}/2$.*

Proof: Consider a random tournament T_n on vertices $1, 2, \ldots, n$, i.e., a tournament chosen randomly from the set of all tournaments on $1, 2, \ldots, n$. Observe that for every pair $i \neq j \in \{1, 2, \ldots, n\}$, $ij \in A(T_n)$ with probability $1/2$.

[2] Clearly, the set of backward arcs form a feedback arc set.

For every pair $i < j \in \{1, 2, \ldots, n\}$, define the random variable $x_{i,j}$ by

$$x_{i,j} := \begin{cases} +1 \text{ if } ij \in A(T_n) \\ -1 \text{ otherwise.} \end{cases}$$

Let $N = \binom{n}{2}$. With respect to the ordering $\pi = 1, 2, \ldots, n$, the number of forward arcs minus the number of backward arcs equals

$$\sum_{1 \le i < j \le n} x_{i,j} =: S_N.$$

Then, $E_\pi := \{|S_N| > a\}$ denotes the event that, in one of the two orderings $\pi = \pi(1), \pi(2), \ldots, \pi(n)(= 1, 2, \ldots, n)$ and $\pi^* = \pi(n), \pi(n-1), \ldots, \pi(1)(= n, n-1, \ldots, 1)$, the number of forward arcs exceeds $n(n-1)/4 + a/2$. On the other hand, S_N is the sum of $\binom{n}{2}$ random independent variables taking values $+1$ and -1, each with probability $1/2$. By Corollary A.2 in [14],

$$\text{Prob}(|S_N| > a) \le 2e^{-a^2/(2N)}, \tag{10.2}$$

for every positive number a.

Observe that the event E that for at least one permutation of $1, 2, \ldots, n$, the number of forward arcs exceeds $n(n-1)/4 + a/2$ equals the union of the events E_ν for all permutations of $1, 2, \ldots, n$, whose total number is $n!$. Put $a = \sqrt{n^3 \log_e n}$. Applying (10.2) we obtain

$$\begin{aligned} \text{Prob}(E) &\le 2n! \exp(-n \log_e n) \\ &\le 2n! n^{-n} \\ &< 1 \end{aligned}$$

for every $n \ge 3$. This means that with positive probability the event E does not hold, i.e. for every permutation of $1, 2, \ldots, n$, the number of forward arcs does not exceed $n(n-1)/4 + \sqrt{n^3 \log_e n}/2$. By the definition of T_n, it follows that there exists a tournament of order n with the above-mentioned property. $\square$

A slightly better result was obtained by de la Vega in [186] who proved that $\sqrt{\log_e n}$ in the inequality of Proposition 10.3.12 can be replaced by a constant.

One may also consider weighted versions of the problems above. Each arc is assigned a non-negative real valued weight and the goal is to find a feedback arc set of minimum total weight (respectively, an acyclic subdigraph of maximum weight). The weighted version of the acyclic subdigraph problem is known as **the linear ordering problem**. It arises naturally in the study of interactions between various sectors of an economical system (see Reinelt [631] and also Funke and Reinelt [284] and Grötschel and Jünger [337]).

For the linear ordering problem there is a very easy way to obtain an ordering which achieves at least half of the optimum value of an ordering. The proof of the following proposition is an easy exercise (Exercise 10.15).

Proposition 10.3.13 *Given any weighted digraph $D = (V, A, w)$, in time $O(m)$ one can find an acyclic subdigraph $D' = (V, A')$ of D such that $w(A') \geq w(A)/2$.* □

This proposition implies that there exists a polynomial 2-approximation[3] algorithm for the linear ordering problem, since $w(A)/2 \leq w(A') \leq w(A_o) \leq w(A)$, where $w(A_o)$ is the optimum weight.

Note that although the linear ordering problem and the feedback arc set problem are equivalent problems from the algorithmic point of view, the approximation algorithm above cannot be used as a 2-approximation algorithm for the feedback arc set problem as well. The reason is that the optimal ordering may have all or almost all arcs in the right direction (implying the number τ_1 is close to zero) whereas the ordering above may still have as little as half the arcs in the right direction. In fact, approximating the number τ_1 seems to be very difficult and so far no c-approximation algorithm is known for any constant c. The following best known approximation guarantee for the feedback arc set problem is due to Seymour [665].

Theorem 10.3.14 *There exists an $O(\log n \log\log n)$ approximation algorithm for the feedback arc set problem.* □

For a detailed account on approximating the number τ_1 of a directed multigraph we refer to the chapter [671] by Shmoys. Another approximation algorithm for a generalization of the feedback arc set problem (as well as the feedback vertex set problem) is described by Even, Naor, Schieber and Sudan [227].

While for arbitrary digraphs the feedback arc set problem is $\mathcal{NP}$-hard (see Theorem 10.3.2), for planar digraphs the situation is quite different (unless $\mathcal{P} = \mathcal{NP}$) due to the following result by Lucchesi:

Theorem 10.3.15 *[526] The feedback arc set problem is polynomially solvable for planar digraphs.*

We give a proof of Theorem 10.3.15 below. First we need the definition of the dual of a plane directed multigraph. Let $G = (V, E)$ be a planar pseudograph and let F be the set of faces of G (with respect to the fixed planar embedding of G). Let G^* be the pseudograph which has a vertex v_i for each face $f_i \in F$ and for every edge $e \in E$ such that e is on the boundary of faces f_i, f_j, the two vertices v_i, v_j corresponding to f_i, f_j are joined by an edge[4]. In general G^* contains parallel edges and may also contain loops. For

[3] For a minimization problem $\mathcal{M}$, an algorithm $\mathcal{A}$ is an $f(n)$-approximation algorithm if, for every instance of $\mathcal{M}$ of size n, $\mathcal{A}$ finds a solution whose value ρ satisfies $\frac{\rho}{\rho^*} \leq f(n)$, where ρ^* is the optimum value.

[4] Note that, if e is not part of the boundary of a facialcycle, then $f_i = f_j$ and we get a loop at v_i.

plane directed pseudographs we can also define a dual called the **directed dual**. This is the same as above but now the orientation of the arc between v_i and v_j is always chosen such that the arc crosses the original arc e from left to right (here left means the left side when we traverse e from its tail to its head). See Figure 10.1 for an example of the dual of a directed multigraph.

(a) (b)

Figure 10.1 (a) A plane directed multigraph D; (b) the directed dual D^* of D drawn on top of D. White circles indicate the vertices of D^* and thin arcs are arcs of D^*. Fat arcs indicate arcs of D.

If $D = (V, A)$ is a plane directed multigraph and D^* is its directed dual, then it is easy to see that D^* is also planar (Exercise 10.10). In fact, we have that $(D^*)^*$ is isomorphic to the converse of D (Exercise 10.11).

Proof of Theorem 10.3.15: Let D be a planar directed multigraph and assume that D is embedded in the plane with directed dual D^*. Clearly we may assume that $UG(D)$ is connected since otherwise we just consider each connected component separately.

We prove that the size of a minimum feedback set of D is equal to the minimum size of a dijoin of D (see the definition of a dijoin in Section 7.15). Recall from Section 7.15 that this is also the minimum number of arcs whose contraction results in a strongly connected directed multigraph.

If we delete an arc a of D the effect on the dual will be the same as if we contract the corresponding dual arc a^* (the one crossing a from left to right). If C is a facial cycle of D, then the vertex v corresponding to C has all arcs directed into it or out of it (depending on whether the orientation of C is clockwise or anti-clockwise). Thus in D^* the arcs incident with v form a directed cut (recall the definition of a directed cut from Section 7.15) in D^* implying that D^* is not strong.

Conversely, if D^* is not strongly connected then let H be an initial strong component (that is, there is no arc from $V - V(H)$ to $V(H)$ in D) of D^*. Now it is not difficult to see that the arcs of D corresponding to the directed cut $(V(H), V-V(H))$ in D^* (which is non-empty since D is connected) form a directed cycle (Exercise 10.12). Thus we have shown that D has a directed cycle if and only if D^* is not strongly connected. Furthermore, deleting arcs of D until we obtain an acyclic directed multigraph is equivalent to contracting arcs of D^* until we obtain a strong directed multigraph. This shows that the size of a minimum feedback arc set of D equals the size of a minimum directed join in D^*. Now it follows from Corollary 8.8.10 that we can find the feedback number (and a minimum feedback arc set) of D in polynomial time. □

Our arguments above imply the following:

Corollary 10.3.16 *For a planar digraph D, $\nu_1(D) = \tau_1(D)$.* □

10.4 Disjoint Cycles Versus Feedback Sets

In this section, we study relations between the parameters ν_0 and ν_1, on one hand, and parameters τ_0 and τ_1 on the other hand. We state the famous Younger's conjecture and present an overview of the proof of this conjecture by Reed, Robertson, Seymour and Thomas. Some (still) open conjectures and problems are mentioned as well.

10.4.1 Relations Between Parameters ν_i and τ_i

Clearly, for every digraph D, $\nu_0(D) \leq \nu_1(D)$ and it is easy to find an infinite family of digraphs D for which the two parameters are not equal. The same is true for the parameters τ_0, τ_1. Furthermore, we obviously have $\nu_i(D) \leq \tau_i(D)$ for $i = 1, 2$. It is easy to construct an infinite family of digraphs D such that $\nu_0(D) < \tau_0(D)$ (Exercise 10.19) and thus, by Proposition 10.3.1, an infinite family of digraphs D such that $\nu_1(D) < \tau_1(D)$.

On the other hand, there are families of digraphs for which the last two inequalities become equalities. Szwarcfiter [686] described a family of digraphs, D for which $\nu_0(D) = \tau_0(D)$. His family generalizes two families introduced by Frank and Gyárfás [267] and by Wang, Floyd and Soffa [732]. Szwarcfiter [686] also provides polynomial algorithms to recognize his family of digraphs and to find k-cycle factors and feedback vertex sets of cardinality k, where $k = \nu_0(D) = \tau_0(D)$. We have already seen that planar digraphs D satisfy $\nu_1(D) = \tau_1(D)$. Seymour [666] showed that the same result holds for a special family of eulerian digraphs. Another class of digraphs with the same property was considered by Ramachandran [620].

Even though not always $\nu_i(D) = \tau_i(D)$, $i = 0, 1$, in which case $\tau_i(D)$ exceeds $\nu_i(D)$, Younger [750] conjectured that the former is bounded by a function of the latter[5]. In other words, he conjectured that for every k, there

[5] The existence of $t_0(2)$ was conjectured earlier by Gallai, see [626].

exists a (least) natural number $t_0(k)$ ($t_1(k)$, respectively) such that for every digraph D the following holds: either D contains k vertex-disjoint (arc-disjoint, respectively) cycles or D has a feedback vertex (arc, respectively) set of cardinality at most $t_0(k)$ ($t_1(k)$, respectively). By Proposition 10.3.1, the validity of the 'vertex' version of Younger's conjecture implies that the 'arc' version holds and vice versa. Moreover, Proposition 10.3.1 implies that, if the functions $t_0(k)$ and $t_1(k)$ exist, then they are equal (Exercise 10.20). Younger's conjecture was completely settled recently by Reed, Robertson, Seymour and Thomas [626]. We discuss their solution in the next subsection. In the rest of this subsection we consider the parameters ν_1 and τ_1 for the class of tournaments.

Even for a tournament T, the parameters $\nu_1(T)$ and $\tau_1(T)$ do not always coincide. By the proof of Theorem 10.3.12, for every $n \geq 3$ a random tournament T_n with n vertices, with probability tending to 1 as $n \rightarrow \infty$, has at least $n(n-1)/4 - \sqrt{n^3 \log_e n}/2$ arcs in a feedback arc set of T. On the other hand, it follows from a result by Chartrand, Geller and Hedetniemi [144] that T_n has at most $\lfloor \frac{n}{3} \lfloor \frac{n-1}{2} \rfloor \rfloor \leq \frac{1}{3}\binom{n}{2}$ arc-disjoint cycles (each cycle has at least three arcs). Isaak conjectured the following:

Conjecture 10.4.1 *[446] If T is a tournament which has a minimum feedback arc set A such that $T\langle A \rangle$ is a transitive subtournament of T, then $\nu_1(T)$ and $\tau_1(T)$ coincide.*

In [446] Isaak posed the following problem. Note that, if the answer to the problem is yes, then this implies Conjecture 10.4.1.

Problem 10.4.2 *Suppose T is a tournament having a minimum feedback arc set which induces an acyclic digraph with a hamiltonian path. Is it true that the maximum number of arc-disjoint cycles in T equals the cardinality of a minimum feedback arc set of T?*

It is easy to see that a minimum feedback arc set of a given digraph must induce an acyclic subdigraph of D (Exercise 10.16). The next result by Barthélémy, Hudry, Isaak, Roberts and Tesman implies that every acyclic digraph arises as a minimum feedback arc set of some tournament.

Theorem 10.4.3 *[95] Let D be an acyclic digraph. Then there exists a tournament T containing D as a subdigraph such that the arcs of D form a minimum feedback arc set in T.* □

The following conjecture is due to Bang-Jensen and Thomassen.

Conjecture 10.4.4 *[89] The feedback arc set problem is $\mathcal{NP}$-hard for tournaments.*

We point out that the feedback vertex set problem is $\mathcal{NP}$-hard for tournaments by Theorem 10.3.3.

10.4.2 Solution of Younger's Conjecture

The vertex and arc versions of Younger's conjecture were proved for various families of digraphs including the families mentioned above. McCuaig [559] proved the existence of $t_0(2)$ by characterizing **intercyclic digraphs**, i.e., digraphs D for which $\nu_0(D) \leq 1$. Moreover, he established that $t_0(2) = 3$. Reed and Shepherd [627] proved the vertex version of Younger's conjecture for planar digraphs using a result of Seymour [665]. The result of Reed and Shepherd combined with a result of Goemans and Williamson [323] implies that $t_0^{pd}(c) = O(c)$, where $t_0^{pd}(c)$ is the function $t_0(c)$ restricted to planar digraphs. Finally, Younger's conjecture was completely settled by Reed, Robertson, Seymour and Thomas [626]. In this subsection, we give a scheme of their proof. In particular, we provide a complete proof of perhaps the most interesting lemma in [626].

One of the important tools in the proof in [626] is the following well-known Ramsey theorem [621].

Theorem 10.4.5 (Ramsey) *For all integers $q, l, r \geq 1$ there exists a (minimum) integer $R_l(r, q) \geq 0$ so that the following holds. Let Z be a set of cardinality at least $R_l(r, q)$ and let every l-subset of Z be assigned a colour from $\{1, \dots, q\}$. Then there exist an r-subset S of Z and a colour $k \in \{1, \dots, q\}$ so that every l-subset of S is of colour k.* □

Some readers may be more familiar with the graph-theoretic special case of this theorem. For every pair of natural numbers q, r there exists an integer $R_2(r, q) \geq 0$ so that every q-edge-coloured complete graph of order at least $R_2(r, q)$ has a monochromatic complete subgraph of order r.

We start describing the scheme of the proof of Younger's conjecture by the following lemma whose proof is left as Exercise 10.24.

Lemma 10.4.6 *[626] Let $c \geq 1$ be an integer such that $t_0(c-1)$ exists. Let D be a digraph with $\nu_0(D) < c$ and let T be a feedback vertex set of D of cardinality $\tau_0(D)$. Suppose U, W are disjoint subsets of T both of cardinality r, where $r \geq 2t_0(c-1)$. Then there is an r-path subdigraph of D from U to W, which contains no vertex in $T - (U \cup W)$.* □

Let $\mathcal{L} = P_1 \cup \ldots \cup P_k$ be a k-path subdigraph in a digraph D and let u_i (w_i) be the initial (terminal) vertex in P_i, $i = 1, \dots, 2$. We say that $\mathcal{L}$ **links** $\boldsymbol{(u_1, \dots, u_k)}$ **to** $\boldsymbol{(w_1, \dots, w_k)}$ and $\mathcal{L}$ **is from** $\boldsymbol{\{u_1, \dots, u_k\}}$ **to** $\boldsymbol{\{w_1, \dots, w_k\}}$.

The following lemma was proved by the authors of [626] in joint work with Alon. Its proof uses Ramsey's theorem as well as Theorem 5.2.3 of Erdős and Szekeres.

Lemma 10.4.7 *Let $c \geq 2$ be an integer such that $t_0(c-1)$ exists, and let $k \geq 1$ be an integer. Then there exists an integer $t \geq 0$ (depending on k) so that the following holds. If D is a digraph with $\nu_0(D) < c$ and $\tau_0(D) \geq t$, then there are distinct vertices $u_1, \ldots, u_k, w_1, \ldots, w_k$ of D and a pair of k-path subdigraphs $\mathcal{L}_1, \mathcal{L}_2$ of D so that*

(i) $\mathcal{L}_1$ links $(u_1, \ldots, u_k)$ to $(w_1, \ldots, w_k)$,
(ii) $\mathcal{L}_2$ links $(w_1, \ldots, w_k)$ to either $(u_1, \ldots, u_k)$ or $(u_k, \ldots, u_1)$,
(iii) every (directed) cycle of $\mathcal{L}_1 \cup \mathcal{L}_2$ meets $\{u_1, \ldots, u_k, w_1, \ldots, w_k\}$.

Proof: Let $l := (k-1)^2 + 1$, $r := \max\{2t_0(c-1), (k+1)l\}$, $q := (l!+1)^2$, and $t := R_l(r, q) + l$, where $R_l(r, q)$ is as in Theorem 10.4.5. Then $r \geq l$ and $t \geq 2r$ as clearly $R_l(r, q) \geq 2r - 1$. We will show that this choice for t satisfies the lemma. Let D be a digraph satisfying $\nu_0(D) < c$ and $\tau_0(D) \geq t$. Choose a feedback vertex set T of D of cardinality $\tau_0(D)$ and an l-subset U of T. Let $Z := \{z_1, z_2, \ldots, z_{|Z|}\} := T - U$. Thus, $|Z| \geq R_l(r, q)$.

For each $X \subseteq Z$, with $X = \{z_{i_1}, \ldots, z_{i_{|X|}}\}$ where $i_1 < \ldots < i_{|X|}$; we put $\bar{X} := (z_{i_1}, \ldots, z_{i_{|X|}})$ and $\bar{X}(h) = z_{i_h}$ for $h = 1, \ldots, |X|$.

Let X be an l-subset of Z. If there is an l-path subdigraph $\mathcal{L}_1(X)$ in D from U to X containing no vertex in $Z - X$, then there is a permutation $(u_1, \ldots, u_l)$ of the vertices of U so that $\mathcal{L}_1(X)$ links $(u_1, \ldots, u_l)$ to $\bar{X}$, and we put $p_1(X) := (u_1, \ldots, u_l)$; if no such path subdigraph exists we put $p_1(X) := \emptyset$. Similarly, if there is an l-path subdigraph $\mathcal{L}_2(X)$ from X to U that links $\bar{X}$ to $(w_1, \ldots, w_l)$ containing no vertex in $Z - X$, we put $p_2(X) := (w_1, \ldots, w_l)$; if no such linkage exists we put $p_2(X) := \emptyset$. We assign to X the colour $(p_1(X), p_2(X))$. Clearly, there are q possible colours (q is defined in the beginning of this proof). By Theorem 10.4.5, there exist an r-subset S of Z and a colour (u, w) such that every l-subset X of S is of colour (u, w).

We claim that both u and w are non-empty. Indeed, suppose that $u = \emptyset$ and choose an r-set U' such that $U \subseteq U' \subseteq T - S$. By Lemma 10.4.6 there is an r-path subdigraph $\mathcal{L}'$ in D from U' to S containing no vertex in $T - (U' \cup S)$. The path subdigraph $\mathcal{L}'$ includes a path subdigraph from U to some $X \subseteq S$ having no vertex in $T - (U \cup X)$. Thus, $u = p_1(X) \neq \emptyset$. Analogously, one proves that $w \neq \emptyset$.

Let $u := (u_1, \ldots, u_l)$ and $w := (w_1, \ldots, w_l)$ and let $\mathcal{L}_1(X), \mathcal{L}_2(X)$ be the corresponding linkings. We have already established that for every l-subset X of S, $\mathcal{L}_1(X)$ links u to $\bar{X}$ and $\mathcal{L}_2(X)$ links $\bar{X}$ to w.

For $i = 1, \ldots, l$ define j_i as follows: $w_{j_i} = u_i$. By the definition of l and Theorem 5.2.3 of Erdős and Szekeres, there are $1 \leq i_1 < i_2 < \ldots < i_k \leq l$ so that the sequence $j_{i_1}, j_{i_2}, \ldots, j_{i_k}$ either increases or decreases. Define $(i'_1, \ldots, i'_k)$ to be $(j_{i_1}, \ldots, j_{i_k})$ in the first case and $(j_{i_k}, \ldots, j_{i_1})$ in the second. Hence, $i'_1 < \ldots < i'_k$.

Let $G := \{\bar{S}(l), \bar{S}(2l), \ldots, \bar{S}(kl)\}$. Choose an l-subset X of S so that $\bar{S}(hl) = \bar{X}(i_h)$ for $h = 1, \ldots, k$. Since $\mathcal{L}_1(X)$ links $(u_1, \ldots, u_l)$ to $\bar{X}$, it includes a path subdigraph $\mathcal{L}_1$ linking $(u_{i_1},\dots,u_{i_k})$ to $\bar{G}$. Moreover, the only vertices of T in $\mathcal{L}_1$ belong to $U \cup G$.

Analogously choose an l-subset Y of S so that $\bar{S}(hl) = \bar{Y}(i'_h)$ for $h = 1,\dots,k$. Since $\mathcal{L}_2(Y)$ links $\bar{Y}$ to $(w_1,\dots,w_l)$, it includes a path subdigraph $\mathcal{L}_2$ linking $\bar{G}$ to $(w_{i'_1},\dots,w_{i'_k})$. Observe that $(w_{i'_1},\dots,w_{i'_k})$ is either $(u_{i_1},\dots,u_{i_k})$ or $(u_{i_k},\dots,u_{i_1})$. Moreover, every (directed) cycle in $\mathcal{L}_1 \cup \mathcal{L}_2$ meets T (since T is a feedback vertex set), and the only vertices of T in $V(\mathcal{L}_1 \cup \mathcal{L}_2)$ are $u_{i_1},\dots,u_{i_k}$ and the elements of G; and so $\mathcal{L}_1, \mathcal{L}_2$ satisfy the lemma. □

A digraph D is **bivalent** if, for every $v \in V(D)$, $d^+(v) = d^-(v) \in \{1,2\}$. The following lemma is the most technically involved basic result in [626].

Lemma 10.4.8 *For every integer $c \geq 1$ there exists $k \geq 0$ such that, for every bivalent digraph D, if there exists a pair of k-path subdigraphs $\mathcal{L}_1, \mathcal{L}_2$ in D so that each path of $\mathcal{L}_1$ meets each path of $\mathcal{L}_2$ and $\mathcal{L}_1 \cup \mathcal{L}_2$ has no (directed) cycles, then $\nu_0(D) \geq c$.* □

Using this lemma and Theorem 10.4.5, one can prove the following:

Lemma 10.4.9 *For every integer $c \geq 1$ there exists $k \geq 0$ so that the following holds. Let D be a digraph and let $u_1,\dots,u_k,w_1,\dots,w_k$ be distinct vertices of D. Let $\mathcal{L}_1, \mathcal{L}_2$ be path subdigraphs in D linking $(u_1,\dots,u_k)$ to $(w_1,\dots,w_k)$ and $(w_1,\dots,w_k)$ to one of $(u_1,\dots,u_k)$, $(u_k,\dots,u_1)$, respectively. If every (directed) cycle of $\mathcal{L}_1 \cup \mathcal{L}_2$ meets $\{u_1,\dots,u_k,w_1,\dots,w_k\}$, then $\nu_0(D) \geq c$.* □

Theorem 10.4.10 (Reed, Robertson, Seymour and Thomas) *[626] For every integer $c \geq 1$ there exists a (minimum) integer $t_0(c)$ such that, for every digraph D with $\nu_0(D) < c$, we have $\tau_0(D) \leq t_0(c)$.*

Proof: We use induction on $c \geq 1$. For $c = 1$, this theorem is trivially true. Assume that $c \geq 2$ and $t_0(c-1)$ exists. Let k be as in Lemma 10.4.9, and let t be as in Lemma 10.4.7. We prove that there is no digraph D with $\nu_0(D) < c$ and $\tau_0(D) > t-1$ (i.e., $t_0(c) \leq t-1$). Suppose that D is such a digraph. By Lemma 10.4.7, there exist $u_1,\dots,u_k,w_1,\dots,w_k$ and $\mathcal{L}_1, \mathcal{L}_2$ as in Lemma 10.4.7. This means, by Lemma 10.4.9, that $\nu_0(D) \geq c$, a contradiction. □

10.5 Application: The Period of Markov Chains

Markov chains are a special type of stochastic processes, which have numerous applications in genetics, economics, sport science, etc. We will see in this section that the corresponding digraph cycle structure is of great importance to Markov chains.

Let $S_1, S_2, \dots, S_n$ be all possible states of some system. The system is initially in a state S_i with probability $p_i^{(0)}$, $i = 1,2,\dots,n$. At every step the system moves from the state S_i, which it is currently in, to a state S_j with probability p_{ij} depending only on i and j. Clearly, for all i, j, we have $0 \le p_{ij} \le 1$ and $\sum_{j=1}^n p_{ij} = 1$ for every $i = 1, 2, \dots, n$. The stochastic process, which we have under these conditions, is called a **Markov chain** (for more details on Markov chains, see e.g. Feller [234] and Kemeny and Snell [476])[6]. Let $\pi^{(0)} = (p_1^{(0)}, \dots, p_n^{(0)})$, let $p_i^{(k)}$ be the probability of the system to be in state S_i after the kth step, and let $\pi^{(k)} = (p_1^{(k)}, \dots, p_n^{(k)})$. It is well-known that the vector $\pi^{(k)}$ can be found as follows: $\pi^{(k)} = \pi^{(0)} P^k$, where $P = [p_{ij}]$. However, this equality is difficult to use directly if we wish to know the probability distribution $\pi^{(k)}$ after a large number of steps. In fact, $\pi = \lim_{n\to\infty} \pi^{(0)} P^k$ is often of interest (if it exists).

To investigate when this limit exists and to see what happens when this limit does not exist, it is very useful to study directed pseudographs D associated with Markov chains. The vertex set of D is $\{v_1, \dots, v_n\}$ and the arc set is $\{v_i v_j : \ p_{ij} > 0, 1 \le i, j \le n\}$; D has no parallel arcs but may have loops. It is not difficult to see that for $n\to\infty$ with probability tending to 1 the system will be in one of the stages corresponding to the vertices in the terminal strong components of D (once the system is in such a 'vertex' it cannot escape the corresponding terminal strong component.) This shows that it suffices to study only strong directed pseudographs D corresponding to Markov chains. When D is strong, the following parameter of D is of interest. The **period** $p(D)$ of D is the greatest common divisor of the cycle lengths of D. If $p(D) = 1$, then it is well-known that the limit above does exist and, moreover, does not depend on the initial distribution $\pi^{(0)}$. If $p(D) \ge 2$, then the situation is absolutely different since D has a quite special structure. Actually, if $p(D)$ is even, then by Theorem 1.8.1 we obtain that D is bipartite. However, the following stronger assertion, which generalizes Theorem 1.8.1, holds[7]:

Theorem 10.5.1 *If a strong digraph $D = (V, A)$ has period $p \ge 2$, then V can be partitioned into sets $V_1, V_2, \dots, V_p$ such that every arc with tail in V_i has its head in V_{i+1} for every $i = 1, 2, \dots, p$, where $V_{p+1} = V_1$.*

Proof: Let $D = (V, A)$ have period $p \ge 2$. Every closed walk W of D, being an eulerian digraph, is the union of cycles (see Theorem 6.8.1); hence the length of W equals 0 modulo p. Let x, y be a pair of distinct vertices of D and let P, Q be a pair of distinct (x, y)-paths in D. We claim that the lengths of P and Q are equal modulo p. Indeed, let R be an (y, x)-path in D. Both P and Q form closed walks with R; hence our last claim follows from the remark above.

[6] Some readers may find useful to consider $S_1, \dots, S_n$ as water containers, $p_i^{(0)}$ as the fraction of water in S_i initially, and p_{ij} as the fraction of water in S_i to be moved to S_j in one step. We are interested in how the water will be distributed after a large number of steps.

[7] We have been unable to trace the first paper, where this result was proved. Our proof of this theorem makes use of some results considered in previous chapters.

Since D is strong, it can be constructed from a cycle using ear composition (see Section 7.2). We start from a cycle C and in every iteration add to the current digraph H a path whose vertices apart from the end-vertices do not belong to H or a cycle with only one vertex in common with H. Initially, all sets $V_1, V_2, \ldots, V_p$ are empty. We choose an arbitrary vertex x in C and consider every vertex y in C; we put y in V_i if the length of $C[x, y]$ equals i modulo p. In the first iteration of ear composition, we add a path or cycle R to C. Let z be the initial vertex of R if R is a path or the only vertex of R in common with C if R is a cycle, and let $z \in V_k$. We consider every vertex y in R and put y in V_{k+i} if the length of $R[z, y]$ equals i modulo p. Note that, if R is a path, then its terminal vertex z' will be put in the same set V_j, where it has been already, since otherwise we could find a pair of (z, z')-paths, whose lengths are not equal modulo p. We proceed with ear composition as above and in the end we will have V partitioned into $V_1, V_2, \ldots, V_p$ such that every arc with tail in V_i has its head in V_{i+1} for every $i = 1, 2, \ldots, p$ (by the way we have formed V_i's). $\square$

Clearly, when the period of the digraph of a Markov chain is larger than 1, the limit introduced above does not exist; instead the Markov chain moves 'cyclically'. Theorem 10.5.1 shows that a strong digraph D of order n and period $p \geq 2$ is a spanning subdigraph of $\vec{C}_p[\overline{K}_{n_1}, \ldots, \overline{K}_{n_p}]$, for some $n_1, n_2, \ldots, n_p$ such that $\sum_{i=1}^{p} n_i = n$. In particular, in terms of homomorphisms, we have $D {\rightarrow} \vec{C}_p$ (see Section 12.5).

There are two algorithms to compute the period of a strong digraph in optimal time $O(n+m)$. The first algorithm is by Balcer and Veinott [39] and based on the following idea. If, for a vertex x of $d^+(x) \geq 2$, we contract all vertices in $N^+(x)$ and delete any parallel arcs obtained, then the resulting digraph has the same period as the original digraph by Theorem 10.5.1. Repeating this iteration, we will finally obtain a cycle C (see Exercise 10.25). Clearly, the length of C is the desired period. For example, the digraph H obtained from a 3-cycle and a 6-cycle by identifying one of their vertices after five iterations above becomes a 3-cycle (see Figure 10.2). The second algorithm is due to Knuth (see [29]) and based on DFS-trees.

10.6 Cycles of Length k Modulo p

The linear-time algorithms mentioned in Section 10.5 show that the problem to verify whether all cycles of a digraph are of length 0 modulo p for some p is polynomial time solvable. This problem has the natural 'existence' analogue: given a (fixed) integer $p \geq 2$, verify whether a digraph D has a cycle of length equal 0 modulo p. In this section, we consider this and the more general problem of the existence of cycles of lengths equal k modulo p. In Subsection 10.6.1, we study the complexity results on these problems; Subsection 10.6.2

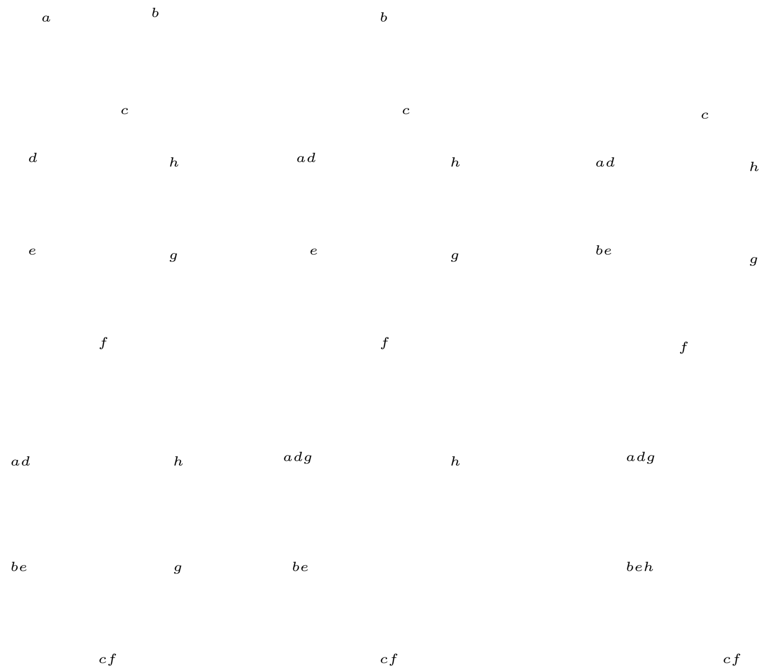


Figure 10.2 Illustrating the Balcer-Veinott algorithm.

is devoted to some sufficient conditions for the existence of cycles of lengths equal k modulo p.

10.6.1 Complexity of the Existence of Cycles of Length k Modulo p Problems

We start our consideration from the following problem. Given a (fixed) integer $p \geq 2$, verify whether a digraph D has a cycle of length equal 0 modulo p. The case of $p = 2$ of this problem is called the **even cycle problem**. The even cycle problem has numerous applications (see e.g. Robertson, Seymour and Thomas [643] and Thomassen [711] and the reference to further literature therein) and is related to several problems on permanents of matrices, so-called Pfaffian orientations of graphs, colouring of hypergraphs, etc. The complexity of the even cycle problem has been an open problem for quite some time: Thomassen [712] proved that the even cycle problem is polynomial time solvable for planar digraphs and Galluccio and Loebl [300] extended this result to digraphs, whose underlying undirected graphs do not contain subgraphs contractible to either K_5 or $K_{3,3}$. Finally, independently McCuaig, and Robertson, Seymour and Thomas (see [643]) found highly non-trivial proofs of the following result:

Theorem 10.6.1 *The even cycle problem is polynomial time solvable.* □

We are not aware of any paper determining the complexity of the problem to check whether a digraph has a cycle of length equal 0 modulo p for fixed $p > 2$.

Problem 10.6.2 *Is there a polynomial algorithm to check whether a digraph has a cycle of length equal 0 modulo p for fixed $p > 2$?*

The last problem can be naturally generalized to the problem to verify whether a digraph D has a cycle of length equal k modulo p for fixed k, p such that $0 \le k < p,\ p \ge 2$. We have considered the case of $k = 0$; the case of $k > 0$ was studied by Arkin, Papadimitriou and Yannakakis [29], who proved the following theorem (observe that the case of $k = 1$ and $p = 2$ is polynomial time solvable since one can check whether a digraph is bipartite in polynomial time):

Theorem 10.6.3 *Let k, p be a pair of fixed integers such that $0 < k < p,\ p > 2$. The problem to verify whether a digraph D has a cycle of length k modulo p is $\mathcal{NP}$-complete.*

Proof: Let D be a digraph and let $k \ge 2$. Choose k arbitrary arcs $a_1, a_2, \ldots, a_k$ in D and replace every arc xy in $A(D) - \{a_1, a_2, \ldots, a_k\}$ by an (x, y)-path of length p, whose intermediate vertices do not belong to D (and the intermediate vertices of all such paths are distinct). Clearly, the obtained digraph D' has a cycle of length equal k modulo p if and only if D has a cycle through all arcs $a_1, a_2, \ldots, a_k$. For a fixed $k \ge 2$, the problem of the existence of a cycle through k given arcs in a digraph is $\mathcal{NP}$-complete (see Proposition 9.2.1 and Theorem 9.2.3); hence this theorem is proved for $k \ge 2$. For $k = 1$, we choose a pair of arcs a, b, replace a by a path of length 2, b by a path of length $p - 1$, and every $c \in A(D) - \{a, b\}$ by a path of length p such that all internal vertices of the paths are distinct and distinct from the vertices of D. Clearly, the obtained digraph D' has a cycle of length equal 1 modulo p if and only if D has a cycle through a and b; the last problem is $\mathcal{NP}$-complete as we remarked above. □

Because of this theorem, the following result of Galluccio and Loebl [299] is of certain interest:

Theorem 10.6.4 *Let k, p be a pair of fixed integers such that $0 \le k < p,\ p \ge 2$. There is a polynomial algorithm to verify whether a planar digraph D has a cycle equal k modulo p.* □

10.6.2 Sufficient Conditions for the Existence of Cycles of Length k Modulo p

A digraph $D = (V, A)$ is called **even** if, for every $B \subseteq A$, the subdivision of all arcs in B results in a digraph with an even cycle. A ***k*-weak-double-cycle** is a digraph which is defined recursively as follows (see Figure 10.3):

Figure 10.3 The 5-double-cycle and a 5-weak-double-cycle.

1. The complete biorientation $\overleftrightarrow{C}_k$ of a k-cycle is a k-weak-double-cycle.
2. If H is a k-weak-double-cycle and D is obtained form H by subdividing an arc or splitting a vertex, then D is a k-weak-double-cycle.

It is easy to see that for odd k a k-weak-double cycle is even because it has an odd number of cycles and every arc is in an even number of distinct cycles (see Exercise 10.27). The following result is much more difficult to prove.

Theorem 10.6.5 (Seymour and Thomassen) *[667] A digraph is even if and only if it contains a k-weak-double-cycle for some odd k.* □

Galluccio and Loebl [301] have extended this result. They call a digraph $D = (V, A)$ **(k, p)-odd** if, for every $B \subseteq A$, the subdivision of all arcs in B results in a digraph with cycle of length different from k modulo p.

Theorem 10.6.6 *[301] A digraph is (k, p)-odd if and only if it contains a q-weak-double-cycle, with $(q-2)k \not\equiv 0$ (mod p).* □

Using Theorem 10.6.5 and other results, Thomassen [711] proved the following very interesting theorem:

Theorem 10.6.7 (Thomassen's even cycle theorem) *If D is a strong digraph with $\delta^0(D) \geq 3$, then D is even.* □

Koh [483] constructed an infinite family of digraphs D with $\delta^0(D) \geq 2$ and with no even cycle. Thomassen [702] strengthened this result by exhibiting, for every $k \geq 2$, a digraph D_k with $\delta^0(D_k) \geq k$ and with no even cycle. This implies that the strong connectivity requirement in Theorem 10.6.7 is necessary. Theorem 10.6.7 implies that every 3-strong digraph has an even cycle. Thomassen [705] pointed out that there exists a 2-strong digraph of order 7 that has no even cycle namely, the digraph in Figure 10.4.

Figure 10.4 A 2-strong digraph with no even cycle.

Thomassen [702] constructed infinitely many 2-strong digraphs that are not even. However, the following question is still open:

Problem 10.6.8 *[705] Are there infinitely many 2-strong digraphs with no even cycle?*

Theorem 10.6.7 was extended by Galluccio and Loebl [301], who proved that every strong digraph D with $\delta^0(D) \geq 3$ contains a cycle of length different from k modulo p, for every $1 \leq k < p,\ p \geq 3$.

Although we do not provide a proof of Theorem 10.6.7, we will prove Theorem 10.6.11 which implies a result weaker than Theorem 10.6.7, i.e. Corollary 10.6.12, but its assertion is not only on even cycles but also on cycles of length 0 modulo $q(\geq 2)$. To prove Theorem 10.6.11, we need two lemmas; the first lemma is the famous Lovász local lemma (cf. Alon and Spencer [14] or McDiarmid [560]). For an event E, $\overline{E}$ means that E does not hold.

Lemma 10.6.9 *Let $E_1, \ldots, E_n$ be events in an arbitrary probability space. Suppose that each event E_i is mutually independent of all other events except for at most d events, and that $\mathrm{Prob}(E_i) \leq p$ for every $i = 1, 2, \ldots, n$. If $ep(d+1) \leq 1$, where e is the basis of natural logarithms, then $\mathrm{Prob}(\cap_{i=1}^{n} \overline{E}_i) > 0$.* $\square$

Lemma 10.6.10 *[12] Let D be a digraph and let $q \geq 2$ be an integer. Suppose that every vertex x of D is assigned a colour $c^*(x)$, an integer in $\{0, 1, \ldots, q-$*

$1\}$*, such that for every* $u \in V(D)$ *there exists an out-neighbour* v *with* $c^*(v) \equiv c^*(u)+1$ *(mod* q*), then* D *contains a cycle of length 0 (mod* q*).*

Proof: Clearly, choosing an arbitrary vertex u_0 in $V(D)$, we can find a sequence $u_0, u_1, \ldots$ of vertices such that $u_iu_{i+1} \in A(D)$ and $c^*(u_{i+1}) \equiv c^*(u_i)+1 \pmod q$ for every $i \geq 0$. Let s be the least integer such that $u_j = u_s$ for some $j < s$. It remains to observe that the cycle $u_ju_{j+1}\ldots u_s$ is of length 0 (mod q). □

The following result is due to Alon and Linial:

Theorem 10.6.11 *[12] For a digraph* $D=(V,A)$*, if*

$$e(\Delta^-(D)\delta^+(D)+1)(1-1/q)^{\delta^+(D)} < 1 \tag{10.3}$$

or if

$$e(\Delta^+(D)\delta^-(D)+1)(1-1/q)^{\delta^-(D)} < 1 \tag{10.4}$$

then D *contains a cycle of length 0 (mod* q*).*

Proof: Since (10.4) tranforms into (10.3) by replacing D by its converse, it suffices to prove that (10.3) implies that D has a cycle of length 0 modulo q.

For every vertex u, delete $d^+(u)-\delta^+(D)$ arcs with tail u and consider the resulting digraph $D'=(V,A')$. Assign to every vertex u of D' a colour $c(u)$, an integer in $\{0,1,\ldots,q-1\}$, independently according to a uniform distribution. For each $u \in V$, let E_u denote the event that there is no $v \in V$ with $uv \in A'$ and $c(v) \equiv c(u)+1 \pmod q$. Clearly, $\mathrm{Prob}(E_u) = (1-1/q)^{\delta^+(D)}$. It is not difficult to verify that each event E_u is mutually independent of all the events E_v except for those satisfying

$$N^+(u) \cap (v \cup N^+(v)) \neq \emptyset.$$

The number of such v's is at most $\Delta^-(D)\delta^+(D)$ and hence, by our assumption (10.3) and Lemma 10.6.9, $\mathrm{Prob}(\cap_{u\in V}\overline{E}_u) > 0$. This means that there is a colouring c^* such that for every $u \in V$ there exists a $v \in V$ with $uv \in A'$ and $c^*(v) \equiv c^*(u)+1 \pmod q$. Now it follows from Lemma 10.6.10 that D has a cycle of length 0 modulo q. □

An easy proof of the following corollary is left as Exercise 10.33.

Corollary 10.6.12 *Every* k*-regular digraph* D *with* $k \geq 8$ *contains an even cycle.* □

We have seen above that no constant k can guarantee that a digraph of out-degree at least k contains an even cycle. This leads to the following natural question (raised by Erdős, see [702]): what is the smallest integer $h(n)$ such that every digraph of order n and minimum out-degree $h(n)$ contains an even cycle? In order to prove an upper bound for $h(n)$ we need a result on hypergraph colouring. The following lemma is due to Beck [98][8]:

[8] Recently, Radhakrishnan and Srinivasan [618] improved the bound of this lemma to $0.7{\cdot}2^m\sqrt{m/\ln m}$. Hence, the bound of Lemma 10.6.14 can slightly improved.

Lemma 10.6.13 *There exists an absolute constant d such that every m-uniform hypergraph with at most $\lfloor dm^{1/3}2^m \rfloor$ edges is 2-colourable.* □

Lemma 10.6.14 *[12] For every $n \geq 2$,*

$$h(n) \leq \log_2 n - \frac{1}{3} \log_2 \log_2 n + O(1).$$

Proof: Let $m \geq 2$ be an integer and let d be a constant satisfying Lemma 10.6.13. Suppose that

$$n = \lfloor dm^{1/3}2^m \rfloor \tag{10.5}$$

and let $D = (V, A)$ be a digraph of order n and $\delta^+(D) \geq m - 1$. Let H be the hypergraph on the set of vertices V, whose n edges are the sets $N^+[u] = N^+(u) \cup u$. Since every edge of H is of cardinality at least m, Lemma 10.6.13 implies that H is 2-colourable. This means that there exists a vertex colouring $c^* : V \to \{0, 1\}$ such that for every $u \in V$ there is $v \in N^+(v)$ with $c^*(v) \equiv c^*(u) + 1 \pmod 2$. Hence, by Lemma 10.6.10, D has an even cycle. Solving for m from (10.5) we obtain that

$$h(n) \leq m - 1 \leq \log_2 n - \frac{1}{3} \log_2 \log_2 n + O(1).$$

□

Clearly, if a digraph D contains cycles of length k and $k + 1$ for some k, then D has an even cycle. Deciding the existence of such cycles of consecutive length in a strong digraph as $\mathcal{NP}$-complete (see Exercise 10.37). Furthermore, it is easy to construct digraphs of arbitrary high vertex-strong connectivity with no such cycles (Exercise 10.38). It would be interesting to find non-trival degree conditions (weaker than conditions implying pancyclicity, such as those in Section 6.5) which guarantee that a non-bipartite digraph has two cycles of consecutive lengths. See also Exercise 1.49 for another type of sufficient condition for the existence of two cycles of consecutive lengths.

10.7 ‘Short’ Cycles in Semicomplete Multipartite Digraphs

As we mentioned in Chapter 5 the hamiltonian cycle problem is $\mathcal{NP}$-complete for arbitrary digraphs and polynomial time solvable for certain families of digraphs including semicomplete multipartite digraphs. In this section we consider the existence of ‘short’ cycles in semicomplete multipartite digraphs. By short cycles in a semicomplete p-partite digraph we mean cycles of length at most p.

The cycle structure of semicomplete bipartite digraphs is quite well understood due to Theorem 5.7.4 and Exercises 6.33, 6.34. The cycle structure of semicomplete p-partite digraphs, $p \geq 3$, is less investigated especially for cycles of length more than p. In this section, we will consider results on cycles of length at most p. Most of the results on short cycles in semicomplete multipartite digraphs were actually obtained for multipartite tournaments. Therefore, we state them for multipartite tournaments. However, all of them can be immediately extended to semicomplete multipartite digraphs due to the following theorem of Volkmann.

Theorem 10.7.1 *[728] Let D be a strong semicomplete p-partite digraph of order n, $p, n \geq 2$, with a cycle C of length at least 3. Then D contains a strong orientation containing the cycle C, if and only if $D \neq \overleftrightarrow{K}_{1,n-1}$.* □

Interestingly enough the analogue of this theorem does not hold for longest paths, see Exercise 10.39 (some relaxation of the analogue still holds, see Exercise 10.40). It is often more convenient to work with the following easy corollary of this theorem.

Corollary 10.7.2 *[728] Every strong semicomplete p-partite digraph, $p \geq 3$, contains a spanning strong oriented subgraph.* □

One of the most interesting results on the topic is the following theorem.

Theorem 10.7.3 (Guo and Volkmann) *[350] Let D be a strong p-partite tournament, $p \geq 3$, with partite sets $V_1, \ldots, V_p$. For each $i \in \{1, 2, \ldots, p\}$, there exists a vertex $v \in V_i$ belonging to an s-cycle of D for every $s \in \{3, 4, \ldots, p\}$.*

Proof: It suffices to prove that V_1 has a vertex v which is on an s-cycle of D for every $s \in \{3, 4, \ldots, p\}$. We proceed by induction on s.

We will first show that D has a 3-cycle through a vertex in V_1. Let $C = v_1v_2 \ldots v_kv_1$ be a shortest cycle through a vertex, say v_1, in V_1. Suppose that $k \geq 4$. By the minimality of k, $v_3 \in V_1$, since otherwise $v_3 \rightarrow v_1$ implying the 3-cycle $v_1v_2v_3v_1$ through a vertex in V_1, a contradiction. This means that $v_4 \notin V_1$; without loss of generality assume that $v_4 \in V_2$. Since $k \geq 4$ is minimal and $v_3 \in V_1$, we conclude that $v_4 \rightarrow v_1$, i.e. $k = 4$, and $v_2 \in V_2$. If there is a vertex $x \in V - (V_1 \cup V_2)$ which dominates a vertex of C and is dominated by a vertex in C, then there exists $i \in \{1, 2, 3, 4\}$ such that $v_{i+1} \rightarrow x \rightarrow v_i$ (indices modulo 4), which implies that there is a 3-cycle through v_1 or v_3, a contradiction.

This means that the set $V(D) - (V_1 \cup V_2)$ can be partitioned into sets S_1, S_2 such that $S_2 \rightarrow V(C) \rightarrow S_1$. Assume without loss of generality that $S_1 \neq \emptyset$. Since D is strong there is a path from S_1 to C. Let $P = x_1x_2 \ldots x_q$ be a shortest such path. Clearly, $q \geq 3$. If P has no vertex in S_2, then one of the vertices x_2, x_3 belongs to V_1 and the other to V_2 ($V - (S_1 \cup S_2) \subset V_1 \cup V_2$). By the minimality of P, $x_3 \rightarrow x_1$ implying that $x_1x_2x_3x_1$ is a 3-cycle containing a vertex in V_1, a contradiction. Therefore, P has a vertex in S_2. By the minimality of P and $S_2 \rightarrow C$, it follows that $x_{q-1} \in S_2$. If $q = 3$, then $v_1x_1x_2v_1$ is a 3-cycle, a contradiction. So, assume that $q \geq 4$. Since x_{q-2} cannot be in $S_1 \cup S_2$, $x_{q-2} \in V_1 \cup V_2$. If $x_{q-2} \in V_1$, we have $v_2 \rightarrow x_{q-2}$ implying that $x_{q-2}x_{q-1}v_2x_{q-2}$ is a 3-cycle, a contradiction. Finally, if $x_{q-2} \in V_2$, then $v_1x_{q-2}x_{q-1}v_1$ is a 3-cycle, a contradiction. We have shown that D has a 3-cycle containing a vertex in V_1.

Suppose now that $3 \leq s < p$ and some vertex u_1 of V_1 is contained in a k-cycle for every $k = 3, 4, \dots, s$. Assume, on the other hand, that

$$\text{no vertex of } V_1 \text{ is in a } k\text{-cycle for any } k = 3, 4, \dots, s, s+1. \qquad (10.6)$$

Let $u_1u_2 \dots u_su_1$ be an s-cycle of D and let S be the union of partite sets of D not represented in C. We claim that there is no vertex in S, which dominates a vertex in C and is dominated by a vertex in C. Indeed, if such a vertex existed one could insert it into C, a contradiction with (10.6). This means that S can be partitioned into sets S_1, S_2 such that $S_2 \rightarrow C \rightarrow S_1$. Assume without loss of generality that $S_1 \neq \emptyset$. Since D is strong there is a path from S_1 to C. Let $P = y_1y_2 \dots y_q$ be a shortest such path. Clearly, $q \geq 3$.

Assume that P has a vertex of S_2. Clearly, $y_{q-1} \in S_2$ and no other vertex of P is in S_2. If $y_{q-2} \notin V_1$, then $y_{q-2}y_{q-1}C[u_3, u_1]y_{q-2}$ is an $(s+1)$-path containing u_1, a contradiction with (10.6). Hence, $y_{q-2} \in V_1$ and $u_2 \rightarrow y_{q-2}$. Now we see that $u_2y_{q-2}y_{q-1}P[u_4, u_2]$ (or $u_1u_2y_{q-2}y_{q-1}u_1$, if $s = 3$) is an $(s+1)$-cycle containing u_1, a contradiction with (10.6). Thus, we conclude that P has no vertex of S_2.

Assume that P contains a vertex y_l of V_1. Let l be chosen such that $\{y_1, y_2, \dots, y_{l-1}\} \cap V_1 = \emptyset$. Assume that $q \leq s$. Due to the facts that every vertex of C dominates y_1, for every $k = 3, 4, \dots, s+1$, and $y_l \rightarrow \{y_1, y_2, ..., y_{l-2}\}$, there is a k-cycle C_k containing parts of C and P; C_k includes $y_l \in V_1$, a contradiction with (10.6). Therefore, $q \geq s+1$. Assume that $l \leq s+1$. Since $y_i \rightarrow y_1$, for every $i = 3, 4, \dots, s+1$, we obtain that $P[y_1, y_i]y_1$ is an i-cycle containing y_l, a contradiction with (10.6). Thus, we conclude that $l \geq s+2$. In the cycle $C' = P[y_1, y_l]y_1$, the vertex y_l dominates every vertex. Hence, for every $i = 3, 4, \dots, s+1$ we can construct an i-cycle using part of the vertices of C' including y_l, a contradiction with (10.6).

Thus, P has no vertex in V_1. Hence, u_1 dominates every vertex in P. If $q \geq k+1$, then $u_1P[y_{q-k}, y_q]C[u_{k+1}, u_1]$ would be an $(s+1)$-cycle containing u_1, a contradiction with (10.6). Therefore, $q \leq k$. Since every vertex of C dominates y_1, $PC[u_{k+1}, u_{k-q+1}]y_1$ is an $(s+1)$-cycle containing u_1, a contradiction with (10.6).

Thus, the assumption (10.6) resulted in a contradiction. This proves our theorem. $\square$

This theorem generalizes several other results on multipartite tournaments and (ordinary) tournaments. Three of them are Moon's theorem on vertex pancyclic tournaments, Theorem 1.5.1, and the following extension of Theorem 1.5.1 by Gutin.

Corollary 10.7.4 *[364] Let D be a strong p-partite tournament, $p \geq 3$, such that one partite set of D consists of a single vertex v. Then for each $k \in \{3, 4, \ldots, p\}$, D contains a k-cycle through v.* $\square$

By Theorem 10.7.1, Corollary 10.7.4 can be extended to semicomplete p-partite digraphs, $p \geq 3$. Theorem 10.7.3 generalizes the following assertion, due to Bondy, which was actually the first non-trivial result on cycles in multipartite tournaments. Again, Corollary 10.7.5 can be extended to semicomplete p-partite digraphs, $p \geq 3$.

Corollary 10.7.5 *[124] A strong p-partite tournament contains an s-cycle for every $s \in \{3, 4, \ldots, p\}$.* $\square$

The assertion of this corollary is the best possible in the sense that for every $p \geq 3$ there exists a strong p-partite tournament with no cycle of length more than p. The following example is due to Bondy [124]. Let H be a p-partite tournament with partite sets $V_1 = \{v\}, V_2, \ldots, V_p$ such that $|V_i| \geq 2$ for each $2 \leq i \leq p$. If $V_2 \rightarrow v \rightarrow \cup_{j=3}^{p} V_j$ and $V_j \rightarrow V_i$ for $2 \leq i < j \leq p$, then H is strong but does not have a k-cycle for every $k > p$.

Another interesting generalization of Moon's theorem is due to Goddard and Oelermann [322].

Theorem 10.7.6 *Every vertex of a strong p-partite tournament D belongs to a cycle that contains vertices from exactly t partite sets of D for each $t \in \{3, 4, \ldots, p\}$.* $\square$

It is left as Exercise 10.41 to show that Theorem 10.7.3 is the best possible in the following sense: for every $p \geq 3$ there exists a strong p-partite tournament T such that some vertex v of T is not contained in a k-cycle for some $3 \leq k \leq p$. If one wishes to consider only cycles through a given vertex of a multipartite tournament, one perhaps should sacrifice the exactness. This is illustrated by the following result due to Guo, Pinkernell and Volkmann.

Theorem 10.7.7 *[347] If D is a strong p-partite tournament and v an arbitrary vertex of D, then v belongs to either a k-cycle or a $(k+1)$-cycle for every $k \in \{3, 4, \ldots, p\}$.* $\square$

For regular multipartite tournaments Guo and Kwak proved the following much stronger result. Observe that the partite sets of a regular multipartite tournament are of the same cardinality.

Theorem 10.7.8 *[346] Let D be a regular p-partite tournament. If the cardinality of the partite sets of D is odd, then every arc of D is on a cycle that contains vertices from exactly k partite sets for each $k \in \{3, 4, \ldots, p\}$.* $\square$

This theorem generalizes the corresponding result by Alspach [19] on regular tournaments. The next theorem is another generalization of Alspach's theorem.

Theorem 10.7.9 *[345] Let D be a regular p-partite tournament. If every arc of D is contained in a 3-cycle, then every arc of D is on a k-cycle for each $k \in \{3, 4, \dots, p\}$.* $\square$

10.8 Cycles Versus Paths in Semicomplete Multipartite Digraphs

For a digraph D, the numbers $\mathrm{lp}(D)$ ($\mathrm{lc}(D)$, respectively) denote the number of vertices in a longest path (cycle, respectively) of D. The existence of an acyclic semicomplete multipartite digraph containing a Hamilton path and a hamiltonian semicomplete multipartite digraph suggests that there are no relations between the lengths of longest paths and cycles apart from trivial ones. However, the situation becomes quite different when we consider strong semicomplete multipartite digraphs. Volkmann [730] conjectured that, if D is a strong semicomplete multipartite digraph then $\mathrm{lp}(D) \leq 2 \cdot \mathrm{lc}(D) - 1$.

The example of Bondy from Section 10.7 shows that the bound on $\mathrm{lp}(D)$ is sharp. Volkmann's conjecture was settled in affirmative by Gutin and Yeo [382] (see Theorem 10.8.3). The aim of this section is to present an interesting proof given in [382]. However, we first state a more general conjecture of Volkmann. Recall that $\alpha(D)$ denotes the cardinality of a maximum independent vertex set of D.

Conjecture 10.8.1 *[728] Let D be a strongly connected semicomplete multipartite digraph with $\kappa(D) < \alpha(D)$. Then $\kappa(D)\mathrm{lp}(D) \leq (\kappa(D) + 1)\mathrm{lc}(D) - \kappa(D)$.* $\square$

The condition $\kappa(D) < \alpha(D)$ is given since every semicomplete multipartite digraph D with $\kappa(D) \geq \alpha(D)$ is hamiltonian by Corollary 5.7.25 and thus the conjecture is not of interest for $\kappa(D) \geq \alpha(D)$. Tewes and Volkmann [693] showed that the conjecture holds for $\kappa(D) = \alpha(D) - 1 \geq 1$.

For a path $P = x_1 x_2 \dots x_p$ we let $P[x_i, x_j[:= P[x_i, x_{j-1}]$.

Lemma 10.8.2 *[382] Let D be a semicomplete multipartite digraph. Let $Q_1, Q_2, \dots, Q_l$ be non-empty sets which form a partition of $V(D)$ such that $Q_i \Rightarrow Q_j$ for every $1 \leq i < j \leq l$. Assume that $|V(D)| > l$ and $D\langle Q_i \rangle$ has a Hamilton path $q_1^i q_2^i \dots q_{|Q_i|}^i$ for every $i = 1, 2, \dots, l$. Then, D has a $(q_1^1, q_{|Q_l|}^l)$-path with at least $|V(D)| - l + 1$ vertices.*

Proof: We use the induction on l. Clearly the theorem holds when $l = 1$, so assume that $l > 1$.

If $|V(D) - Q_l| > (l-1)$ then, by the induction hypothesis, there is a $(q_1^1, q_{|Q_{l-1}|}^{l-1})$-path, $p_1 p_2 .. p_k$, in $D - Q_l$ which contains $k \geq |V(D) - Q_l| - (l - 1) + 1 \geq 2$ vertices. Since $\{p_{k-1}, p_k\} \Rightarrow q_1^l$ and p_{k-1} and p_k belong to different partite sets, $p_{k-1} \rightarrow q_1^l$ or $p_k \rightarrow q_1^l$. Therefore, the path $p_1 p_2 \ldots p_s q_1^l q_2^l \ldots q_{|Q_l|}^l$, where $s = k-1$ or k, is of the desired type.

If $|V(D) - Q_l| \leq (l-1)$, then clearly $|Q_l| > 1$. Since $q_1^1 \Rightarrow \{q_1^l, q_2^l\}$ and q_1^l and q_2^l belong to different partite sets, $q_1^1 \rightarrow q_1^l$ or $q_1^1 \rightarrow q_2^l$. Therefore, the path $q_1^1 q_s^l q_{s+1}^l \ldots q_{|Q_l|}^l$, where $s \in \{1, 2\}$, is of the desired type. □

Theorem 10.8.3 *[382] Let D be a strong semicomplete multipartite digraph and let $l = \mathrm{lp}(D)$ be the number of vertices in a longest path in D and let $c = lc(D)$ be the number of vertices in a longest cycle in D. Then $l \leq 2c - 1$.*

Proof: Let $P = p_1 p_2 .. p_l$ be a path in D of maximum length and let $R = V(D) - V(P)$. Let $x_0 = p_l$ and define S_i, x_i and y_i recursively as follows ($i = 1, 2, \ldots$).

First let S_1' be a (p_l, p_k)-path in $D - V(P - \{p_l, p_k\})$, such that k is chosen as small as possible. Let $x_1 = p_k$, let $y_1 = p_l$ and let $S_1 = S_1' - \{x_1, y_1\}$ (note that $S_1 = \emptyset$, by the maximality of l). Now for $i = 2, 3, 4, \ldots$ let S_i' be a (p_t, p_k)-path in $D\langle \{p_t, p_k\} \cup R - (V(S_1) \cup V(S_2) \cup \ldots \cup V(S_{i-1}))\rangle$, such that $p_t \in V(P[x_{i-1}, p_l])$ and $p_k \in V(P[p_1, x_{i-1}[)$, and firstly k is chosen as small as possible, thereafter t is chosen as large as possible. Let also $x_i = p_k$, $y_i = p_t$ and $S_i = S_i' - \{x_i, y_i\}$. (Some paths S_i can be empty, meaning that S_i' is just an arc.)

We continue the above process until $x_i = p_1$. Let the last value of i found above be denoted by m (i.e. $x_m = p_1$). Observe that the paths S_i' always exist as D is strong. Observe also that $y_1 = p_l$ and that T below is a path in D:

$$T = y_1 S_1 P[x_1, y_2] S_2 P[x_2, y_3] S_3 \ldots P[x_{m-1}, y_m] S_m x_m.$$

Let $U_0 = P[x_m, x_{m-1}] - \{x_m, x_{m-1}\}$ and let $U_i = P[y_{m-i+1}, x_{m-i-1}] - \{y_{m-i+1}, x_{m-i-1}\}$ for $i = 1, 2, .., m-1$. Note that some of the U_i's ($i = 0, 1, 2, .., m-1$) can be empty. Observe that $V(T), U_0, U_1, \ldots, U_{m-1}$ partitions the set $V(P) \cup V(S_1) \cup \ldots \cup V(S_m)$. Let $Z_0, Z_1, \ldots, Z_{m'-1}$ be the non-empty sets among $U_0, U_1, \ldots, U_{m-1}$, where the relative ordering has been kept (i.e. if $Z_i = U_{i'}$, $Z_j = U_{j'}$ and $i' < j'$ then $i < j$). Let $B_0 = Z_0 \cup Z_2 \cup \ldots \cup Z_f$ and $B_1 = Z_1 \cup Z_3 \cup \ldots \cup Z_g$, where f (g, respectively) is the maximum even integer (odd integer, respectively) not exceeding $m' - 1$.

If $p_l \rightarrow p_1$, then we are done (the cycle Pp_1 is of length l). Thus, we may assume that p_1 is not dominated by p_l. As $x_m = p_1$, it follows from the way we constructed the paths above (always going as far back as possible) that

$$p_1 \Rightarrow Z_1 \cup Z_2 \cup \ldots \cup Z_{m'-1} \cup \{p_l\}. \tag{10.7}$$

Similarly, by the definitions of x_i, y_i and S_i,

$$Z_i \Rightarrow Z_{i+2} \cup Z_{i+3} \cup \ldots \cup Z_{m'-1} \cup \{p_l\},\ i = 0, 1, .., m' - 2. \tag{10.8}$$

As $\{x_0, x_1, \ldots, x_m\} \subseteq V(T)$ and $m \geq m'$, we have

$$|V(T)| \geq m' + 1. \tag{10.9}$$

As $V(T), B_0, B_1$ partitions the set $V(P) \cup V(S_1) \cup \ldots \cup V(S_m)$,

$$|V(T)| + |B_0| + |B_1| \geq l. \tag{10.10}$$

All $Z_i \cup \{x_i\}$ are disjoint sets containing at least two vertices. Thus, there are at most $l/2$ such sets. Hence, we obtain

$$m' \leq \frac{l}{2}. \tag{10.11}$$

We only consider the case when m' is odd since the case of even m' can be treated similarly.

If $|B_0|+2 > \frac{m'+1}{2}$ then, by (10.7), (10.8) and Lemma 10.8.2, there is a path W_0 from p_1 to p_l in $D\langle \{p_1, p_l\} \cup B_0 \rangle$ containing at least $|B_0| + 2 - \frac{m'+1}{2} + 1$ vertices (to use Lemma 10.8.2 we take $V(Q_1) = \{p_1\} \cup Z_0$, $V(Q_2) = Z_2$, $\ldots$, $V(Q_{(m'-1)/2}) = Z_{m'-3}$, $V(Q_{(m'+1)/2}) = Z_{m'-1} \cup \{p_l\}$). Analogously if $|B_1| > \frac{m'-1}{2}$, then there is a path W_1 from p_1 to p_l in $D\langle \{p_1, p_l\} \cup B_1 \rangle$ containing at least $|B_1| - \frac{m'-1}{2} + 1$ vertices (this time we take $V(Q_1) = \{p_1\}$, $V(Q_2) = Z_1$, $\ldots$, $V(Q_{(m'-1)/2}) = Z_{m'-2}$, $V(Q_{(m'+1)/2}) = \{p_l\}$).

We now consider the cases where none, one or two of the paths W_0 and W_1 exist.

Case 1 Both W_0 and W_1 exist: The cycle $C_0 = W_0T$ contains $|V(T)| + |V(W_0)| - 2$ vertices (as p_1 and p_l are counted twice). The cycle $C_1 = W_1T$ contains $|V(T)| + |V(W_1)| - 2$ vertices. By (10.9) and (10.10), this implies the following:

$$\begin{aligned} |V(C_0)| + |V(C_1)| &= 2|V(T)| + |V(W_0)| + |V(W_1)| - 4 \\ &\geq 2|V(T)| + (|B_0| + 2 - \tfrac{m'+1}{2} + 1) \\ &+ (|B_1| - \tfrac{m'-1}{2} + 1) - 4 \\ &= |V(T)| + (|V(T)| + |B_0| + |B_1|) - m' \\ &\geq |V(T)| + l - m' \\ &\geq l + 1. \end{aligned}$$

This implies that the largest cycle of C_0 and C_1 contains at least $\lceil (l+1)/2 \rceil$ vertices. Thus, we are done.

Case 2 Exactly one of W_0 or W_1 exists: Let $j \in \{0, 1\}$ be defined such that W_j exists, but W_{1-j} does not exist. Using (10.10) and (10.11), and observing that either $|B_0| + 2 - \frac{m'+1}{2} + 1 \leq 1$ or $|B_1| - \frac{m'-1}{2} + 1 \leq 1$, we obtain the following ($C_j = W_jT$, as above):

$$\begin{aligned}|V(C_j)| &\geq |V(T)| - 2 + (|B_0| + 2 - \tfrac{m'+1}{2} + 1) + (|B_1| - \tfrac{m'-1}{2} + 1) - 1\\ &\geq |V(T)| + |B_0| + |B_1| - m' + 1\\ &\geq l - m' + 1\\ &\geq l - \tfrac{l}{2} + 1\\ &\geq \tfrac{l+2}{2}.\end{aligned}$$

This is the desired result.

Case 3 Neither W_0 nor W_1 exists: This means that $|B_0|+2-\frac{m'+1}{2} \leq 0$ and $|B_1| - \frac{m'-1}{2} \leq 0$. Thus,

$$\begin{aligned}|V(T)| &\geq |V(T)| + (|B_0| + 2 - \tfrac{m'+1}{2}) + (|B_1| - \tfrac{m'-1}{2})\\ &= |V(T)| + |B_0| + |B_1| + 2 - m'\\ &\geq l - m' + 2\\ &\geq \tfrac{l+2}{2} + 1.\end{aligned}$$

If $p_1 \rightarrow p_l$ then there is a cycle of length at least $\lceil \frac{l+2}{2} + 1 \rceil$ (using all vertices in $V(T)$). By (10.7), p_l does not dominate p_1, so we may assume that p_1 and p_l are in the same partite set. We have $S_m = \emptyset$ as otherwise $S_m P$ is longer than P which is impossible, hence y_m and p_l are in different partite sets. If $p_l \rightarrow y_m$ then either $P[y_m, p_l]y_m$ or $P[p_1, y_m]p_1$ is a cycle with at least $\lceil \frac{l+1}{2} \rceil$ vertices. Therefore, we may assume that $y_m \rightarrow p_l$. Then the cycle $T[p_l, y_m]p_l$ contains at least $\lceil \frac{l+2}{2} \rceil$ vertices. We are done. □

10.9 Girth

Recall that the girth $g(D)$ of a digraph D is the length of a shortest cycle in D. The girth is an important parameter of a digraph and has been studied in a number of papers especially with respect to its extreme values.

Theorem 5.6.10 claims that, if the minimum degree of every vertex in a strong digraph D is large enough, then the length of a longest cycle in D is large as well. Caccetta and Häggkvist [139] conjectured a somewhat similar result for girth (with obvious replacement of upper bound to a lower bound):

Conjecture 10.9.1 (Caccetta and Häggkvist) *[139] Every digraph of minimum out-degree k and order n has a cycle of length at most $\lceil n/k \rceil$.*

This conjecture is trivially true for $k = 1$; it was proved for $k = 2$ by Caccetta and Häggkvist [139], for $k = 3$ by Hamidoune [396], and for $k = 4$ and 5 by Hoang and Reed [430]. Hamidoune [395] proved that the conjecture is true for digraphs with transitive group of automorphisms. As an application, he showed in [395] that for a finite group G of order n and a subset S of G of cardinality s, there is a collection of at most $\lceil n/s \rceil$ elements of S whose product equals the unit element of G. For an arbitrary integer $k \geq 1$, we have the following:

Theorem 10.9.2 (Chvátal and Szemerédi) *[163] There is a constant c such that every digraph of minimum out-degree $k \geq 1$ and order n contains a cycle of length at most $\lceil n/k \rceil + c$. Moreover, $c \leq 2500$.* $\square$

A straightforward refinement of the proof in [163] was used by Nishimura [593] to show that $c \leq 304$. For relatively small values of n/k, the following result of Chvátal and Szemerédi [163] is of interest.

Theorem 10.9.3 *Every digraph of minimum out-degree k and order n has a cycle of length at most $\lceil 2n/(k+1) \rceil$.*

Proof: By induction on $n \geq 2$. For $n = 2$ or 3 and $k \geq 1$, the digraph in question has either a 2-cycle or a 3-cycle and hence the claim holds. Let D be a digraph of order $n \geq 4$ and minimum out-degree $k \geq 1$. Since the size of D is at least kn, D contains a vertex v of in-degree at least k. If D has a 2-cycle, we are done. So, assume that D is an oriented graph. Let D' be the digraph obtained from D by deleting the vertices of $N^-[v] = N^-(v) \cup \{v\}$ and adding the new arc xy for every ordered pair x, y such that $xy \notin A(D)$, $y \in N^+(v)$ and x dominates an in-neighbour of v. Clearly, D' is of order at most $n-k-1$ and minimum out-degree at least k. By the induction hypothesis, D' contains a cycle C of length at most $2(n-k-1)/(k+1)$. Replacing each of the new arcs xy in C by the path $xuvy$, we obtain a closed walk C^* in D. If C has precisely s new arcs, then v appears on C^* exactly s times, and so C^* is the union (see Exercise 1.12) of at least s cycles, whose total length is at most $2(n-k-1)/(k+1) + 2s$. Clearly, the shortest of these cycles has length at most $2n/(k+1)$. $\square$

Searching for new approaches to the Caccetta-Hággkvist conjecture, Hoang and Reed [430] came up with the following conjecture that implies the Caccetta-Hággkvist conjecture (Exercise 10.43).

Conjecture 10.9.4 *Every digraph D of minimum out-degree k contains a sequence $C_1, C_2, \ldots, C_k$ of cycles such that $\cup_{i=1}^{j-1} C_i$ and C_j have at most one vertex in common.*

In the case of $k = 2$, the last conjecture was proved by Thomassen [704].

Theorem 10.9.5 *Every digraph D of minimum out-degree 2 contains a pair of cycles with precisely one vertex in common.*

Proof: By induction on n, the order of D. If $n = 3$, the claim trivially holds, so assume that $n \geq 4$. Since the minimum out-degree in the terminal strong component of D is at least 2, we may assume that D is strong. Moreover, since $\delta^+(D) \geq 2$, D has a vertex x such that $D - x$ is strong (see Exercise 10.44). If $D\langle N^-(x) \rangle$ contains a cycle C, then the required pair of cycles consists of C and a cycle formed by a shortest path P from x to C and the arc from the terminal vertex of P to x. So, we may assume that $D\langle N^-(x)\rangle$ is acyclic, and, thus, $D\langle N^-(x)\rangle$ has a vertex y of in-degree 0.

If we delete all arcs with tail y and identify x and y, we obtain the digraph D' of order $n-1$ and minimum out-degree at least 2. By the induction hypothesis, D' has a pair of cycles with precisely a vertex in common; these cycles correspond to cycles C_1 and C_2 in D. We may assume that C_1 and C_2 have yx in common for otherwise they have precisely a vertex in common. Since $D-x$ is strong, y is in a cycle C_3 of $D-x$. It is not difficult to see that $C_1 \cup C_2 \cup C_3$ contains a pair of cycles having precisely y in common. Indeed, if C_3 has only y in common with C_1 or C_2, then there is nothing to prove. If C_3 intersects with $C_1 \cup C_2$ at a vertex distinct from y, then let z be such a vertex with $C_3[y,z]$ being as short as possible (meaning that $C_3[y,z]$ has only y and z in common with $V(C_1) \cup V(C_2)$). Choose i such that z is in C_i, where $i=1$ or 2. Then C_{3-i} and $C_i[z,y]C_3[y,z]$ is the required pair of cycles. □

The **density** of a digraph D is the ratio of its size and order (i.e. m/n). Clearly, high density of a strong digraph D guarantees that $g(D)$ is small. Thomassen (see [112]) asked to determine the least number $m(n,k)$ such that every strong digraph of order n and size at least $m(n,k)$ contains a cycle of length at most k. Bermond, Germa, Heydemann and Sotteau [112] solved this problem by proving the following:

Theorem 10.9.6 *Let D be a strong digraph of order n and let $k \geq 2$. Then*

$$|A(D)| \geq \frac{n^2+(3-2k)n+k^2-k}{2}$$

implies that $g(D) \leq k$. □

This theorem is best possible since there exist strong digraphs of order n and size $(n^2+(3-2k)n+k^2-k)/2-1$ with shortest cycle of length $k+1$ (Exercise 10.45).

In many questions on properties of (di)graphs, one may ask whether all (di)graphs satisfying a certain property must have cycles of length at most a constant. Perhaps the most famous such question is the problem regarding the chromatic number of an undirected graph: given $k \geq 3$ and $g \geq 3$, is there an undirected graph of chromatic number k and of girth at least g? This problem was resolved in affirmative by Erdős [220] using probabilistic argument (a simplification of the original proof is given by Alon and Spencer [14]). Clearly, many digraphs of large vertex-strong connectivity are quite dense and, thus, of small girth. However, it is not difficult to construct digraphs of large vertex-strong connectivity and large girth. The 'vertex-strong connectivity' and 'girth' parts of the next result were proved by Ayoub and Frisch [34] (see Exercise 7.24) and Liu and Zhou [517] (see Exercise 10.42), respectively.

Proposition 10.9.7 *If $n = gs$, $g \geq 2$, then there exists an s-regular round digraph of order n which is s-strong and has girth g.* □

10.10 Additional Topics on Cycles

10.10.1 Chords of Cycles

The existence of chords of cycles is not only an interesting problem by itself, it has also several applications. One of these applications is the existence of kernels in digraphs (see Subsection 12.3.1), another one will be described in this subsection.

Let D be a directed multigraph with $\delta^0(D) \geq k$. It is not difficult to see that D has a cycle with at least $k-1$ chords. Indeed, let $P = p_1 p_2 \ldots p_k$ be a longest path in D. Clearly, there are k arcs from p_k to vertices of P. These arcs and part of P form the desired cycle with $k-1$ chords. While for $k = 1$ this result cannot be improved (consider $\vec{C}_n$ or 'tree-like' strong digraphs obtained from several cycles in such a way that every pair of cycles has at most one common vertex). Marcus [551] showed that for $k \geq 2$ the above simple result can be improved to the following:

Theorem 10.10.1 (Marcus' theorem) *[551] Let D be a strong directed multigraph with at least two vertices and $\delta^0(D) \geq k \geq 2$. Then D contains a cycle with at least k chords.* □

This result improves and extends the main assertion by Thomassen [713] that every 2-arc-strong directed multigraph has a cycle with at least two chords. The proof of Theorem 10.10.1 in [551] is quite involved and lengthy, and thus is not given here. Instead, we will consider an interesting application of Theorem 10.10.1 to the problem of minimum size strong spanning subgraphs of strong directed multigraphs (often called the minimum equivalent subdigraph problem, see the end of Section 4.3 and Section 6.11).

Lemma 10.10.2 *[550] Let k be a positive integer, let a and b be non-negative real numbers, and suppose that every k-arc-strong directed multigraph with at least two vertices has a strong subgraph H with at least two vertices and a strong factor[9] H_0 of H such that*

$$e_0 \leq ae + b(h-1),$$

where h is the order of H and e (e_0) is the size of H (H_0). Then every k-arc-strong directed multigraph of order n and size m has a strong factor with at most $am + b(n-1)$ arcs.

[9] Recall that a factor is a spanning subdigraph.

Proof: This holds trivially for directed multigraphs with one vertex since $a \geq 0$. Thus, consider a directed multigraph D of order $n \geq 2$ and assume that the result is true for all directed multigraphs with less than n vertices. By the assumption, D has a subgraph H as in the lemma. Clearly the contracted directed multigraph D/H is k-arc-strong and has $n-h+1 < n$ vertices; so D/H has a factor with $a(m-e)+b(n-h)$ arcs. The corresponding arcs of D, along with the e_0 arcs of H_0, form a factor of D of size at most $am+b(n-1)$. $\square$

Setting $a = \frac{1}{k+1}$ and $b = \frac{k}{k+1}$ in this lemma and using this lemma together with Theorem 10.10.1, we obtain the following (see Exercise 10.46):

Corollary 10.10.3 *[551] For $k \geq 2$, every k-arc-strong directed multigraph of size m and order $n \geq 2$ contains a strong factor of size at most $(m+k(n-1))/(k+1)$.* $\square$

10.10.2 Ádám's Conjecture

Ádám's conjecture [1, 2] seems one of the most interesting conjectures on cycles in digraphs.

Conjecture 10.10.4 (Ádám) *Every digraph has an arc whose reversal decreases the total number of cycles.*

Originally, Ádám formulated the conjecture for directed multigraphs. This extension was disproved independently by Grinberg and by Thomassen (see [334, 461, 706]). Thomassen [706] used the following result of Penn and Witte [601], which is of independent interest and was established with the aid of knot theory on the torus. Note that this theorem generalizes Theorem 5.11.6.

Theorem 10.10.5 *The cartesian product $\vec{C}_p \times \vec{C}_q$ has a cycle of length k if and only if there is a pair a, b of relatively prime natural numbers such that $ap + bq = k$.* $\square$

The main idea of Thomassen is to apply the following corollary:

Corollary 10.10.6 *[706] Infinitely many digraphs of the type $\vec{C}_p \times \vec{C}_q$ have the property that the reversal of any arc increases the length of a longest cycle.*

Proof: By the above theorem, $\vec{C}_5 \times \vec{C}_{7+10k}$, $k \geq 0$, has no cycle of length $35+50k$ or $34+50k$ (Exercise 10.47). However, the reversal of any arc creates a $(34+50k)$-cycle. This is depicted in Figure 10.5 (due to Thomassen [706]) for $k = 0$ and a similar structure can be used to obtain a cycle of length $34+50k$ when $k \geq 1$. (Actually, Figure 10.5 shows a 35-cycle, too, and this cycle can be generalized for every $k \geq 0$.) $\square$

Theorem 10.10.7 *[706] There is an infinite family of counterexamples to Ádám's conjecture in the case of directed multigraphs.*

Proof: Let $D(k, f)$ be the directed multigraph obtained from $\vec{C}_5 \times \vec{C}_{7+10k}$ by replacing each arc by f parallel arcs. Let t denote the maximum number of cycles through an arc of $\vec{C}_5 \times \vec{C}_{7+10k}$ and let s be the length of a longest cycle in $\vec{C}_5 \times \vec{C}_{7+10k}$. Then no arc of $D(k, f)$ is contained in more than tf^{s-1} cycles, but if we reverse an arc e of $\vec{C}_5 \times \vec{C}_{7+10k}$, then e is is contained in a cycle of length at least $s+1$ and hence e is contained in at least f^s cycles. Hence, if $f > t$, $D(k, f)$ is a counterexample to Ádám's conjecture. □

Figure 10.5 $\vec{C}_5 \times \vec{C}_7$ and (directed) cycles of lengths 34 and 35 when an arc is reversed. (All arcs represented by vertical or horizontal straight line segments are directed upwards or to the right.)[706]

Grinberg's counterexamples are inspired by projective geometry. All the examples by Thomassen and Grinberg have parallel arcs. At the same time, Ádám's conjecture holds for some families of digraphs. Actually, it holds when a digraph has a 2-cycle.

Proposition 10.10.8 *[462] If a digraph D contains a 2-cycle, then D has an arc whose reversal decreases the total number of cycles in D.*

Proof: Let uvu be a 2-cycle in D and, for every $a \in A(D)$, let c_a be the number of cycles in D containing a. Without loss of generality, we may assume that $c_{uv} \leq c_{vu}$. Then, the reversal of vu decreases the number of cycles in D by $c_{vu} - c_{uv} + 1 > 0$. □

Apart from this proposition, Jirásek [462] proved several other assertions on families of digraphs that satisfy Ádám's conjecture. The most interesting is the following:

Theorem 10.10.9 *If, after reversal of at most three arcs a non-acyclic digraph D becomes acyclic, then D has an arc whose reversal decreases the total number of cycles in D.*

To the best of our knowledge, Ádám's conjecture is still open for oriented graphs.

Problem 10.10.10 *[706] Verify Ádám's conjecture for oriented graphs and, in particular, for tournaments.*

10.11 Exercises

10.1. **(−)** Prove that for a strong digraph D the cycle space is generated by oriented cycles without chords.

10.2. Prove Proposition 10.1.1.

10.3. **(−)** Let D be a digraph such that is $UG(D)$ has c connected components. Prove that the dimension of the cycle space of D is $m - n + c$. Hint: apply Theorem 10.1.3 to every component of D.

10.4. Prove the following assertion. Let D, H be digraphs and let G_D and G_H be sets of oriented cycles generating the cycle spaces of D and H, respectively. Suppose further that $f: A(D) \rightarrow A(H)$ is a bijection such that $f(G_D) = G_H$. Then f and f^{-1} preserve oriented cycles (Thomassen [709]).

10.5. **(−)** Let $1 \leq k \leq n$ be integers. Let $a_1, a_2, \ldots, a_k$ be a sequence of objects and let c be a colouring that assigns one of the colours $\{1, 2, \ldots, n\}$ to every object such that no colour is assigned to two objects. Prove that the probability of the event $c(a_1) < c(a_2) < \ldots < c(a_k)$ equals $1/k!$.

10.6. **(−)** Let M be an $n \times n$ matrix and let k be a natural number. Describe an algorithm that finds the kth power of M using only $O(\log k)$ multiplications of two $n \times n$ matrices.

10.7. Prove the first equality in the proof of Lemma 10.2.1.

10.8. Prove Lemma 10.2.4 using Lemma 10.2.3.

10.9. Prove that the following problem is $\mathcal{NP}$-complete. Given a digraph D and an integer k, decide whether D has at least k disjoint cycles. Hint: use a reduction from the 3-dimensional matching problem. (Given three sets X^1, X^2, X^3 of the same cardinality n and a subset R of $X^1 \times X^2 \times X^3$, decide whether the elements of every X^i can be labelled $x_1^i, x_2^i, \ldots, x_n^i$ so that $(x_j^1, x_j^2, x_j^3) \in R$ for each $j = 1, 2, \ldots, n$. This problem is $\mathcal{NP}$-complete, see Gary and Johnson [303].) In the reduction you may utilize the gadget L given in Figure 10.6. We start from the digraph G on vertices $X^1 \cup X^2 \cup X^3$ and with no arcs. For each $(x, y, z) \in R$, we add L to G. Prove that the resulting digraph has $n + 2|R|$ cycles (all of which are 3-cycles) if and only if there exists the required labelling of the elements in X^1, X^2 and X^3 (A. Yeo, personal communication).

x

y

z

L

Figure 10.6 The gadget for Exercise 10.9.

10.10. **The directed dual of a plane directed multigraph is planar.** Show that, if D is a plane directed multigraph, then its directed dual D^* is also planar.

10.11. **Taking duals repeatedly.** Let D be a plane directed multigraph and let D^* be the directed dual of D. Show that the directed dual of D^* is isomorphic to the converse of D.

10.12. Let D be a plane directed multigraph and let D^* be the directed dual of D. Show that, if $(S, \bar{S})$ is a directed cut in D^*, then the corresponding arcs in D form a directed cycle.

10.13. Let $D = (V, A)$ be the plane digraph in Figure 10.1(a). Find two arcs in A whose deletion leaves an acyclic directed multigraph. Then check that contracting the corresponding two arcs in D^*, the directed dual of D, results in a strongly connected digraph.

10.14. **(−)** Show that the problem of finding a maximum size acyclic subdigraph of a directed multigraph $D = (V, A)$ is equivalent to that of finding an ordering $v_1, v_2, \ldots, v_n$ of V such that the number of arcs $v_i v_j$ with $i < j$ is maximum.

10.15. Prove Proposition 10.3.13.

10.16. Let D be an arbitrary directed multigraph. Prove that every minimum feedback arc set of D induces an acyclic subdigraph of D.

10.17. Show that the tournament T in Figure 10.7 has a minimum feedback arc set which induces a transitive subtournament of T.

Figure 10.7 A tournament T on 5 vertices.

10.18. Show that, if there exists a polynomial approximation algorithm with approximation guarantee $\rho(n)$ for the feedback arc set problem, then there also exists a polynomial approximation algorithm with approximation guarantee $\rho(n)$ for the feedback vertex set problem and vice versa.

10.19. $(-)$ Construct an infinite family of digraphs D such that $\nu_0(D) < \tau_0(D)$.

10.20. $(-)$ Prove that, if the functions $t_0(k)$ and $t_1(k)$ exist, then they are equal. Hint: apply Proposition 10.3.1.

10.21. For every $n \geq 3$, construct a digraph of minimum out-degree 2 not having two disjoint cycles.

10.22. Prove that every digraph D with $\delta^+(D) \geq 3$ has a pair of vertex-disjoint cycles. Hint: use Lemma 10.3.9 (Thomassen [700]).

10.23. Prove Corollary 10.3.6 using Theorem 10.3.5. Hint: first observe that every digraph D with $\delta^+(D) \geq k$ has at least $\frac{k}{64}$ vertex disjoint cycles. Remove the arcs of these and continue recursively.

10.24. $(+)$ Prove Lemma 10.4.6. Hint: use Menger's theorem.

10.25. $(-)$ Prove that the Balcer-Veinott algorithm (in Section 10.5) terminates with a cycle, whose length is the period of the input digraph.

10.26. $(-)$ Prove that a digraph D is even if and only if, for every assignment of weights 0 and 1 to its arcs, D contains a cycle of even weight.

10.27. Let D be a k-weak-double-cycle for some odd k. Prove that D has an odd number of cycles and that every arc is in an even number of cycles. Hint: use the recursive definition of a k-weak-double-cycle.

10.28. Let D be a k-weak-double-cycle for some odd k. Prove that D has an even cycle. Hint: assume that all cycles in D are odd and use Exercise 10.27 to obtain a contradiction.

10.29. Prove that given an arc e in a digraph D it is $\mathcal{NP}$-complete to decide whether D has an odd cycle through e (even cycle through e, respectively) (Thomassen [702]).

10.30. **Digraphs for which all cycles have the same parity.** Show that there is a polynomial algorithm to decide if the length of all cycles of a given digraph have the same parity.

10.31. **(−)** Give a short direct proof that the problem to verify whether a digraph D has cycle of length 0 modulo p, where both D and p form an input, is $\mathcal{NP}$-complete.

10.32. **(−)** Prove that the period of a strong non-bipartite digraph D with $\delta^0(D) \geq 3$ equals 1. Hint: use Theorem 10.6.7.

10.33. Prove Corollary 10.6.12.

10.34. **(−)** Prove the following generalization of Lemma 10.6.10. Let $D = (V, A, w)$ be a weighted digraph and let $k \geq 2$ be an integer. If there is a vertex colouring $c^* : V \rightarrow \{0, 1, \ldots, k-1\}$ of D such that for every $u \in V$ there is a $v \in N^+(u)$ with $c^*(v) \equiv c^*(u) + w(u, v) \pmod{k}$, then D has a cycle of weight 0 (mod k) (Alon and Linial [12]).

10.35. **Cycles modulo *k* in weighted digraphs.** Using the result of the previous exercise and the method of proof of Theorem 10.6.11 prove the following generalization of Theorem 10.6.11: Let $D = (V, A, w)$ be a weighted digraph and let $k \geq 2$ be an integer. If either (10.3) or (10.4) holds then D contains a cycle of weight 0 (mod k) (Alon and Linial [12]).

10.36. Prove that a 3-weak-double cycle is (k, p)-odd for every pair k, p such that $1 \leq k < p,\ p \geq 3$ (Galluccio and Loebl [301]).

10.37. Prove that it is $\mathcal{NP}$-complete to decide whether a strong digraph has two cycles whose lengths differ by one. Hint: reduce the hamiltonian cycle problem to this problem.

10.38. Construct for every k an infinite family of k-strong digraphs such that no digraph in the family has two cycles whose lengths differ by one.

10.39. **(+)** For $p \geq 3$, construct an infinite family $\mathcal{F}_p$ of strong semicomplete p-partite digraphs such that every digraph D in $\mathcal{F}_p$ contains a hamiltonian path, yet, a longest path of any strong orientation of D has $n-2$ vertices, where n is the order of D (Gutin, Tewes and Yeo [372]).

10.40. **(++)** Prove the following theorem. Let D be a strong semicomplete multipartite digraph of order n such that $D \neq \overleftrightarrow{K}_{n-1,1}$ and let l be the length of a longest path in D. Then D contains a strong spanning oriented subgraph with a path of length at least $l-2$ (Gutin, Tewes and Yeo [372]).

10.41. For every $p \geq 3$ construct a strong p-partite tournament T such that some vertex v of T is not contained in a k-cycle for some $3 \leq k \leq p$.

10.42. **(−)** Prove that if $n = gs$, then the s-regular round digraph of order n is of girth g.

10.43. Prove that Conjecture 10.9.4 implies Conjecture 10.9.1.

10.44. Let D be a strong digraph of minimum out-degree 2. Prove that D contains a vertex x such that $D - x$ is strong. Hint: consider D', a maximal strong proper subdigraph of D. Prove that D' contains all vertices of D but one.

10.45. For every $k \geq 2$, construct strong digraphs on n vertices such that the number of arcs is $\frac{n^2+(3-2k)n+k^2-k}{2} - 1$ and the shortest cycle has length $k+1$.

10.46. Derive Corollary 10.10.3 from Lemma 10.10.2 and Marcus' theorem (Theorem 10.10.1).

10.47. Prove that $\vec{C}_5 \times \vec{C}_{7+10k}$, $k \geq 0$, has no cycle of length $35+50k$ or $34+50k$. Hint: apply Theorem 10.10.5.

11. Generalizations of Digraphs

In this chapter, several results proved for digraphs are extended to edge-coloured graphs, arc-coloured digraphs and hypertournaments. We will see that some results remain the same with respect to their formulation, but their proofs become much more involved. Other results do not hold any more. This gives an additional insight to the theory of digraphs. In particular, we can more clearly see which properties of digraphs allow us to obtain various results on them.

In Section 11.1 we study properly coloured trails (i.e. trails whose consecutive edges differ in colour) in edge-coloured undirected multigraphs. In Subsection 11.1.1 we prove Kotzig's characterization of edge-coloured multigraphs containing properly coloured (PC) Euler trails and Pevzner's theorem that shows how to generate all PC Euler trails of an edge-coloured multigraph from some initial one. Yeo's theorem on PC cycles in edge-coloured graphs, which in a sense characterizes edge-coloured graphs not having PC cycles, is proved in Subsection 11.1.2. Subsection 11.1.3 is devoted to generalizations of strong connectivity to edge-coloured multigraphs. We consider various interesting results on hamiltonian and longest PC paths and cycles in 2-edge-coloured multigraphs in Subsection 11.1.4. Many of these results can be easily obtained from the corresponding results on digraphs using some transformations also described in this subsection. The characterization of 2-edge-coloured complete graphs containing hamiltonian PC cycles, due to Bankfalvi and Bankfalvi, is given in Subsection 11.1.5. There we prove Saad's theorem characterizing longest PC cycles in 2-edge-coloured complete graphs. PC paths and cycles in c-edge-coloured complete graphs, $c \geq 3$, are studied in Subsections 11.1.6 and 11.1.7; along with results on the topic, we describe several interesting open problems.

The somewhat surprising result, due to Gutin, Sudakov and Yeo, that the problem of checking the existence of a PC directed cycle in a 2-arc-coloured digraph is $\mathcal{NP}$-complete is proved in Section 11.2. There we also consider the PC Euler trail problem for arc-coloured directed multigraphs; the complexity of this problem remains unknown. We generalize the classic theorems on tournaments, Rédei's theorem, Camion's theorem and Landau's theorem, to hypertournaments in Section 11.3. Despite the existence of elegant characterization of hamiltonian hypertournaments proved by Gutin and Yeo, it turns out that the hamiltonian cycle problem for hypertournaments, in general, is $\mathcal{NP}$-complete. We finish this chapter by a short overview of an application of alternating Hamilton cycles in 2-edge-coloured multigraphs to genetics (see Section 11.4).

11.1 Properly Coloured Trails in Edge-Coloured Multigraphs

In this section we consider **edge-coloured multigraphs**, i.e. undirected multigraphs such that each edge has a colour and no two parallel (i.e. joining the same pair of vertices) edges have the same colour. If the number of colours is restricted by an integer c, we speak about ***c*-edge-coloured multigraphs.** We usually use the integers $1, 2, \ldots, c$ to denote the colours in c-edge-coloured multigraphs. In case $c = 2$, we also use the names **red** and **blue** for colours 1 and 2, respectively. The **red subgraph** (**blue subgraph**, respectively) of a 2-edge-coloured multigraph G consists of the vertices of G and all red (blue, respectively) edges of G.

Let G be a c-edge-coloured multigraph ($c \geq 2$). A trail T in G is **properly coloured (PC)** if no two consecutive edges of T have the same colour. A **PC *m*-path-cycle subgraph** $\mathcal{F}_m$ of G is a union of m PC paths and a number of PC cycles in G, all vertex-disjoint. When $m = 0$, we will call $\mathcal{F}_0$ a **PC cycle subgraph**. If G is 2-edge-coloured, then we call a properly coloured trail in G **alternating** . To see that the alternating path and cycle structure of 2-edge-coloured multigraphs generalizes the path and cycle structure of directed multigraphs, we consider the following simple transformation attributed to Häggkvist in [548]; see Figure 11.1. Let D be a directed multigraph. Replace each arc xy of D by two (unoriented) edges xz_{xy} and $z_{xy}y$ by adding a new vertex z_{xy} and then colour the edge xz_{xy} red and the edge $z_{xy}y$ blue. Let G be the 2-edge-coloured graph obtained in this way. It is easy to see that each alternating cycle in G corresponds to a directed cycle in D and vice versa. Hence, in particular, we obtain the following proposition.

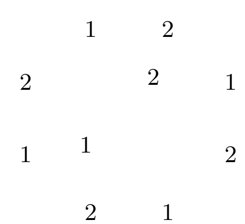


Figure 11.1 Häggkvist's transformation.

Proposition 11.1.1 *The following problems on paths and cycles in 2-edge-coloured graphs are $\mathcal{NP}$-complete:*

(a) The alternating Hamilton cycle problem.
(b) The problem to find an alternating cycle through a prescribed pair of vertices.

Proof: Exercise 11.1. □

Clearly, a directed path in D corresponds to an alternating path in G as well. Thus, we may conclude that the alternating path and cycle structure in 2-edge-coloured graphs generalizes the (directed) path and cycle structure of directed multigraphs. In fact, we will see, in this section, that the former is certainly more complicated than the latter. Still, several methods and results obtained for directed multigraphs can be adapted to edge-coloured multigraphs.

Petersen's famous paper [603] seems to be the first place where one can find applications of PC trails (cf. [575]). Besides a number of applications in graph theory and algorithms (cf. the papers [738, p. 58] by Woodall and [386] by Häggkvist), the concept of PC trails and its special cases, PC paths and cycles, appears in various other fields including genetics (cf. the papers [200, 201] by Dorninger, [202] by Dorninger and Timischl and [606] by Pevzner; see also the last section of this chapter) and social sciences (cf. the paper [156] by Chow, Manoussakis, Megalakaki, Spyratos and Tuza).

Let G be a c-edge-coloured multigraph. The $\boldsymbol{j}$th **degree** of v, $d_j(v)$, is the number of edges of colour j incident to v ($1 \leq j \leq c$). The **maximum monochromatic degree** of G is defined by

$$\Delta_{mon}(G) = \max\{d_j(v) : \; v \in V(G), \; j = 1, 2, \ldots, c\}.$$

The colour of an edge e in G will be denoted by $\chi(e)$. Let X and Y be two sets of the vertices of G. Then XY denotes the set of all edges having one end vertex in X and the other in Y and $\chi(XY)$ stands for the set of colours of edges in XY. In case all the edges in XY have the same colour, say i, we write $\chi(XY) = i$.

Edge-coloured multigraphs G and H are **colour-isomorphic** if there exists an isomorphism $f : \; V(G) \rightarrow V(H)$ such that $\chi(xy) = \chi(f(x)f(y))$ for every pair x, y of distinct vertices of G. Let $T = p_1 p_2 \ldots p_l$ be a trail in G. Then, the trail $p_l p_{l-1} \ldots p_1$, called the **reverse of $\boldsymbol{T}$**, will be denoted by T^{rev}. Also, if $l \geq 2$, then

$$\chi_{end}(T) = \chi(p_{l-1} p_l), \; \chi_{start}(T) = \chi(p_1 p_2).$$

Let G be a 2-edge-coloured multigraph of even order n; G is **alternating-pancyclic** if G has an alternating cycle of length $2k$ for every $k = 2, 3, 4, \ldots, n/2$; G is **vertex alternating-pancyclic** if, for every vertex $v \in V(G)$ and every integer $k \in \{2, 3, 4, \ldots, n/2\}$, G contains an alternating cycle through v of length $2k$.

11.1.1 Properly Coloured Euler Trails

In [502], Kotzig proved the following characterization of edge-coloured multigraphs which contain properly coloured Euler trails.

Theorem 11.1.2 (Kotzig) *[502] An edge-coloured multigraph G has a properly coloured Euler trail if and only if G is connected, each vertex of G is of even degree, and for every vertex x and every colour i, $d_i(x) \leq \sum_{j \neq i} d_j(x)$.*

Proof: Obviously, the conditions above are necessary.

Suppose G satisfies the conditions of Theorem 11.1.2. We will first show that, for every vertex x, the edges of G incident to x can be partitioned into disjoint pairs of distinct edges so that the colours of the edges in each pair are different. This guarantees that each time we enter x through an edge e we can leave it through the edge f forming one of the above pairs with e. (We will denote f by $\mathrm{match}_x(e)$.)

In order to determine this partition, for each vertex x we define an auxiliary graph G_x so that the vertices of G_x are the edges incident to x. Two vertices are connected in G_x if their corresponding edges in G have different colours. It is easy to see that the above partition exists if and only if each G_x has a perfect matching. It remains to prove that each G_x indeed has a perfect matching.

Observe that each G_x is a complete multipartite graph with partite sets of some cardinalities $n_1, n_2, \ldots, n_t$ satisfying the following inequality:

$$n_i \leq \sum_{j \neq i} n_j \tag{11.1}$$

for every $i = 1, 2, \ldots, t$. Choose an edge b between two largest partite sets of G_x. Delete the vertices of b from G_x. Clearly, the partite sets of the obtained graph satisfy the inequality (11.1). This means we can proceed by choosing another edge as above. This process will clearly produce a perfect matching of G_x. (One could easily arrive at the same conclusion using Tutte's theorem on perfect matchings in multigraphs, see e.g. the book [127] by Bondy and Murty.)

Fix a perfect matching

$$\{(e, \mathrm{match}_x(e)) : \ e \in V(G_x)\}$$

in G_x for every x in G. We call a PC trail Q of G an ***M*-trail** if $\mathrm{match}_x(e) \in E(Q)$ for every $x \in V(Q)$ and every $e \in E(Q)$ incident to x. Clearly, every M-trail is closed. In the obvious way (see the construction of R below), one can build an M-trail. Let T be an M-trail of G with maximum number of edges. Assume that $E(T) \neq E(G)$. Since G is connected, $G - E(T)$ contains an edge e_1 incident to a vertex x_1 in T. We construct a trail R in $G - E(T)$ as follows: $x_1, e_1, x_2, e_2 = \mathrm{match}_{x_2}(e_1), x_3, e_3 = \mathrm{match}_{x_3}(e_2), x_4, \ldots, x_k, e_k =$

$\text{match}_{x_k}(e_{k-1}), x_{k+1}$, where $e_i = x_i x_{i+1}$ for every $i = 1, 2, \ldots, k$, $x_{k+1} = x_1$ and $e_1 = \text{match}_{x_1}(e_k)$. Observe that T and R are edge-disjoint by the definition of M-trails.

Since x_1 is in T, we can write down T as $\ldots f, x_1, g, \ldots$. Assume, without loss of generality, that $\chi(f) = 1$, $\chi(g) = 2$ and $\chi(e_1) \neq 1$. If $\chi(e_k) \neq 2$, then replace the appearance of x_1 between f and g in T with the trail R obtaining, as a result, an M-trail of G with more edges than T, a contradiction. If $\chi(e_k) = 2$, then replace the appearance of x_1 between f and g in T with the trail R^{rev} obtaining, as a result, an M-trail (observe that $\chi(e_1) > 2$) of G with more edges than T, a contradiction.

Thus, $E(T) = E(G)$, i.e., T is eulerian. $\square$

Benkouar, Manoussakis, Paschos and Saad [103] described an $O(n^2 \log n)$-algorithm for finding a properly coloured eulerian trail in an edge-coloured multigraph G on n vertices that satisfies the conditions of Theorem 11.1.2. Pevzner [606] suggested the following simple and practical algorithm to find a PC eulerian trail in G. Let $P = x_1 x_2 \ldots x_k$ be a PC trail. A colour χ' is **critical with respect to $\boldsymbol{P}$** if it is the most frequent colour $\chi' \neq \chi(x_{k-1}x_k)$ of edges with one end at x_k and the other in $V(G) - V(P)$. Pevzner's algorithm for an edge-coloured multigraph G satisfying Theorem 11.1.2 proceeds as follows. Let x_1 be an arbitrary vertex in G. Put $P_1 = x_1$ and build up $P_k = x_1 x_2 \ldots x_k$ by adding an arbitrary edge $x_k x_{k+1}$ of colour $\chi(x_1 x_2)$, if this colour is critical with respect to P, or of any critical colour with respect to P, otherwise. We stop when no critical colour edge is available. Pevzner [606] proved that this simple algorithm always produces a PC eulerian trail if one exists (Exercise 11.3).

Using the above transformation by Häggkvist, one can readily obtain the following result (see a direct proof of it in Theorem 1.6.3):

Corollary 11.1.3 *A directed multigraph D is eulerian if and only if D is connected and $d^+(x) = d^-(x)$ for every vertex x in D.* $\square$

Fleischner, Sabidussi and Wegner [242] and Pevzner [606] independently investigated what operations can be used to transform an alternating eulerian trail of a 2-edge-coloured multigraph to any other one. Interestingly enough, while the first paper has had a pure theoretical motivation, in the second paper, the author showed some applications of alternating eulerian trails, in general, and those transformations, in particular, to an important $\mathcal{NP}$-hard problem in genetics. We discuss below only the characterization of the transformations in [606].

Let G be 2-edge-coloured multigraph containing an alternating eulerian trail. In the rest of this subsection, for the sake of convenience, we consider alternating trails as ordered sets of edges. Let $T = T_1T_2T_3T_4T_5$ be an alternating trail (where T_i are fragments of T viewed as subsets of $E(G)$). The transformation $T \rightarrow T^* = T_1T_4T_3T_2T_5$ is called an **order exchange** if T^* is an alternating trail. Let $T = T_1T_2T_3$ be an alternating trail. The transformation $T \to T^* = T_1T_2^{rev}T_3$ is an **order reflection**, if T^* is an alternating trail. Let X and Y be a pair of alternating trails in G. The number of vertices in the largest common subtrail of X and Y is the **index** $\mathrm{ind}(X, Y)$ of X and Y.

Theorem 11.1.4 (Pevzner) *[606] Every pair of alternating eulerian trails X and Y in a 2-edge-coloured multigraph can be transformed into each other by means of a sequence of order transformations (exchanges and reflections).*

Proof: In the set of alternating eulerian trails $\mathcal{T}$, which can be obtained from X by means of a sequence of order transformations, choose an element, $X^* = x_1x_2 \dots x_q$, having the largest common subtrail with $Y = y_1y_2 \dots y_q$. (Clearly, $x_1 = x_q$ and $y_1 = y_q$.) Let us assume that $\mathrm{ind}(X^*, Y) = \ell < q$. Due to the fact that both X^* and Y are closed, without loss of generality, we may assume that $x_i = y_i$ for $1 \le i \le \ell$.

Let $e_1 = x_\ell x_{\ell+1}$ and $e_2 = y_\ell y_{\ell+1}$. Clearly, $\chi(e_1) = \chi(e_2)$. Since X^* is eulerian, X^* contains e_2. There are two possibilities depending on the direction in which we traverse the edge e_2 in X^* (going from x_1 to x_q).

Case 1: In X^* the edge e_2 is traversed from $y_{\ell+1}$ to y_ℓ. In this case,

$$X^* = x_1 \dots x_\ell x_{\ell+1} \dots y_{\ell+1} y_\ell \dots x_q.$$

Let $T_1 = x_1 \dots x_\ell$, $T_2 = x_\ell x_{\ell+1} \dots y_{\ell+1} y_\ell$ and $T_3 = y_\ell \dots x_q$. Since $\chi(e_1) = \chi(e_2)$, the transformation $X^* \to X^{**} = T_1T_2^{rev}T_3$ is an order reflection. But $X^{**} \in \mathcal{T}$ and $\mathrm{ind}(X^{**}, Y) > \mathrm{ind}(X^*, Y)$, a contradiction to the choice of X^*.

Case 2: In X^* the edge e_2 is traversed from y_ℓ to $y_{\ell+1}$. In this case,

$$X^* = x_1 \dots x_\ell x_{\ell+1} \dots (x_p = y_\ell)(x_{p+1} = y_{\ell+1}) \dots x_q.$$

Let $X_1 = x_1 \dots x_\ell$, $X_2 = x_\ell x_{\ell+1} \dots x_p$ and $X_3 = x_p x_{p+1} \dots x_q$.

Claim. *The trail X_3 contains a vertex x_j $(j > p)$ belonging to X_2.*

Proof of Claim: Let $i > \ell$ be the minimum number fulfilling the following condition: vertex y_i of the trail Y is in X_2. The existence of such an i follows from the fact that Y contains the edge $e_1 = y_{t-1}y_t$ for some $t > \ell$. Due to the minimality of i the edge $y_{i-1}y_i$ does not belong to X_2. Condition $i > l$ implies that this edge is not in X_1. Hence, this edge is in X_3 implying that X_2 and X_3 have a common vertex. The claim is proved.

Due to the claim, the trail X^* can now be rewritten as

$$X^* = x_1 \dots x_\ell x_{\ell+1} \dots (x_k = x_j) \dots (x_p = x_\ell)(x_{p+1} = y_{\ell+1}) \dots x_j \dots x_q.$$

Let $T_1 = x_1 \dots x_\ell$, $T_2 = x_\ell x_{\ell+1} \dots x_k$, $T_3 = x_k \dots x_p$, $T_4 = x_p \dots x_j$, and $T_5 = x_j \dots x_q$. Consider the edges $f_1 = x_{k-1}x_k$ and $f_2 = x_{j-1}x_j$. If $\chi(f_1) = \chi(f_2)$, then $\chi(f_2) \ne \chi(x_kx_{k+1})$ and $X^{**} = T_1T_4T_3T_2T_5$ is the alternating trail obtained from X^* by means of some order exchange. Clearly, $\mathrm{ind}(X^{**}, Y) > \mathrm{ind}(X^*, Y)$, a contradiction to the choice of X^*.

If $\chi(f_1) \neq \chi(f_2)$, then $X^{**} = T_1T_4T_2^{rev}T_3^{rev}T_5$ is an alternating trail. This trail is obtained from X^* by means of two order reflections:

$$\begin{aligned} & T_1T_2T_3T_4T_5 \rightarrow T_1T_2(T_3T_4)^{rev}T_5 \\ = {} & T_1T_2T_4^{rev}T_3^{rev}T_5 \rightarrow T_1(T_2T_4^{rev})^{rev}T_3^{rev}T_5 \\ = {} & T_1T_4T_2^{rev}T_3^{rev}T_5. \end{aligned}$$

Clearly, $\mathrm{ind}(X^{**}, Y) > \mathrm{ind}(X^*, Y)$, a contradiction to the choice of X^*. □

11.1.2 Properly Coloured Cycles

Using Häggkvist's transformation, we see that the problem to check whether a c-edge-coloured graph has a properly coloured cycle is more general (even for $c = 2$) than the simple problem to verify whether a digraph contains a directed cycle (see Proposition 1.4.2 and the remark afterwards). In the rest of this subsection we consider the following:

Problem 11.1.5 *Given a c-edge-coloured graph G, check whether G contains a properly coloured cycle.*

Grossman and Häggkvist [335] were the first to study this problem. They proved Theorem 11.1.6 below in the case $c = 2$. Yeo [743] showed Theorem 11.1.6 for every $c \geq 2$.

Let G be a c-edge-coloured graph and let x, y be arbitrary distinct vertices of G. We will use the following additional notation:

$$\begin{aligned} \chi_{end}(x, y) &= \{\chi_{end}(P) : P \text{ is a PC } (x, y)\text{-path}\}; \\ \chi_{start}(x, y) &= \{\chi_{start}(P) : P \text{ is a PC } (x, y)\text{-path}\}. \end{aligned}$$

Theorem 11.1.6 (Yeo) *[743] Let G be a c-edge-coloured graph, $c \geq 2$, with no PC cycle. Then, G has a vertex $z \in V(G)$ such that no connected component of $G - z$ is joined to z with edges of more than one colour.*

Proof: Let $G = (V, E)$ be an edge-coloured graph with no PC cycle. Let $p_1 \in V$ be arbitrary. Set $S = \{p_1\} \cup \{s \in V - \{p_1\} : |\chi_{end}(p_1, s)| = 1\}$. Now let $P = p_1p_2 \dots p_l$ $(l \geq 1)$ be a PC path of maximum length such that $p_l \in S$, and set $T_k = \{t \in V - \{p_l\} : k \in \chi_{start}(p_l, t)\}$ for every colour $k \in \{1, 2, \dots, c\}$. If $l = 1$, then let C^* be the set of all colours in G, and if $l \geq 2$ then let C^* be the set of all colours in G except $\chi_{end}(P)$. We will prove this theorem in three steps.

(1) $V(P) \cap T_k = \emptyset$ for all $k \in C^*$.

If $l = 1$ then this statement is trivially true (since $p_\ell \notin T_k$), so assume that $l \geq 2$ and that the statement is false, which implies that there is a PC (p_l, p_i)-path $R = p_l r_1 r_2 \ldots r_{m-1} r_m p_i$ $(m \geq 0)$ with $\chi_{start}(R) = k$, $i \in \{1, 2, \ldots, l-1\}$ and $V(R) \cap V(P) = \{p_i, p_l\}$. Clearly $\chi(p_i p_{i+1}) = \chi_{end}(R)$, since otherwise we would obtain the PC cycle $p_i p_{i+1} \ldots p_l r_1 r_2 \ldots r_{m-1} r_m p_i$. This implies that $Q = p_1 p_2 \ldots p_i r_m r_{m-1} \ldots r_1 p_l$ is a PC (p_1, p_l)-path, with $\chi_{end}(Q) = \chi_{start}(R) = k \neq \chi_{end}(P)$. We have thus shown that $\{\chi_{end}(Q), \chi_{end}(P)\} \subseteq \chi_{end}(p_1, p_l)$, which implies that $|\chi_{end}(p_1, p_l)| \geq 2$. Therefore $p_l \notin S$, contradicting the definition of P.

(2) If $xy \in E$, $x \in T_k$, $y \notin T_k$ for some $k \in C^*$, then $y = p_l$ and $\chi(xy) = k$.

First we claim that there is a PC (p_l, x)-path R with $\chi_{end}(R) \neq \chi(xy)$ and $\chi_{start}(R) = k$.

By the definition of T_k, there is a PC (p_l, x)-path Q with $\chi_{start}(Q) = k$. If $\chi_{end}(Q) \neq \chi(xy)$ we set $R := Q$, so assume that $\chi_{end}(Q) = \chi(xy)$. By (1), PQ is a PC (p_1, x)-path, which is longer than P. This implies that $x \notin S$, so $|\chi_{end}(p_1, x)| \geq 2$. Thus there is a PC (p_1, x)-path L with $\chi_{end}(L) \neq \chi(xy)$. Let $w \in (V(L) \cap V(P \cup Q)) - \{x\}$ be chosen so that $V(L[w, x]) \cap V(P \cup Q) = \{w, x\}$.

Suppose that $w \in V(P) - \{p_l\}$. Then $QL^{rev}[x, w]$ is a PC (p_l, w)-path whose first edge has colour k. This implies that $w \in T_k$, which contradicts (1). Hence $w \in V(Q)$ and $\chi_{start}(Q[w, x]) = \chi_{start}(L[w, x])$, since otherwise $Q[w, x]L^{rev}[x, w]$ is a PC cycle. This implies that $R = Q[p_l, w]L[w, x]$ is a PC (p_l, x)-path with $\chi_{start}(R) = k$ and $\chi_{end}(R) \neq \chi(xy)$. Thus, the claim is proved.

Let R be as guaranteed by the claim. If $y \neq p_l$, then Ry is a PC (p_l, y)-path with $\chi_{start}(Ry) = k$, which contradicts the assumption that $y \notin T_k$. Thus $y = p_l$. If $\chi(xy) \neq k$, then we obtain the PC cycle Ry, which is also a contradiction. Thus $\chi(xy) = k$.

(3) No connected component of $G - p_l$ is joined to p_l with edges of more than one colour.

Assume that the statement is false, and let $p_l x$ and $p_l y$ be a pair of distinct edges in G such that x and y belong to the same connected component of $G - p_l$ and $\chi(p_l x) \neq \chi(p_l y)$. Assume without loss of generality that $\chi(p_l x) \in C^*$ (otherwise interchange x and y). In $G - p_l$ there is a (not necessarily PC) path $R = r_1 r_2 \ldots r_m$ $(m \geq 2)$ between $x = r_1$ and $y = r_m$. If $y \in T_{\chi(p_l x)}$, then since $p_l \notin T_{\chi(p_l x)}$, (2) implies that $\chi(p_l y) = \chi(p_l x)$, which is a contradiction. Therefore $y \notin T_{\chi(p_l x)}$, which together with $x \in T_{\chi(p_l x)}$ implies that there exists an i $(1 \leq i \leq m-1)$ such that $r_i \in T_{\chi(p_l x)}$ and $r_{i+1} \notin T_{\chi(p_l x)}$. This, however, contradicts (2), since $r_i r_{i+1} \in E$ but $p_l \neq r_{i+1}$. $\square$

One can see that Theorem 11.1.6 actually solves Problem 11.1.5. Indeed, if G has no vertex z such that all edges from z to any of the components of $G - z$ are of the same colour, then Theorem 11.1.6 implies that G contains a PC cycle. If G has such a vertex z, we may consider only $G-z$ or its components (if $G-z$ is disconnected), since no PC cycle can contain z. (See also Figure 11.2.) This leads to an obvious polynomial recursive algorithm (for a vertex $x \in G$, the components of $G-x$ can be found in $O(|V(G)|+|E(G)|)$ time).

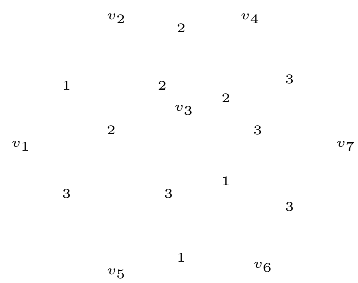


Figure 11.2 An edge-coloured graph with no PC cycle. To see this, it suffices to check that every vertex v_i has only edges of the same colour to $\{v_1, \ldots, v_{i-1}\}$.

Interesting corollaries of Theorem 11.1.6 are given as exercises (Exercises 11.7,11.8) in this chapter. Theorem 11.1.6 also implies:

Corollary 11.1.7 *[100, 501, 534] There does not exist a bridgeless graph that contains a unique perfect matching.*

Proof: Exercise 11.6. □

Another possibility to solve Problem 11.1.5 is to use the following construction by Bang-Jensen and Gutin [61] illustrated in Figure 11.3. Here, we can actually find a PC cycle subgraph with maximum number of vertices of a c-edge-coloured multigraph in polynomial time. This result is very useful, as a starting point, for a number of problems on PC cycles and paths.

Let G be an arbitrary c-edge-coloured multigraph (with colours 1,2,…,c). For each vertex v of G we form the a graph H_v with vertex set $V(H_v) = \{v_1, \ldots, v_{2c-2}\}$ and v_i is adjacent to v_j $(i<j)$ in H_v if and only if either both $i, j \in \{1, \ldots, c\}$ or $i \in \{1, \ldots, c\}, j \in \{c+1, \ldots, 2c-2\}$. Construct a new graph $R = R(G)$ from the disjoint union of the graphs H_v $(v \in V(G))$ as follows. An edge $v_i u_j$ is in R if and only if $i = j = \chi_G(vu)$. Let the edges of R of the form $v_i v_j$ where both $i, j \in \{1, \ldots, c\}$ have the weight 0 and all other edges have the weight 1. Then, a maximum weight perfect matching in R corresponds to a PC cycle subgraph F of G with maximum number of vertices. To see this, it suffices to observe that for any perfect matching of R and any H_v (corresponding to one vertex v of G), all but two of the vertices $v_1, v_2, \ldots, v_c$ will be matched to vertices within H_v and with index at least $c+1$. Hence if the edge between the two remaining vertices in H_v is not in the matching, then in G this corresponds to v being on a PC cycle and vice

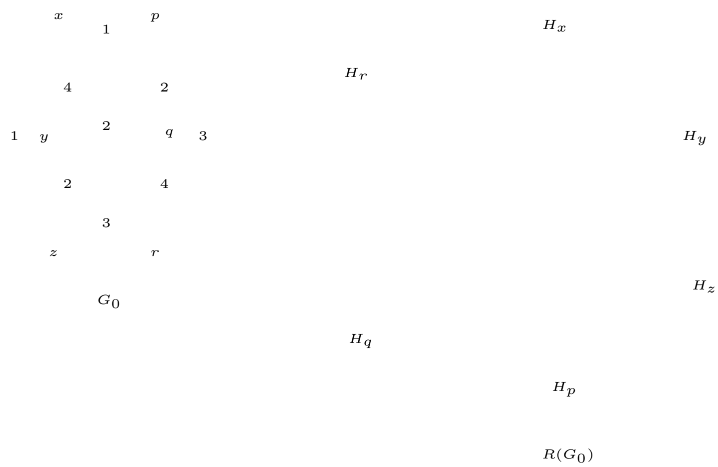


Figure 11.3 The left figure shows a 4-edge-coloured graph G_0. The right figure depicts the construction from [61] for G_0. The big circle in every H_v has $c = 4$ vertices and the small one $c - 2 = 2$ vertices. Only edges of $R(G_0)$ between the graphs H_v are shown. The fat edges are part of a maximum weight perfect matching of $R(G_0)$, they correspond to PC cycles $abca$ and $defd$ of G_0.

versa. This construction implies the existence of a polynomial algorithm for finding F since a maximum weight perfect matching in a weighted graph on p vertices can be found in time $O(p^3)$ (cf. the book by Papadimitriou and Steiglitz [600, Chapter 11]).

Sometimes, one needs to find a maximum PC 1-path-cycle subgraph of a c-edge-coloured multigraph G. We can easily transform the last problem to the maximum PC cycle subgraph problem as follows. Add an extra-vertex x to G and join x to every vertex of G by two edges of colour $c+1$ and $c+2$ respectively (new colours). Clearly, a maximum PC cycle subgraph of the new multigraph corresponds to a maximum PC 1-path-cycle subgraph of G. We formulate the obtained results as a theorem:

Theorem 11.1.8 *[61] One can construct a maximum PC cycle subgraph and a maximum PC 1-path-cycle subgraph, respectively, in a c-edge-coloured multigraph G on n vertices in time $O((cn)^3)$.* □

Let G be a c-edge-coloured multigraph and let x be a vertex in G. Consider the following modification $R'(G)$ of $R(G)$: change the weight of the edges $x_i x_j$, $1 \le i < j \le c$ from 0 to $-\infty$. There is a perfect matching of finite weight in $R'(G)$ if and only if G has a PC cycle through x. This implies the next proposition.

Proposition 11.1.9 *Given a c-edge-coloured multigraph G and a vertex x of G, one can verify whether G has a PC cycle through x in polynomial time.* □

This proposition is in sharp contrast with Proposition 11.1.1 (b).

11.1.3 Connectivity of Edge-Coloured Multigraphs

Strong connectivity plays a central role in the study of digraphs. Hence, it is natural to try to obtain some extensions of strong connectivity to edge-coloured graphs. Such extensions have been introduced and studied in the literature. In fact, there are two useful extensions of strong connectivity: one of them generalizes the usual definition of strong connectivity that refers to paths between pairs of vertices and the other extends the definition of cyclic connectivity in digraphs (see Exercise 1.30), which is equivalent to strong connectivity (for digraphs). However, for edge-coloured graphs these two generalizations are not equivalent any more.

In this subsection we study the above-mentioned generalizations of strong connectivity. We restrict ourselves to 2-edge-coloured multigraphs since we will later use connectivity results only for 2-edge-coloured graphs. Also this will make our arguments easier to follow. However, the reader should bear in mind that some of the results below could be generalized to c-edge-coloured multigraphs, $c \geq 2$.

The following notion of colour-connectivity was introduced by Saad [648] (he used another name for this notion). Let G be a 2-edge-coloured multigraph. A pair of vertices x, y of G are **colour-connected** if there exist alternating (x, y)-paths P and Q such that $\chi_{start}(P) \neq \chi_{start}(Q)$ and $\chi_{end}(P) \neq \chi_{end}(Q)$. (Notice that P and Q are paths, not trails.) We define a vertex x to be colour-connected to itself. We say that G is **colour-connected** if every pair of vertices of G is colour-connected.

Clearly, every alternating cycle is a colour-connected graph. This indicates that colour-connectivity may be useful for solving alternating cycle problems. We can use colour-connectivity more effectively if we know that this is an equivalence relation on the vertices of the graph under consideration. This leads us to the following definition: a 2-edge-coloured multigraph G is **convenient** if colour-connectivity is an equivalence relation on the vertices of G. Unfortunately, there are non-convenient multigraphs. Consider the graph H in Figure 11.4. It is easy to check that the vertices x and y are colour-connected to u, but x and y are not colour-connected in H.

The following proposition can be easily proved using only the definition of colour-connectivity. The following result due to Bang-Jensen and Gutin provides another way of checking colour-connectivity. Its proof is left to the reader as Exercise 11.9.

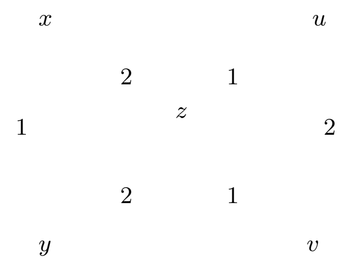


Figure 11.4 A non-convenient 2-edge-coloured graph.

Proposition 11.1.10 *[64] A pair of vertices, x_1, x_2, in a 2-edge-coloured multigraph G is colour-connected if and only if G has four (not necessarily distinct) alternating (x_1, x_2)-paths, P_1, P_2, Q_1, Q_2, such that $\chi_{start}(P_i) = \chi_{end}(Q_i) = i$, $i = 1, 2$.* □

Let G be a graph with matching M. A path P in G is **augmenting with respect to M** if, for any pair of adjacent edges in P, exactly one of them belongs to M, and the first and last edges of P do not belong to M. Let G be a 2-edge-coloured multigraph. The following proposition by Bang-Jensen and Gutin shows that we can check whether a pair of vertices of G are colour-connected in polynomial time.

Proposition 11.1.11 *[64] Let $G = (V, E)$ be a connected 2-edge-coloured multigraph and let x and y be distinct vertices of G. For each choice of $i, j \in \{1, 2\}$ we can find an alternating (x, y)-path P with $\chi_{start}(P) = i$ and $\chi_{end}(P) = j$ in time $O(|E|)$ (if one exists).*

Proof: Let $W = V - \{x, y\}$. Create an uncoloured graph $G_{xy,ij}$ in the following way: $V(G_{xy,ij}) = \{x, y\} \cup W^1 \cup W^2$, where $W^r = \{z^r : z \in W\}$ for $r = 1, 2$, $E(G_{xy,ij}) = \{xz^i : z \in W \text{and } \chi(xz) = i\} \cup \{z^j y : z \in W \text{and } \chi(zy) = j\} \cup \{u^k v^k : u, v \in W \text{ and } \chi(uv) = k\} \cup \{z^1 z^2 : z \in W\}$.

The reader can easily verify that G has the desired path if and only if there exists an augmenting path in G_{xy} with respect to the matching $M = \{z^1 z^2 : z \in W\}$. The latter can be checked, and a path constructed if one exists, in time $O(|E|)$. From any augmenting path P in G_{xy} we can obtain the desired path in G, simply by contracting those edges of M which are on P. □

Since colour-connectivity is not an equivalence relation on the vertices of every 2-edge-coloured multigraph, another notion of connectivity, cyclic connectivity, introduced by Bang-Jensen and Gutin [61], is sometimes more useful. Let $\mathcal{P} = \{H_1, \ldots, H_p\}$ be a set of subgraphs of a multigraph G. The **intersection graph** $\Omega(\mathcal{P})$ of $\mathcal{P}$ has the vertex set $\mathcal{P}$ and the edge set $\{H_i H_j : V(H_i) \cap V(H_j) \neq \emptyset, 1 \leq i < j \leq p\}$. A pair, x, y, of vertices in a 2-edge-coloured multigraph H is **cyclic connected** if H has a collection of alternating cycles $\mathcal{P} = \{C_1, \ldots, C_p\}$ such that x and y belong to some cycles in $\mathcal{P}$ and $\Omega(\mathcal{P})$ is a connected graph.

We formulate the following trivial but useful observation as a proposition.

Proposition 11.1.12 *Cyclic connectivity is an equivalence relation on the vertices of a 2-edge-coloured multigraph.* □

This proposition allows us to consider cyclic connectivity components similar to strong connectivity components of digraphs.

The following theorem due to A. Yeo (private communication, 1998) shows that cyclic connectivity between a pair of vertices can be checked in polynomial time.

Theorem 11.1.13 *For a pair x, y of vertices in a 2-edge-coloured multigraph $H = (V, E)$, one can check whether x and y are cyclic connected in time $O(|E|(|V| + |E|))$.*

Proof: By Proposition 11.1.11, in time $O(|E|)$, one can check whether H has an alternating cycle through a fixed edge $e \in E$. This implies that, in time $O(|V||E|)$, one can verify whether H has an alternating cycle through a fixed vertex $v \in V$.

We now describe a polynomial algorithm to check whether x and y are cyclic connected. Our algorithm starts by initiating $X := \{x\}$. Then, we find an alternating cycle through x; let X' be the vertices except for x of such a cycle. If $y \in X'$, then we are done. Otherwise, delete the vertices of X from H, set $X := X'$ and $X' := \emptyset$. Then, for each edge e with one end-vertex in X and the other not in X find an alternating cycle through the edge (if one exists). Now append all the vertices, except for those in X, in the cycles we have found to X' and check whether $y \in X'$. If $y \notin X'$, then we continue as above. We proceed until either $y \in X'$ or there is no alternating cycle through any edge with one end-vertex in X and the other not in X. Clearly, if $y \in X'$ at some stage, then x and y are cyclic connected, otherwise they are not.

The total time required for the operation of deletion is $O(|V||E|)$. By the complexity bounds above and the fact that we may want to find an alternating cycle through an edge at most once, the complexity of the described algorithm is $O(|E|(|V| + |E|))$. □

The following theorem by Bang-Jensen and Gutin shows that cyclic connectivity implies colour-connectivity.

Theorem 11.1.14 *[64] If a pair, x, y, of vertices in a 2-edge-coloured multigraph G is cyclic connected, then x and y are colour-connected.*

Proof: If x and y belong to a common alternating cycle, then they are colour-connected. So, suppose that this is not the case.

Since x and y are cyclic connected, there is a collection $\mathcal{P} = \{C_1, \ldots, C_p\}$ of alternating cycles in G so that $x \in V(C_1)$, $y \in V(C_p)$, and, for every $i = 1, 2, \ldots, p-1$ and every $j = 1, 2, \ldots, p$, $|i - j| > 1$, $V(C_i) \cap V(C_{i+1}) \neq \emptyset$, $V(C_i) \cap V(C_j) = \emptyset$. ($\mathcal{P}$ corresponds to a (C_1, C_p)-path in $\Omega(\mathcal{R})$, where $\mathcal{R}$ is the set of all alternating cycles in G.) We traverse $\mathcal{P}$ as follows. We start at the red (blue, respectively) edge of C_1 incident to x and go along C_1 to the first vertex u that belongs to both C_1 and C_2. After meeting u, we go along C_2 such that the path that we are forming will stay alternating. We repeat the procedure above when we meet the first vertex that belongs to both C_2 and C_3 and so on. Clearly, we will eventually reach y. It follows that there is an (x, y)-path that starts from a red (blue, respectively) edge. By symmetry, we can construct an (x, y)-path that ends at a red (blue, respectively) edge. It follows from Proposition 11.1.10 that x and y are colour-connected. $\square$

11.1.4 Alternating Cycles in 2-Edge-Coloured Bipartite Multigraphs

The aim of this subsection is to describe two simple approaches which allow one to obtain results for bipartite 2-edge-coloured multigraphs using results on directed graphs.

Let D be a bipartite digraph with partite sets V_1, V_2. Define a 2-edge-coloured bipartite multigraph $CM(D)$ in the following way: $CM(D)$ has the same partite sets as D; every arc (v_1, v_2) from V_1 to V_2 is replaced with red edge v_1v_2 and every arc (v_2, v_1) from V_2 to V_1 is replaced with blue edge v_1v_2. Moreover, $CM^{-1}(G) = H$ if $CM(H) = G$. This simple correspondence which we call the **BB-correspondence** leads us to a number of easy and some more complex results which are described in this and the next subsections. (One example is the fact that the alternating Hamilton cycle problem for bipartite 2-edge-coloured graphs is $\mathcal{NP}$-complete.) In many of our results on cycles we will exploit the following easily verifiable proposition (see Exercise 11.10).

Proposition 11.1.15 *The following three claims are equivalent for a bipartite digraph D:*

(a) D is strongly connected.
(b) $CM(D)$ is colour-connected.
(c) $CM(D)$ is cyclic connected. $\square$

The following correspondence which we call the **BD-correspondence** is less universal but may allow one to exploit the wider area of results on arbitrary digraphs. The idea of the BD-correspondence can be traced back to Häggkvist [386]. Let G be a 2-edge-coloured bipartite multigraph with partite sets V_1 and V_2 so that $|V_1| = |V_2| = m$ and let G' be the red subgraph of G. Suppose that G' has a perfect matching $v_{11}v_{21}, v_{12}v_{22}, \ldots, v_{1m}v_{2m}$, where $v_{ij} \in V_i$ ($i = 1, 2$ and $1 \leq j \leq m$). Construct a digraph $D = D(G)$ as follows:

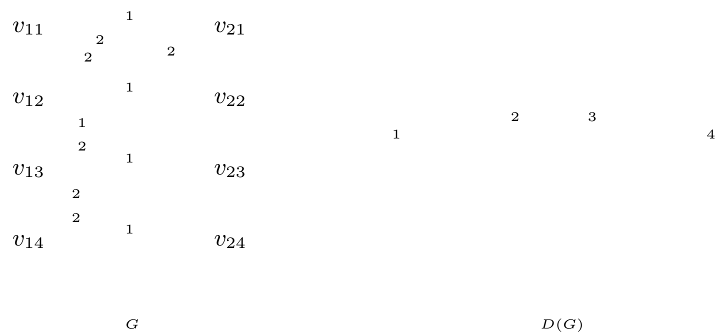


Figure 11.5 An illustration of BD-correspondence.

$V(D) = \{1, 2, \ldots, m\}$ and, for $1 \le i \ne j \le m$, (i, j) is an arc of D if and only if $v_{1i}v_{2j} \in E(G) - E(G')$ (see Figure 11.5). It is easy to see that, if D has a Hamilton cycle, then G has a Hamilton alternating cycle including all the edges of the perfect matching. Using the BD-correspondence and Corollary 5.6.3 Hilton [429] proved the following:

Theorem 11.1.16 *Let G be a 2-edge-coloured r-regular bipartite graph such that each of the partite sets of G has m vertices and let G'' be the blue subgraph of G. If $r \ge \frac{m}{2} + 1$ and G'' is s-regular such that $\frac{m}{2} \le s \le r - 1$, then G has an alternating Hamilton cycle.*

Proof: Exercise 11.11. □

Although the last theorem is the best possible (consider two disjoint copies of $K_{m/2,m/2}$ with perfect matchings in both copies in red and all other edges in blue), Hilton [429] believes that the bound on r could be lowered considerably if we assume that G is connected. It was noticed by Chetwynd and Hilton [155] that Theorem 11.1.16 follows easily from the following result by Häggkvist [386] (using the BB-correspondence).

Theorem 11.1.17 *Let G be a bipartite graph so that each of the partite sets contains m vertices. If $d(v) + d(w) \ge m + 1$ for every pair v, w of vertices from different partite sets, then every perfect matching of G lies in a Hamilton cycle of G.* □

The BB-correspondence is very useful when we consider 2-edge-coloured complete bipartite multigraphs. In this case we can use the rich theory of semicomplete bipartite digraphs (discussed in Chapters 5, 6). By the BB-correspondence, Proposition 11.1.15 and Theorem 5.7.4, we obtain the following:

Theorem 11.1.18 *A 2-edge-coloured complete bipartite multigraph contains an alternating Hamilton cycle if and only if it is colour-connected and has an* *alternating cycle factor. There is an algorithm for constructing an alternating Hamilton cycle in a colour-connected 2-edge-coloured complete bipartite multigraph on n vertices in time $O(n^{2.5})$ (if one exists).* □

Another condition for a 2-edge-coloured complete multigraph to contain an alternating Hamilton cycle was obtained by Chetwynd and Hilton [155]:

Theorem 11.1.19 *A 2-edge-coloured complete bipartite graph B with partite sets U and W ($|U| = |W| = n$) has an alternating Hamilton cycle if and only if B has an alternating cycle factor and, for every $k = 2, \ldots, n-1$ and every pair of k-sets X and Y such that $X \subset U$, $Y \subset W$, we have*

$$\min\{\sum_{x \in X} d_i(x) + \sum_{y \in Y} d_{3-i}(y) : \ i = 1, 2\} > k^2.$$

□

We point out that the original proof of Theorem 11.1.19 is quite similar to that of Theorem 5.7.4. (Another proof of Theorem 11.1.19 is given by Bang-Jensen and Gutin [61]; see also Exercise 11.14.) To see that the set of inequalities of this theorem is necessary, observe that the number of edges between X and Y is precisely k^2. If B has a Hamilton cycle C, then C contains an edge e_1 from $U - X$ to Y as well as an edge e_2 from X to $W - Y$ such that $\chi(e_1) = \chi(e_2)$. Precisely one of these edges contributes to the sum in the corresponding inequality.

Using the corresponding result on longest cycles in semicomplete bipartite digraphs (Theorem 5.7.6), one can obtain the following:

Theorem 11.1.20 *The length of the longest alternating cycle in a colour-connected 2-edge-coloured complete bipartite multigraph G is equal to the number of vertices in maximum alternating cycle subgraph of G. There is an algorithm for finding a longest alternating cycle in a colour-connected 2-edge-coloured complete bipartite multigraph on n vertices in time $O(n^3)$.* □

Let B_r and B'_r are 2-edge-coloured complete bipartite graphs with the same partite sets $\{v_1, \ldots, v_{2r}\}$ and $\{w_1, \ldots, w_{2r}\}$. The edge set of the red (blue) subgraph of B_r (B'_r) consists of

$$\{v_i w_j : \ 1 \leq i, j \leq r\} \cup \{v_i w_j : \ r+1 \leq i, j \leq 2r\}.$$

The following result is a characterization of vertex-alternating-pancyclic 2-edge-coloured complete bipartite multigraphs that can be readily obtained from the corresponding characterization for semicomplete bipartite digraphs in Theorem 6.13.1.

Theorem 11.1.21 *A 2-edge-coloured complete bipartite multigraph is vertex-alternating-pancyclic if and only if it has an alternating Hamilton cycle and is not colour-isomorphic to one of the graphs B_r, B'_r ($r = 2, 3, \ldots$).* □

Since none of the graphs B_r, B'_r $(r = 2, 3, \ldots)$ is alternating-pancyclic, we obtain the following:

Corollary 11.1.22 *Let G be a 2-edge-coloured complete bipartite multigraph. Then G is is alternating-pancyclic if and only if it has an alternating Hamilton cycle and is not colour-isomorphic to one of the graphs B_r, B'_r $(r = 2, 3, \ldots)$.* □

This result was obtained by Das [182]. The equivalent (via the BB-correspondence) claim was proved by Beineke and Little [99] for bipartite tournaments. (Both results were published in the same year!)

To save the space we will not give any other 'BB-translations' of results obtained for cycles and paths in semicomplete bipartite digraphs (see Chapters 5, 6) into the alternating cycles and paths language.

11.1.5 Longest Alternating Paths and Cycles in 2-Edge-Coloured Complete Multigraphs

Since the longest alternating path problem for 2-edge-coloured complete multigraphs is much simpler than the longest alternating cycle problem, we start our study from the former. Bang-Jensen and Gutin characterized 2-edge-coloured complete multigraphs which have an alternating Hamilton path (see Corollary 11.1.24).

Theorem 11.1.23 *[61] Let G be a 2-edge-coloured complete multigraph with n vertices. Then for any 1-path-cycle subgraph $\mathcal{F}$ of G there is an alternating path P of G satisfying $V(P) = V(\mathcal{F})$ (if $\mathcal{F}$ is a maximum alternating 1-path-cycle subgraph of G, then P is a longest alternating path in G); there exists an $O(n^3)$ algorithm for finding a longest alternating path in G.*

Proof: Obviously, $\mathcal{F}$ is a 1-path-cycle factor of a complete bipartite subgraph B of G. The factor $\mathcal{F}$ corresponds to a directed path together with a collection of directed cycles, all vertex disjoint, $\mathcal{F}'$ of $CM^{-1}(B)$. Therefore, by Theorem 5.7.1 restricted to semicomplete bipartite digraphs, there is a path P' in $CM^{-1}(B)$ such that $V(P') = V(\mathcal{F}')$. This path corresponds to an alternating path P of B so that $V(P') = V(P)$. Clearly, P is an alternating path in G and, moreover, $V(P) = V(\mathcal{F})$.

The complexity result easily follows from the construction above, and Theorems 5.7.1 and 11.1.8. □

Corollary 11.1.24 *[61] A 2-edge-coloured complete multigraph has an alternating Hamilton path if and only if it contains an alternating 1-path-cycle factor.* □

It is not difficult to prove Corollary 11.1.24 directly (see Exercise 11.17). Clearly, Corollary 11.1.24 implies immediately the first part of Theorem

11.1.23. Thus, the first part of Theorem 11.1.23 and Corollary 11.1.24 are in fact equivalent.

In 1968, solving a problem by Erdős, Bankfalvi and Bankfalvi [91] gave the following characterization of 2-edge-coloured complete graphs which have an alternating Hamilton cycle.

Theorem 11.1.25 (Bankfalvi and Bankfalvi) *[91] A 2-edge-coloured complete graph G of order $2n$ has an alternating Hamilton cycle if and only if it has an alternating cycle factor and, for every $k = 2, \dots, n-1$ and every pair of disjoint k-subsets X and Y of $V(G)$, $\sum_{x \in X} d_1(x) + \sum_{y \in Y} d_2(y) > k^2$.* □

It is easy to see that the conditions of this theorem are necessary (Exercise 11.13). Saad [648] proved the following more general result, using the notion of colour-connectivity rather than degree conditions. We provide a proof of Theorem 11.1.26 in the end of this subsection after some discussion of implications and generalizations of Theorem 11.1.26.

Theorem 11.1.26 (Saad) *[648] The length of a longest alternating cycle in a colour-connected 2-edge-coloured complete multigraph G is equal to the number of vertices in a maximum alternating cycle subgraph of G.*

Corollary 11.1.27 *[648] A 2-edge-coloured complete multigraph G has an alternating Hamilton cycle if and only if G is colour-connected and contains an alternating cycle factor.* □

Corollary 11.1.27 and the fact that colour-connectivity can be checked in polynomial time (see Propositions 11.1.10 and 11.1.11) shows that the alternating hamiltonian cycle problem for 2-edge-coloured complete multigraphs is polynomial time solvable. However, one cannot deduce the analogous result for the longest alternating cycle problem (for 2-edge-coloured complete multigraphs) from Theorems 11.1.26 and 11.1.8 and Propositions 11.1.10 and 11.1.11, only. The reason is that we do not know how to obtain all maximal colour-connected subgraphs of an arbitrary 2-edge-coloured multigraph in polynomial time. Fortunately, for 2-edge-coloured complete multigraphs G, colour-connectivity is an equivalence relation on the set of vertices (this was first proved by Saad [648] and also follows from Proposition 11.1.12 and the following deeper theorem by Bang-Jensen and Gutin [64]):

Theorem 11.1.28 *[64] A 2-edge-coloured complete multigraph G is colour-connected if and only if G is cyclic connected.* □

Proof: Exercise 11.15.

Thus, we can use Propositions 11.1.10 and 11.1.11 to obtain (vertex-disjoint) colour-connected components of G. Hence, the longest alternating cycle problem for 2-edge-coloured complete multigraphs is also polynomial time solvable. In [64], Bang-Jensen and Gutin showed the following more general result. (Clearly, the case $f(x) = 1$ for every $x \in V(G)$ corresponds to the longest alternating cycle problem.)

Theorem 11.1.29 *[64] The following problem is polynomial time solvable. Given a function f from $V(G)$, the vertex set of a 2-edge-coloured complete multigraph G, to $\mathcal{Z}_0$, find a maximum size alternating closed trail H in G such that $d_{1,H}(x) = d_{2,H}(x) \leq f(x)$ for every $x \in V(G)$.* □

Das [182] and later Häggkvist and Manoussakis [389] observed that the alternating hamiltonian cycle problem for 2-edge-coloured complete bipartite multigraphs can be reduced to the same problem for 2-edge-coloured complete multigraphs using the following simple construction. Consider a 2-edge-coloured complete bipartite multigraph L with bipartition (X, Y). Add to L the edges $\{x'x'', y'y'' \ : \ x', x'' \in X, \ y', y'' \in Y\}$ and set $\chi(XX) = 1, \ \chi(YY) = 2$. Let K be the 2-edge-coloured complete multigraph obtained in this way. It is not difficult to verify that K has no alternating cycle containing any of the edges from $XX \cup YY$. Hence, K contains an alternating hamiltonian cycle if and only if L has one. Moreover, it is easy to check that K is colour-connected if and only if L is colour-connected. In the following, we will call the construction above the **DHM-construction**. The DHM-construction shows that (the non-algorithmic part of) Theorem 11.1.18 follows immediately from Corollary 11.1.27. This illustrates the fact that many problems on alternating cycles for 2-edge-coloured complete multigraphs are more general than those for 2-edge-coloured complete bipartite multigraphs.

Consider the following Hamiltonian 2-edge-coloured complete graphs which are not even-pancyclic (see the proof of this fact below). Let $r \geq 2$ be an integer. Each of the graphs $H(r), \ H'(r), \ H''(r)$ has a vertex set $A \cup B \cup C \cup D$ so that the sets A, B, C, D are pairwise disjoint and each of these sets contains r vertices. Moreover, the edge set of the red subgraph of $H(r)$ consists of $AA \cup CC \cup AC \cup AD \cup CB$. The edge set of the red (blue) subgraph of $H'(r)$ ($H''(r)$) consists of $AC \cup CB \cup BD \cup DA$. By the DHM-construction, the following result by Bang-Jensen and Gutin [61] is a generalization of Theorem 11.1.21 (the proof is left as Exercise 11.16).

Theorem 11.1.30 *Let G be a 2-edge-coloured complete multigraph. Then G is vertex-alternating-pancyclic if and only if G has an alternating Hamilton cycle and is not colour-isomorphic to the graphs $H(r), \ H'(r), \ H''(r)$ for $r = 2, 3, \ldots$.* □

Since the graphs $H(r), \ H'(r), \ H''(r)$ are not alternating-pancyclic for $r = 2, 3, \ldots$, we obtain the following characterization first proved by Das [182].

Corollary 11.1.31 *A 2-edge-coloured complete multigraph G is alternating-pancyclic if and only if G has an alternating Hamilton cycle and is not colour-isomorphic to the graphs $H(r)$, $H'(r)$, $H''(r)$ for $r = 2, 3, \ldots$.* □

The rest of this subsection is devoted to the proof of Theorem 11.1.26 adapted from Bang-Jensen and Gutin [64]. In the statements and the proofs of the rest of this subsection, we use the following notation: G is a 2-edge-coloured complete multigraph with n vertices, $\mathcal{F}_p = C_1 \cup \ldots \cup C_p$ is an alternating cycle subgraph in G consisting of p cycles, $C_1, \ldots, C_p$; for each $i = 1, 2, \ldots, p$, $C_i = v_1^i v_2^i \ldots v_{2k(i)}^i v_1^i$ such that $\chi(v_1^i v_2^i) = 1$, $\chi(v_{2k(i)}^i v_1^i) = 2$, and $X_i = \{v_1^i, v_3^i, \ldots, v_{2k(i)-1}^i\}$, $Y_i = V(C_i) - X_i$. We write $C_j \rightarrow C_i$ to denote that

$$\chi(X_i X_i) = \chi(X_i V(C_j)),\ \chi(Y_i Y_i) = \chi(Y_i V(C_j)) \text{ and } \chi(X_i X_i) \neq \chi(Y_i Y_i).$$

We point out that the meaning of $C_j \rightarrow C_i$ is that, for any choice of vertices $x \in V(C_j)$ and $y \in V(C_i)$, there exist alternating (x, y)-paths P and P' such that the colours of the edges incident with x in P and P' are distinct, but for every such choice of paths P and P', the colours of the edges in P and P' incident with y are equal. Hence, if $C_j \rightarrow C_i$, then the multigraph induced by the vertices of these two cycles is not colour-connected. (See Figure 11.6, where $C_2 \rightarrow C_3$.)

Lemma 11.1.32 *Suppose G has an alternating cycle factor $\mathcal{F}_2 = C_1 \cup C_2$. Then, G has an alternating Hamilton cycle if and only if neither $C_1 \rightarrow C_2$ nor $C_2 \rightarrow C_1$. Given a pair C_1 and C_2 of cycles of G, so that neither $C_1 \rightarrow C_2$ nor $C_2 \rightarrow C_1$, an alternating Hamilton cycle of G can be found in time $O(|V(C_1)||V(C_2)|)$.*

Proof: It is easy to see that, if either $C_1 \rightarrow C_2$ or $C_2 \rightarrow C_1$, then G is not colour-connected. Hence, G has no alternating Hamilton cycle. Assume that neither $C_1 \rightarrow C_2$ nor $C_2 \rightarrow C_1$, but G has no alternating Hamilton cycle. Consider the bipartite digraph T with partite sets $V_1 = X_1 \cup X_2$ and $V_2 = Y_1 \cup Y_2$ obtained from G in the following way: delete all edges between vertices both on C_1 or on C_2 except those edges that are on the cycles and delete all edges between vertices both in the same partite set. Now make the following orientations of the edges in the resulting bipartite multigraph. For $i = 1, 2$ and any pair $v_1 \in V_1$, $v_2 \in V_2$, if there is an edge e between v_1 and v_2, then delete the colour of the edge e and orient it as the arc (v_i, v_{3-i}) if and only if $\chi(e) = i$. Obviously, T has a spanning cycle subgraph consisting of two directed cycles Z_1, Z_2 which are orientations of the cycles C_1, C_2, respectively. Similarly we see that every directed cycle in T corresponds to an alternating cycle in G. Thus, since G has no alternating Hamilton cycle, T is not hamiltonian. By Exercise 5.34, this means that T is not strong, i.e. all arcs between Z_1 and Z_2 have the same orientation. Without loss of generality we may assume that all these arcs are oriented from Z_1 to Z_2. Then, by the definition of T, we obtain that $\chi(X_1Y_2) = 1,\ \chi(Y_1X_2) = 2$.

Consider next the bipartite digraph T' with partite sets $V_1' = X_1 \cup Y_2$ and $V_2' = Y_1 \cup X_2$. The rest of the definition of T' coincides with that of T. T' also contains a spanning cycle subgraph consisting of orientations of C_1 and C_2. Since G has no alternating Hamilton cycle, T' is not hamiltonian. By Corollary 5.34, this means that T' is not strongly connected. This leads us to the conclusion that either $\chi(X_1X_2) = 1$ and $\chi(Y_1Y_2) = 2$ or $\chi(X_1X_2) = 2$ and $\chi(Y_1Y_2) = 1$. The first possibility together with the conclusion of the previous paragraph implies $\chi(X_1V(C_2)) = 1, \chi(Y_1V(C_2)) = 2$. The second gives $\chi(X_2V(C_1)) = 2, \chi(Y_2V(C_1)) = 1$. Without loss of generality we may assume that $\chi(X_1V(C_2)) = 1, \chi(Y_1V(C_2)) = 2$.

Suppose that, for some $i \neq j$, there exists an edge $v^1_{2i+1}v^1_{2j+1}$ of colour 2. Then G has the alternating Hamilton cycle

$$v^2_1 v^1_{2j} v^1_{2j-1} \dots v^1_{2i+1} v^1_{2j+1} \dots v^1_{2k(1)} v^1_1 \dots v^1_{2i} v^2_{2k(2)} \dots v^2_1.$$

Hence, $\chi(X_1X_1) = 1$. Analogously, $\chi(Y_1Y_1) = 2$. Now $C_2 \rightarrow C_1$ and we have obtained a contradiction.

The complexity bound follows from that of Corollary 5.34. $\square$

An alternating cycle subgraph $\mathcal{R}$ of G is **irreducible** if there is no other alternating cycle subgraph $\mathcal{Q}$ in G so that $V(\mathcal{R}) = V(\mathcal{Q})$ and $\mathcal{Q}$ has fewer cycles than $\mathcal{R}$. (See Figure 11.6.)

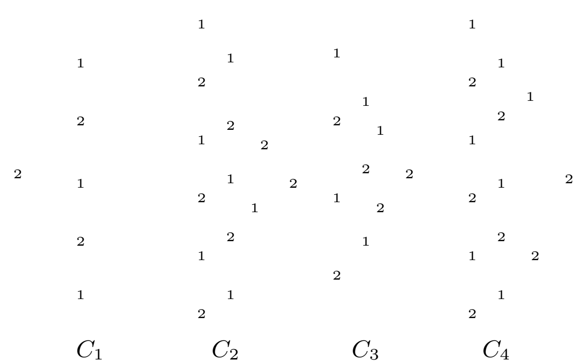


Figure 11.6 An irreducible PC cycle factor. The number $s \in \{1, 2\}$ on the edge emanating to the left from a vertex on C_i, $2 \leq i \leq 2$ indicates that the colour of all edges from that vertex to all the vertices of C_j with $j < i$ is s. The vertices are partitioned into two equal sized sets indicated by black and white vertices. The number $r \in \{1, 2\}$ on an edge between two black (white vertices) on the same cycle indicates that all edges between black (white) vertices on that cycle have the same colour r.

Theorem 11.1.33 *Let G have an alternating cycle factor $\mathcal{F}$ consisting of $p \geq 2$ cycles. $\mathcal{F}$ is an irreducible alternating cycle factor of G if and only if we can label the cycles in $\mathcal{F}$ as $C_1, \ldots, C_p$, such that, with the notation introduced above, for every $1 \leq i < j \leq p$, $\chi(X_jV(C_i)) = 1$, $\chi(Y_jV(C_i)) = 2, \chi(X_jX_j) = 1, \chi(Y_jY_j) = 2$. An irreducible alternating cycle factor of G (if any) can be found in time $O(n^{2.5})$.*

Proof: If the edges have the structure described above, then $C_i \rightarrow C_j$ for all $i < j$ and each of the cycles in $\mathcal{F}$ form a colour-connected component and $\mathcal{F}$ is clearly irreducible.

To prove the other direction we let $\mathcal{F}$ be an irreducible alternating cycle factor of G and let $p \geq 2$ be the number of cycles in $\mathcal{F}$. By Lemma 11.1.32, no two cycles in $\mathcal{F}$ induce a colour-connected subgraph. Thus, for all $1 \leq i < j \leq p$, either $C_i \rightarrow C_j$ or $C_j \rightarrow C_i$. Therefore, the digraph with vertex set $\{C_1, \ldots, C_p\}$ and arc set $\{(C_i, C_j) : \; C_i \rightarrow C_j; 1 \leq i \neq j \leq p\}$ is a tournament. So, if there exist cycles $C'_1, C'_2, \ldots, C'_k$ from $\mathcal{F}$ such that $C'_1 \rightarrow C'_2 \rightarrow \ldots \rightarrow C'_k \rightarrow C'_1$, then there also exists such a collection for $k = 3$ and the reader can easily find an alternating cycle covering precisely the vertices of those cycles, contradicting the irreducibility of $\mathcal{F}$. Hence we can assume that there is no such cycle. Thus there is a unique way to label the cycles in $\mathcal{F}$ as $C_1, C_2, \ldots, C_p$, so that $C_i \rightarrow C_j$ if and only if $i < j$. If there are three cycles C_i, C_j and C_k from $\mathcal{F}$ such that $C_i \rightarrow C_j, C_k$ and $C_j \rightarrow C_k$, but $\chi(X_kV(C_i)) \neq \chi(X_kV(C_j))$, then we can easily find an alternating cycle covering precisely the vertices of C_i, C_j and C_k, contradicting the irreducibility of $\mathcal{F}$. Hence we may assume that for all $1 \leq i < j \leq p$, $\chi(X_jV(C_i)) = 1$ and $\chi(Y_jV(C_i)) = 2$. The fact that $\chi(X_jX_j) = 1, \chi(Y_jY_j) = 2$ follows from the proof of Lemma 11.1.32 and the minimality of $\mathcal{F}$.

Using the proof of Lemma 11.1.32, the proof above can be converted into an $O(n^2)$-algorithm for transforming any alternating cycle factor into an alternating hamiltonian cycle or an irreducible alternating cycle factor. Now the complexity bound of the lemma follows from a simple fact that one can find a spanning alternating cycle subgraph (if any) in a 2-edge-coloured multigraph L in time $O(|V(L)|^{2.5})$. Indeed, find maximum matchings in the red and blue subgraphs of L. Obviously, L has a spanning alternating cycle subgraph if and only if both subgraphs have perfect matchings. The complexity bound follows from that of the algorithm for finding a maximum matching in an arbitrary graph described in the book [231] by Even. $\square$

We will make use of the following simple lemma.

Lemma 11.1.34 *Let $P = x_1x_2 \ldots x_k$ be an alternating path and C an alternating cycle disjoint from P in G. Suppose $\chi(x_1V(C)) = i \neq \chi(x_1x_2)$ where $i = 1$ or $i = 2$ and that G contains an edge x_kz, where $z \in V(C)$ and $\chi(x_{k-1}x_k) \neq \chi(x_kz)$. If $\chi(x_kz) = i$, then G contains a cycle C' with $V(C') = V(P) \cup V(C)$. Otherwise G has a cycle C'' with $V(C'') = V(P) \cup V(C) - w$, where w is the neighbour of z on C for which $\chi(wz) = 3 - i$.*

Proof: Exercise 11.18. □

Proof of Theorem 11.1.26: Let $\mathcal{F} = C_1 \cup \ldots \cup C_p$ be an alternating cycle subgraph of G and let $\mathcal{F}' = C_1 \cup \ldots \cup C_{p-1}$. We will show by induction on p that G has an alternating cycle C^* having at least the same number of vertices as $\mathcal{F}$. If $p = 1$, we are done. So, we may suppose that $p \geq 2$. By Theorem 11.1.33, we may assume, using the (obvious) induction hypothesis, that, for all $1 \leq i < j \leq p$,

$$\chi(X_j V(C_i)) = 1,\ \chi(Y_j V(C_i)) = 2, \chi(X_j X_j) = 1, \chi(Y_j Y_j) = 2. \qquad (11.2)$$

Since G is colour-connected there is an alternating (x, y)-path R of minimum length such that $x \in V(C_p)$, $\{y\} = V(R) \cap V(\mathcal{F}')$ and $\chi(xx') \neq \chi(xV(\mathcal{F}'))$, where x' is the successor of x in R. We prove that $(V(R) - \{x, y\}) \cap V(\mathcal{F}) = \emptyset$. Assume this is not so, that is, R contains at least two vertices from C_p. Consider a vertex z in $(V(R) \cap V(C_p)) - x$. Let z' be the successor of z in R. Clearly, $\chi(zz') = \chi(zV(\mathcal{F}'))$ since the (z, y)-part of R is shorter than R. On the other hand, by (11.2) x' is not in C_p and by the minimality of R, $\chi(x'V(\mathcal{F}')) = \chi(xx')$. Then, the alternating path Qv, where Q is the reverse of the (x', z)-part of R and v is a vertex in C_{p-1}, is shorter than R; a contradiction.

Now consider an alternating (x, y)-path R with the properties above including $(V(R) - \{x, y\}) \cap V(\mathcal{F}) = \emptyset$. We may assume without loss of generality that $x = v_1^p$ and $\chi(xV(\mathcal{F}')) = \chi(v_2^p v_1^p)$. Choose t such that $y \in V(C_t)$. Apply Lemma 11.1.34 to the path

$$v^p_{2k(p)} v^p_{2k(p)-1} \ldots v^p_2 R',$$

where R' is the path R without y, and the cycle C_t. We get a new alternating cycle C', with $V(C') \subset V(R) \cup V(C_t) \cup V(C_p)$, covering at least as many vertices as C_t and C_p together, so by replacing C_t and C_p by C' in $\mathcal{F}$, we obtain a new alternating cycle subgraph with fewer cycles which covers at least as many vertices as $\mathcal{F}$ and the existence of C^* follows by induction. □

The proof above can be converted into an $O(n^3)$-algorithm for finding a longest cycle in G, provided we are given a maximum cycle subgraph as input.

11.1.6 Properly Coloured Hamiltonian Paths in c-Edge-Coloured Complete Graphs, $c \geq 3$

Let K_n^c denote a c-edge-coloured complete graph with n vertices. The properly coloured (PC) Hamilton path problem for c-edge-coloured complete graphs seems to be much more difficult in the case $c \geq 3$, than in the case $c = 2$ treated above.

Problem 11.1.35 *[61] Determine the complexity of deciding whether a c-edge-coloured complete graph, $c \geq 3$, has a PC Hamilton path.* $\square$

There is a polynomial time algorithm for Problem 11.1.35 if the following generalization of Corollary 11.1.24 is true.

Conjecture 11.1.36 *[61] A K_n^c ($c \geq 2$) has a PC Hamilton path if and only if K_n^c contains a PC spanning 1-path-cycle subgraph.* $\square$

We know that the claim of this conjecture is true when $c = 2$ (see Corollary 11.1.24). The following weaker result is proved by Bang-Jensen, Gutin and Yeo [73].

Theorem 11.1.37 *If a K_n^c ($c \geq 2$) contains a PC spanning cycle subgraph, then it has a PC Hamilton path.*

Proof: Let $C_1, C_2, \ldots, C_t$ be the cycles of a PC spanning cycle subgraph $\mathcal{F}$ of K_n^c. Let $\mathcal{F}$ be chosen so that, among all PC spanning cycle subgraphs of K_n^c, the number of cycles t is minimum. We say that C_i **edge-dominates** C_j ($i \neq j$) if, for every edge xy of C_i, there exists an edge between x and C_j and an edge between y and C_j whose colours differ from the colour of xy. Construct a digraph D as follows. The vertices of D are $1, 2, \ldots, t$ and an arc (i, j) is in D ($1 \leq i \neq j \leq t$) if and only if C_i edge-dominates C_j.

First we show that D is semicomplete. Suppose this is not so, i.e. there exist vertices i and j which are not adjacent. This means that neither C_i edge-dominates C_j nor C_j edge-dominates C_i. Thus C_i has an edge xy such that $\chi(xV(C_j)) = \chi(xy)$ and C_j has an edge uv such that $\chi(uV(C_i)) = \chi(uv)$. It follows that $\chi(xy) = \chi(xu) = \chi(uv) = \chi(xv) = \chi(uy)$. Therefore, we can merge the two cycles to obtain a new properly coloured one as follows: delete xy and uv, and append xv and yu. However, this is a contradiction to t being minimum. Thus, D is indeed semicomplete.

Since D is semicomplete, it follows from Theorem 1.4.5 that D has a Hamilton directed path: $i_1 i_2 \ldots i_t$. Without loss of generality we may assume that $i_k = k$ for every $k = 1, 2, \ldots, t$. In other words, C_i edge-dominates C_{i+1} for every $1 \leq i \leq t-1$. Let $C_i = z_1^i z_2^i \ldots z_{m_i}^i z_1^i$ ($i = 1, 2, \ldots, t$). Since C_1 edge-dominates C_2, without loss of generality, we may assume the labellings of the vertices in C_1 and C_2 are such that $\chi(z_{m_1}^1 z_1^1) \neq \chi(z_1^1 z_2^2)$. Since the edges $z_1^2 z_2^2$ and $z_2^2 z_3^2$ have different colours, without loss of generality we may assume that $\chi(z_2^2 z_3^2) \neq \chi(z_1^1 z_2^2)$. Analogously, for every $i = 1, 2, \ldots, t-1$, we may assume that $\chi(z_{m_i}^i z_1^i) \neq \chi(z_1^i z_2^{i+1}) \neq \chi(z_2^{i+1} z_3^{i+1})$. Now we obtain the following PC Hamilton path:

$$z_2^1 z_3^1 \ldots z_{m_1}^1 z_1^1 z_2^2 z_3^2 \ldots z_{m_2}^2 z_1^2 \ldots z_2^t z_3^t \ldots z_{m_t}^t z_1^t.$$

$\square$

The above theorem can be considered as a sufficient condition for an edge-coloured complete graph to have a PC Hamilton path. We state two other sufficient conditions. The first theorem is by Barr and has a simple inductive proof (Exercise 11.19). The proof of the second theorem, due to Manoussakis, Spyratos, Tuza and Voigt, is much more involved; it is omitted.

Theorem 11.1.38 *[94] Every K_n^c without monochromatic triangles has a PC Hamilton path.* □

Theorem 11.1.39 *[549] If $c \geq \frac{1}{2}(n-3)(n-4)+2$, then there is an $n_0 = n_0(c)$ such that, for every $n \geq n_0$, each K_n^c has a PC Hamilton path.* □

11.1.7 Properly Coloured Hamiltonian Cycles in c-Edge-Coloured Complete Graphs, $c \geq 3$

Benkouar, Manoussakis, Paschos and Saad posed the following problem which is analogous to Problem 11.1.35:

Problem 11.1.40 *[103] Determine the complexity of the PC Hamilton cycle problem for c-edge-coloured complete graphs when $c \geq 3$.*

Another interesting problem is to find a non-trivial characterization of c-edge-coloured ($c \geq 3$) complete graphs containing PC hamiltonian cycles. In this subsection, we consider results from [103] related to Problem 11.1.40. We give an example showing that the obvious analogue of Corollary 11.1.27 is not valid for $c \geq 3$. Later we present some conditions which guarantee the existence of a PC Hamilton cycle in a c-edge-coloured complete graph.

A **strictly alternating cycle** in K_n^c is a cycle of length pc (p is an integer) so that the sequence of colours $(12\ldots c)$ is repeated p times. Benkouar, Manoussakis, Paschos and Saad [103] proved the following:

Theorem 11.1.41 *[103] Let $c \geq 3$. The problem of determining the existence of a strictly alternating Hamilton cycle in K_n^c is $\mathcal{NP}$-complete.*

Proof: Exercise 11.20. □

The following result shows that, if we relax the property of colours to be at strict places, but maintain the number of their appearances in a Hamilton cycle, then we still have an $\mathcal{NP}$-complete problem.

Theorem 11.1.42 *[103] Given positive integers p and $c \geq 3$, the problem of determining the existence of a PC Hamilton cycle C of K_{cp}^c so that each colour appears p times in C is $\mathcal{NP}$-complete.*

Proof: Exercise 11.21. □

The following example shows that the obvious analogue of Corollary 11.1.27 is not valid for $c \geq 3$. The graph G_6 is a 3-edge-coloured complete graph on vertices 1,2,3,4,5,6. All the edges of G_6 has colour 1 except for the following: the triangles 2342 and 2562 have colours 2 and 3, respectively, $\chi(36) = \chi(45) = 2$, $\chi(12) = 3$. It is easy to check that G_6 is colour-connected and has the alternating spanning cycle subgraph $1231 \cup 4564$, but G_6 contains no PC Hamilton cycle (Exercise 11.22). Note that alternating paths showing that G_6 is colour-connected may be chosen so that for each choice of vertices x and y the two paths P and P' described in the definition of colour-connectivity are internally disjoint. Hence it will not be enough to change this definition to require that P and P' are disjoint, a condition which is obviously necessary for the existence of a PC Hamilton cycle. For every even n, using the definition of G_6, one can easily construct a 3-edge-coloured complete graph on $n \geq 8$ vertices which is colour-connected and has a PC spanning cycle subgraph, but contains no PC Hamilton cycle (see Exercise 11.23).

We start our consideration of sufficient conditions for an edge-coloured complete graph to contain a PC Hamilton cycle with the following simple result by Manoussakis, Spyratos, Tuza and Voigt:

Proposition 11.1.43 *[549] If* $c \geq \frac{1}{2}(n-1)(n-2)+2$*, then every* K_n^c *has a PC Hamilton cycle.*

Proof: Exercise 11.24. □

To see that the bound of Proposition 11.1.43 is sharp consider the following K_n^c. Assign colour 1 to all edges incident to a fixed vertex $x \in V(K_n^c)$. Each of the remaining edges has a distinct colour not equal 1. Clearly, such K_n^c has no PC Hamilton cycle and $c = \frac{1}{2}(n-1)(n-2)+1$.

In [184] Daykin posed the following interesting problem. Find a positive constant d such that every K_n^c with $\Delta_{mon}(K_n^c) \leq dn$ has a PC Hamilton cycle. This problem was independently solved by Bollobás and Erdős [121], and Chen and Daykin [145]. In [121] (in [145], respectively), it was proved that, if $69\Delta_{mon}(K_n^c) < n$ ($17\Delta_{mon}(K_n^c) \leq n$, respectively), then K_n^c has a PC Hamilton cycle. Shearer [668] improved the last result showing that if $7\Delta_{mon}(K_n^c) < n$, then K_n^c has a PC Hamilton cycle. So far, the best asymptotic estimate was obtained by Alon and Gutin [11].

Theorem 11.1.44 *[11] For every* $\epsilon > 0$ *there exists an* $n_0 = n_0(\epsilon)$ *so that for each* $n > n_0$*, every* K_n^c *satisfying*

$$\Delta_{mon}(K_n^c) \leq (1 - \frac{1}{\sqrt{2}} - \epsilon)n \quad (\ = (0.2928\ldots - \epsilon)n \ ) \tag{11.3}$$

contains a PC Hamilton cycle.

However, Theorem 11.1.44 seems to be far from the best possible, at least, if the following conjecture by Bollobás and Erdős [121] is true.

Conjecture 11.1.45 *Every K_n^c with $\Delta_{mon}(K_n^c) \leq \lfloor n/2 \rfloor - 1$ has a PC Hamilton cycle.*

The rest of this subsection is devoted to a probabilistic[1] proof of Theorem 11.1.44. For simplicity we assume first that $n = 2m$ is even, and remark at the end of the subsection how to modify the argument for the case of odd n. Fix a positive ϵ, and let $K = K_n^c$ be an edge-coloured complete graph on $n = 2m$ vertices satisfying (11.3). We first prove the following lemma.

Lemma 11.1.46 *For all sufficiently large m, K contains a spanning edge coloured complete bipartite graph $K_{m,m}^c$ satisfying*

$$\Delta_{mon}(K_{m,m}^c) \leq (1 - \frac{1}{\sqrt{2}} - \frac{\epsilon}{2})m. \tag{11.4}$$

Proof: Let u_iv_i $(1 \leq i \leq m)$ be an arbitrary perfect matching in K and choose a random partition of the set of vertices of K into two disjoint subsets A and B of cardinality m each by choosing, for each i, $1 \leq i \leq m$, randomly and independently, one element of the set $\{u_i, v_i\}$ to be a member of A and the other to be a member of B. Fix a vertex w of K and a colour, say red, that appears in the edge-colouring of K. The number of neighbours a of w in A so that the edge wa is red can be written as a sum of m independent indicator random variables $x_1, \ldots, x_m$, where x_i is the number of red neighbours of w in A among u_i, v_i. Thus each x_i is either 1 with probability one (in case both edges wu_i, wv_i are red) or 0 with probability 1 (in case none of the edges wu_i, wv_i is red) or 1 with probability 1/2 (in case exactly one of these two edges is red). It follows that, if the total number of red edges incident with w is r then the probability that w is adjacent with more than $(r+s)/2$ vertices in A by red edges is equal to the probability that more than $(q+s)/2$ flips among q independent flips of a fair coin give 'heads', where q is the number of nonconstant indicator random variables among the x_i's. This can be bounded by the well known inequality of Chernoff (cf. e.g. [14, Theorem A.4, page 235]) by $e^{-2s^2/q} < e^{-2s^2/m}$. Since the same argument applies to the number of 'red' neighbours of w in B, and since there are less than $8m^3$ choices for a vertex w, a colour in the given colouring of K and a partite set (A or B), we conclude that the probability that there exists a vertex with more than

$$(1 - \frac{1}{\sqrt{2}} - \frac{\epsilon}{2})m$$

neighbours of the same colour in either A or B is at most

$$8m^3e^{-2\epsilon^2 m},$$

[1] Probabilistic methods have proved to be very powerful for various problems (see e.g. the book [14] by Alon and Spencer).

which is (much) smaller than 1 for all sufficiently large m. Therefore, there exists a choice for A and B so that the above does not occur, completing the proof. □

The next lemma can be proved by applying a large deviation result for martingales, i.e., Azuma's inequality [14].

Lemma 11.1.47 *[11] Let U be a subset of $M = \{1, 2, \ldots, m-1\}$ and suppose that for each $u \in U$ there is a subset $S_u \subset M$, where $|S_u| \leq r$ for all u. Let $f : U \mapsto M$ be a random one-to-one mapping of U into M, chosen uniformly among all one-to-one mappings of U into M, and define:*

$$B(f) = |\{u \in U : f(u) \in S_u\}|.$$

Then the expectation of $B(f)$ is given by

$$E = E(B(f)) = \sum_{u \in U} \frac{|S(u)|}{m-1} \quad (\leq \frac{|U|r}{m-1},)$$

and the probability that $B(f)$ is larger satisfies the following inequality. For every $\lambda > 0$

$$Prob[B(f) - E > 4\lambda\sqrt{m-1}] < e^{-\lambda^2}.$$

□

Corollary 11.1.48 *Let $K^c_{m,m}$ be an edge-coloured complete bipartite graph on the partite sets A and B, and suppose that (11.4) holds. Then, for all sufficiently large m, there exists a perfect matching a_ib_i, $1 \leq i \leq m$, in $K^c_{m,m}$ so that the following two conditions hold.*
(i) For every i the number $d^+(i)$ of edges a_ib_j between a_i and B whose colours differ from those of a_ib_i and of a_jb_j is at least $m/2+1$.
(ii) For every j the number $d^-(j)$ of edges a_ib_j between b_j and A whose colours differ from those of a_ib_i and of a_jb_j is at least $m/2+1$.

Proof: Let a_ib_i, $1 \leq i \leq m$, be a random perfect matching between A and B, chosen among all possible matchings with uniform probability. Put $r = \Delta_{mon}(K^c_{m,m})$ and notice that by (11.4)

$$r \leq (1 - \frac{1}{\sqrt{2}} - \frac{\epsilon}{2})m.$$

Fix an i, say $i = m$, and let us estimate the probability that the condition (i) fails for i. Suppose the edge a_mb_m has already been chosen for our random matching, and the rest of the matching still has to be chosen randomly. There are at most r edges a_mb ($b \in B$) having the same colour as a_mb_m. Let U be the set of all the remaining elements B. Then $|U| \geq m - r$. For each $u \in U$, let S_u denote the set of all elements $a \in A - a_m$ so that the colour of the edge au is equal to that of the edge $a_m u$. The random matching restricted to U is simply a random one-to-one function f from U to $A - a_m$. Moreover, the edge $a_m u$ will not be counted among the edges incident with a_m and having colours that differ from those of $a_m b_m$ and of the edge matched to u if and only if the edge matched to u will lie in S_u. It follows that the random variable counting the number of such edges of the form $a_m u$ behaves precisely like the random variable $B(f)$ in Lemma 11.1.47. By choosing say, $\lambda = \sqrt{\log(4m)}$ we conclude that the probability that $B(f)$ exceeds $|U|r/(m-1) + 4\lambda\sqrt{m-1}$ is smaller than $1/(4m)$. Therefore, with probability at least $1 - \frac{1}{4m}$

$$\begin{aligned} d^+(m) &\geq |U| - \frac{|U|r}{m-1} - 4\sqrt{m}\sqrt{\log(4m)} \\ &\geq \frac{(m-r)(m-r-1)}{m-1} - 4\sqrt{m}\sqrt{\log(4m)} \\ &> m/2 + 1, \end{aligned}$$

for all sufficiently large m (using the fact that $r \leq (1 - \frac{1}{\sqrt{2}} - \frac{\epsilon}{2})m$.)

Since there are m choices for the vertex a_i (and similarly m choices for the vertex b_j for which the computation is similar) we conclude that with probability at least a half $d^+(i) > m/2 + 1$, and $d^-(j) > m/2 + 1$ for all i and j. In particular there exists such a matching, completing the proof of the corollary. □

Returning to the proof of Theorem 11.1.44 with $n = 2m$, and given an edge-coloured K_n^c satisfying (11.3) apply Lemma 11.1.46 and Corollary 11.1.48 to obtain a matching $a_i b_i$ satisfying the two conditions in the corollary. Construct a digraph $D = (V, E)$ on the set of vertices $V = \{v_1, v_2, \ldots, v_m\}$ by letting $v_i v_j$ be a directed edge (for $i \neq j$) if and only if the colour of $a_i b_j$ in K_n^c differs from that of $a_i b_i$ and that of $a_j b_j$. By Corollary 11.1.48 the in-degree and the out-degree of every vertex of D exceeds $m/2$, implying, by Corollary 5.6.3, that D contains a directed Hamilton cycle $v_{\pi(1)} v_{\pi(2)} \ldots v_{\pi(m)} v_{\pi(1)}$, where $\pi = \pi(1), \pi(2), \ldots, \pi(m)$ is a permutation of $\{1, 2, \ldots, m\}$. The cycle $b_{\pi(1)} a_{\pi(1)} b_{\pi(2)} a_{\pi(2)} \ldots b_{\pi(m)} a_{\pi(m)} b_{\pi(1)}$ is clearly a PC Hamilton cycle in K_n^c, as needed.

In case $n = 2m+1$ is odd we fix a path $P = a_1 c_1 b_1$ of length 2, so that the edges $a_1 c_1$ and $c_1 b_1$ have distinct colours, choose a random perfect matching $a_2 b_2, \ldots, a_m b_m$ in the rest of the graph and show that with high probability there is a PC Hamilton cycle containing the path P and the matching by applying Corollary 5.6.3 as before. Since the details are almost identical to the ones for the even case, we omit them. This completes the proof of the theorem. □

11.2 Arc-Coloured Directed Multigraphs

In this section we show that surprisingly the problem to verify whether a 2-arc-coloured digraph has an alternating directed cycle is $\mathcal{NP}$-complete. We prove some sufficient conditions for a 2-arc-coloured digraph to contain an alternating directed cycle. These results are due to Gutin, Sudakov and Yeo [371]. We will obtain an original characterization of weakly eulerian arc-coloured directed multigraphs (see a definition below) and pose the problem to find the complexity of the eulerian trail problem for arc-coloured directed multigraphs.

In this section we assume that the arcs of directed multigraphs are coloured with c colours: $1, 2, \ldots, c$. We adapt notation and terminology of the previous section in the obvious way.

The problem of the existence of an alternating cycle in a 2-arc-coloured digraph (the **ADC problem**) generalizes the following two polynomially solvable problems: the existence of an alternating cycle in a 2-edge-coloured graph (see the previous section) and the existence of an even length cycle in a digraph (see Chapter 10). To see that the ADC problem generalizes the even cycle problem, replace every arc (x, y) of a digraph D by two vertex disjoint alternating paths of length three, one starting from colour 1 and the other from colour 2. Clearly, the obtained 2-edge-coloured digraph has an alternating cycle if and only if D has a cycle of even length. We will prove that the ADC problem is $\mathcal{NP}$-complete [371] by providing a transformation from the well-known 3-SAT problem (see Section 1.10) to the ADC problem. This is in contrast to the simple fact that the ADC problem restricted to bipartite 2-arc-coloured digraphs is polynomial time solvable.

To indicate that an arc (x, y) has colour $i \in \{1, 2, \ldots, c\}$ we will write $(x, y)_i$. For a vertex v in a c-arc-coloured directed multigraph D, $d_i^+(v)$ ($d_i^-(v)$) denotes the number of arcs of colour i leaving (entering) v, $i = 1, 2, \ldots, c$;

$$\delta_{mon}^0(v) = \min\{d_i^+(v), d_i^-(v) : \ i = 1, 2, \ldots, c\}.$$

The following parameter is of importance to us:

$$\delta_{mon}^0(D) = \min\{\delta_{mon}^0(v) : \ v \in V(D)\}.$$

Let $f(n)$ be the minimum integer such that every strongly connected 2-arc-coloured digraph D with n vertices and $\delta_{mon}^0(D) \geq f(n)$ has an alternating cycle. Similarly let $g(n)$ be the minimum integer such that every 2-arc-coloured digraph D with n vertices and $\delta_{mon}^0(D) \geq g(n)$ has an alternating cycle. We show below that $f(n) = \Theta(\log n)$ and $g(n) = \Theta(\log n)$.

By contrast with that, the corresponding function $f(n)$ for the even cycle problem does not exceed three (see Thomassen's even cycle theorem in Section 8.3). Using Theorem 3.2 in [702], one can show that the corresponding function $g(n)$ for the even cycle problem equals $\Theta(\log n)$. By Theorem 3.2 in [702], there exists a digraph H_n with n vertices and minimum out-degree at least $\frac{1}{2}\log n$ not containing even cycles. Let H'_n be the converse of H_n. Take vertex disjoint copies of H_n and H'_n and add all arcs from H'_n to H_n. The obtained digraph and the upper bound in Theorem 3.2 of [702] provide the estimate $\Theta(\log n)$.

A directed trail is **properly coloured (PC)** if its consecutive arcs differ in colour. In case of two colours, we speak of **alternating** trails. An arc-coloured directed multigraph D is **weakly eulerian** if the arc set of D can be partitioned into PC closed trails $T_1, \dots, T_k$. If D has a PC closed trail containing all arcs of D, then D is **eulerian**.

11.2.1 Complexity of the Alternating Directed Cycle Problem

The proof of the following proposition is left as a simple exercise (Exercise 11.27).

Proposition 11.2.1 *The ADC problem restricted to 2-arc-coloured bipartite digraphs is polynomial time solvable.*

In contrast with Proposition 11.2.1, we have the following:

Theorem 11.2.2 (Gutin, Sudakov, Yeo) *[371] The ADC problem is $\mathcal{NP}$-complete.*

Proof: To show that the ADC problem is $\mathcal{NP}$-complete, we transform the 3-SAT problem to the ADC problem (recall the definition of the 3-SAT problem from Section 1.10). Let $X = \{x_1, \dots, x_k\}$ be a set of variables, and let $\mathcal{F} = c_1 * c_2 * \dots * c_m$ be an instance of the 3-SAT problem such that every c_i has three literals and all of these are variables or negations of variables from X.

We construct a 2-arc-coloured digraph D which has an alternating cycle if and only if C is satisfiable. We use the same reduction as in [371], but rather than giving a formal definition of D, we describe its structure in the caption of Figure 11.7 and argue using this picture. This can easily be formalized to a precise description of D (see [371]). Based on the definition of D, it is not difficult to prove the following lemma which gives important structural properties of D (Exercise 11.26).

Lemma 11.2.3 *let C be an alternating directed cycle in D. Then the following holds:*

(a) C uses precisely one of the three paths of length two from c_j to c_{j+1} for $j = 1, 2, \dots, m$.
(b) For each $j = 1, 2, \dots, m$, the subpath $C[c_j, c_{j+1}]$ has length 2 and contains precisely one vertex from $\cup_{i=1}^{k}(V(P_i) \cup V(Q_i))$.
(c) C contains each of the vertices $c_1, c_2, \dots, c_m, c_{m+1}$ and in that order.
(d) If C uses a path $c_j u c_{j+1}$ such that $u \in V(P_i)$ ($u \in V(Q_i)$), then no other subpath of C of the type $c_q v c_{q+1}$, $q \neq j$, uses a vertex from $V(Q_i)$ ($V(P_i)$) and C contains the whole path Q_i (P_i) as a subpath.

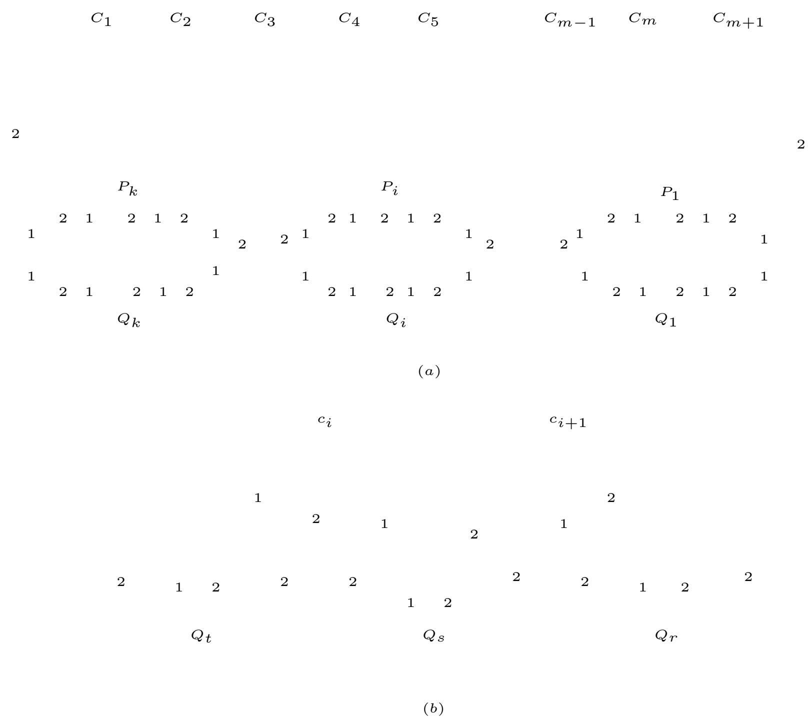


Figure 11.7 A schematic view of the digraph D. The digraph has one vertex c_j for each clause c_j in $\mathcal{F}$ and an extra vertex c_{m+1}. For each variable x_i, $i = 1, 2, \ldots, k$, D contains two alternating directed paths P_i, Q_i such that these start and end in the same vertices but are otherwise disjoint and both paths start and end with colour 1. Part (a) shows the way these structures are put together to form D. There is a unique arc from a pair P_i, Q_i to the next pair P_{i+1}, Q_{i+1} and this arc has colour 2. For every $j = 1, 2, \ldots, m$, c_j is joined to c_{j+1} by three paths of length 2. Part (b) of the figure shows a detailed picture of the three (c_jc_{j+1})-paths of length 2 when c_j is the clause $c_j = x_r + \overline{x}_s + x_t$. These paths are c_juc_{j+1}, where $u \in V(T_i)$, $T_i \in \{P_i, Q_i\}$. The first arcs of these paths are of colour 1. Furthermore the paths $P_1, \ldots, P_k, Q_1, \ldots, Q_k$ are chosen sufficiently long so that no vertex $u \in V(P_i) \cup V(Q_i)$ is used on two different paths of the type c_juc_{j+1}.

(e) C uses precisely one of the alternating directed paths P_i, Q_i for each $i = 1, 2, \ldots, k$ and it uses them in order of increasing i.

Lemma 11.2.4 *The digraph D has an alternating directed cycle if and only if $\mathcal{F}$ is satisfiable.*

Proof: Suppose C is an alternating directed cycle in D. By Lemma 11.2.3, the following is a truth assignment to $X = \{x_1, x_2, \ldots, x_k\}$. For each $i = 1, 2, \ldots, k$, if C uses P_i then put $x_i = 0$ otherwise (C uses Q_i by (e)) put $x_i = 1$. We claim that each clause c_j is satisfied by this assignment. By Lemma 11.2.3, the subpath of C from c_j to c_{j+1} has the form c_juc_{j+1} for some $u \in \cup_{i=1}^{k}(V(P_i) \cup V(Q_i))$. Let ℓ be the literal of c_j corresponding to u (that is, if $u \in P_i$, then by (d) and the definition of D, $\ell = x_i$ and if $u \in Q_i$, then $\ell = \overline{x}_i$). If $u \in V(P_i)$, then C uses the path Q_i and the truth assignment above will put $\ell = x_i = 1$. If $u \in V(Q_i)$, then C uses P_j and x_i is assigned the value 0, implying that $\ell = \overline{x}_i = 1$. This shows that the clause c_j is satisfied. Since this argument is valid for all clauses we see that the truth assignment described above satisfies $\mathcal{F}$.

Suppose now that $\mathcal{F}$ has a satisfying truth assignment $t = \{t_1, t_2, \dots, t_k\}$ (see Section 1.10). Then we can fix, for each clause c_j one literal l_j which is true according to this assignment. Let $\ell_1, \ell_2, \dots, \ell_m$ denote these fixed literals. Note that since t is a truth assignment, none of the chosen literals is the negation of another. By the construction of D there is a unique path $C_j = c_j u_j c_{j+1}$ which corresponds to the choice of ℓ_j (that is, $u_j \in P_i$ if $\ell_j = x_i$ and $u_j \in Q_j$ if $\ell_j = \overline{x}_i$). Furthermore if $\ell_{j_1} = \ell_{j_2}$, for some $j_1 \neq j_2$, then $u_{j_1} \neq u_{j_2}$. For each $i = 1, 2, \dots, k$ fix one of the paths P_i, Q_i as follows: if $\ell_r = x_i$ for some $r \in \{1, 2, \dots, m\}$, then $T_i = Q_i$, otherwise $T_i = P_i$. By the comment above this assignment of subpaths always chooses one subpath for each $i = 1, 2, \dots, k$.

Now it is easy to see that the following is an alternating directed cycle in D

$$C_1 C_2 \dots C_m T_1 T_2 \dots T_k c_1.$$

This completes the proof of the lemma. □

To complete the proof of Theorem 11.2.2, it suffices to observe that the digraph D can be constructed in polynomial time for any given instance of the 3-SAT problem. □

We do not know what the complexity of the ADC problem is when restricted to tournaments.

Problem 11.2.5 *[371] Does there exist a polynomial algorithm to check whether a 2-arc-coloured tournament has an alternating cycle?*

Figure 11.8 illustrates the difficulty of this problem. In contrast to the 'uncoloured' case, the 2-arc-coloured tournament T in Figure 11.8 has a unique alternating cycle, which is hamiltonian. Therefore, a reduction to 'short' alternating cycles may well be impossible.

Proposition 11.2.6 *The cycle C in the tournament T of Figure 11.8 is hamiltonian and consists of the matching of colour 1 from R to B and the matching of colour 2 from B to R. If we reverse any arc of colour 1 in C, we obtain a tournament with no alternating cycle.*

Proof: Exercise 11.31. □

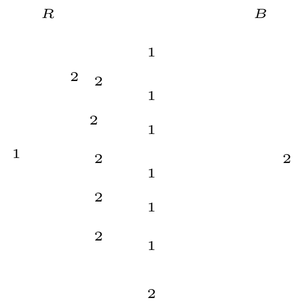


Figure 11.8 A 2-arc-coloured tournament with a unique alternating cycle C. All arcs within R (B) are of colour 1 (2). The cycle C is hamiltonian and consists of the matching of colour 1 from R to B and the matching of colour 2 from B to R. If we reverse any arc of colour 1 in C, we obtain a tournament with no alternating cycle.

11.2.2 The Functions $f(n)$ and $g(n)$

Since $f(n) \leq g(n)$ we will only prove a lower bound for $f(n)$ in Theorem 11.2.10 and an upper bound for $g(n)$ in Theorem 11.2.11.

Let $S(k)$ be the set of all sequences whose elements are from the set $\{1, 2\}$ such that neither 1 nor 2 appears more that k times in a sequence. We assume that the sequence without elements (i.e. the empty sequence) is in $S(k)$. We start with three technical lemmas. Their proofs are formulated as Exercises 11.28, 11.29, 11.30.

Lemma 11.2.7 *[371]* $|S(k)| = \binom{2(k+1)}{k+1} - 1$. □

Lemma 11.2.8 *[371] For every* $k \geq 1$,

$$\binom{2k}{k} < \frac{1}{\sqrt{\pi}} \frac{4^k}{\sqrt{k}}. \tag{11.5}$$

□

Let $d(n) = \lfloor \frac{1}{4} \log n + \frac{1}{8} \log \log n - a \rfloor$, where $a = \frac{5 - \log \pi}{8}$ (≤ 0.5).

Lemma 11.2.9 *[371]* $\binom{2(2d(n)+1)}{2d(n)+1} < n$, *for all* $n \geq 24$. □

Now we are ready to prove the following theorem by Gutin, Sudakov and Yeo [371].

Theorem 11.2.10 *For every integer* $n \geq 24$, *there exists a 2-arc-coloured strongly connected digraph* G_n *with* n *vertices and* $\delta^0_{mon}(G_n) \geq d(n)$ *not containing an alternating cycle.*

Proof: Let the vertex set of a digraph D_n be $S(2d(n))$ and let two vertices of D_n be connected if and only if one of them is a prefix of the other one. Moreover, if $x = (x_1, x_2, \ldots, x_p)$ and $y = (y_1, y_2, \ldots, y_q)$ are vertices of D_n, and x is a prefix of y (namely, $x_i = y_i$ for every $i = 1, 2, \ldots, p$), then the arc $a(x, y)$ between x and y has colour y_{p+1} and $a(x, y)$ is oriented from x to y if and only if $|\{j : \ j \geq p+1 \text{ and } y_j = y_{p+1}\}| \leq d(n)$.

The digraph D_n is strongly connected since the arc between a pair of vertices $x = (x_1, x_2, \ldots, x_p)$ and $y = (x_1, x_2, \ldots, x_p, x_{p+1})$ is oriented from x to y, and the arc between the empty sequence $\emptyset$ and a vertex v of D_n which is a sequence with $4d(n)$ elements is oriented from v to $\emptyset$ (every vertex of D_n belongs to a cycle containing $\emptyset$ and a vertex corresponding to a sequence of $4d(n)$ elements).

Let $x = (x_1, x_2, \ldots, x_p)$ be a vertex of D_n. It is easy to see that $d_1^+(x) \geq d(n)$. Indeed, if x contains at most $d(n)$ elements equal one, then $(x, x^r)_1$ is in D_n, where $r = 1, 2, \ldots, d(n)$ and x^r is x followed by r ones. If x contains $t > d(n)$ elements equal one, then $(x, y)_1$ is in D_n, where y is obtained from x by either adding at most $2d(n) - t$ ones or deleting more than $d(n)$ rightmost ones, together with 2's between them, from x.

Analogously, one can show that $d_1^-(x) \geq d(n)$. By symmetry, $\delta^0_{mon}(D_n) \geq d(n)$.

Now we prove that D_n contains no alternating cycle. Assume that D_n contains an alternating cycle C. The empty sequence $\emptyset$ is not in C as $\emptyset$ is adjacent with the vertices of the form $(i, \ldots)$ by arcs of colour $i \in \{1, 2\}$, but the vertices of the form $(1, \ldots)$ are not adjacent with the vertices of the form $(2, \ldots)$. Analogously, one can prove that the vertices (1) and (2) are not in C. In general, after proving that C has no vertex with p elements, we can show that C has no vertex with $p+1$ elements.

By Lemma 11.2.7, D_n has $b(n) = \binom{2(2d(n)+1)}{2d(n)+1} - 1$ vertices. By Lemma 11.2.9, $b(n) < n$. Now we append $n - b(n)$ vertices along with arcs to D_n to obtain a digraph G_n with $\delta^0_{mon}(G_n) \geq d(n)$. Take a vertex $x \in D_n$ with $4d(n)$ elements. We add $n - b(n)$ copies of x to D_n such that every copy has the same out- and in-neighbours of each colour as x. The vertex x and its copies form an independent set of vertices.

The construction of G_n implies that $\delta^0_{mon}(G_n) \geq d(n)$, G_n is strongly connected and G_n has no alternating cycle, by the same reason as D_n. $\square$

Now we are ready to prove an upper bound[2] for $g(n)$.

Theorem 11.2.11 *Let D=(V,A) be a 2-arc-coloured digraph on $|V| = n$ vertices. If $d_i^+(v) \geq \log n - 1/3 \log\log n + O(1)$ for every $i = 1, 2$ and $v \in V$, then D contains an alternating cycle.*

[2] As we mentioned in the footnote just before Lemma 10.6.13 the bound of the lemma can be slightly improved. Hence the bound for $g(n)$ can also be improved slightly.

Proof: Without loss of generality assume that $d_i^+(v) = k$ for all $v \in V$ (k will be defined later), otherwise simply remove extra arcs. For each vertex $v \in V$ and each colour $i = 1, 2$, let

$$B_v^i = \{u \in V : \ (v, u) \text{ is an arc of colour } i\}.$$

The size of each of the sets B_v^i is equal to k, thus they form a k-uniform hypergraph H with n vertices and $2n$ edges. Let $k = \log n - 1/3 \log\log n + b$, where b is a constant. Then it is easy to see that by choosing b large enough we get that $ck^{1/3}2^k > 2n$. By Lemma 10.6.13, our hypergraph H is 2-colourable. By taking a 2-colouring of H we get a partition $V = X \cup Y$ such that B_v^i intersects both X and Y for every $i = 1, 2$ and $v \in V$. Let D_1 be a subdigraph of D which contains only arcs of colour 1 from X to Y and arcs of colour 2 from Y to X. The out-degree of every vertex in D_1 is positive, since all sets B_v^i intersect both X and Y. Therefore D_1 contains a cycle, which is alternating by the construction of D_1. $\square$

11.2.3 Weakly Eulerian Arc-Coloured Directed Multigraphs

1 2 2

2 1 1

Figure 11.9 Weakly eulerian non-eulerian 2-arc-coloured digraph.

The following theorem yields a characterization of weakly eulerian arc-coloured directed multigraphs. Due to Theorem 11.1.2, every connected weakly eulerian edge-coloured multigraph is eulerian (the definition of weakly eulerian edge-coloured multigraphs is analogous to that of arc-coloured directed multigraphs). This is in contrast to the fact that not every connected weakly eulerian arc-coloured directed multigraph is eulerian. For example, let C and Z be a pair of 2-arc-coloured alternating directed cycles with only one common vertex x and $1 = \chi(xx_C^+) \neq \chi(xx_Z^+) = 2$; see Figure 11.9. The union H of C and Z is weakly eulerian, but H has no PC eulerian trail.

The proof of the following theorem is similar to that of Theorem 11.1.2 and left as Exercise 11.34.

Theorem 11.2.12 *An arc-coloured directed multigraph D is weakly eulerian if and only if $d^+(x) = d^-(x)$ for every vertex x in D, and for every vertex x in D and every colour i, we have*

$$d_i^-(x) \leq \sum_{j \neq i} d_j^+(x).$$

Neither characterization nor complexity is known so far for the eulerian trail problem in arc-coloured directed multigraphs.

Problem 11.2.13 *Find the complexity of checking whether an arc-coloured directed multigraph is eulerian.*

For the case of just two colours the following simple transformation by Fleischner [240] can be applied. Let D be a 2-arc-coloured directed multigraph. Split every vertex v with $\delta^0(v) > 0$ into a pair v', v'' of vertices such that v' 'inherits' all red arcs entering v and all blue arcs leaving v, and v'' 'inherits' all blue arcs entering v and all red arcs leaving v. By disregarding all colours in the obtained 2-arc-coloured directed multigraph, we yield the directed multigraph H. Clearly, D is eulerian if and only if H is eulerian. Some sufficient conditions for an arc-coloured directed multigraph to be eulerian are given in [240].

11.3 Hypertournaments

Given two integers n and k, $n \geq k > 1$, a ***k*-hypertournament** T on n vertices is a pair (V, A), where V is a set of **vertices**, $|V| = n$ and A is a set of k-tuples of vertices, called **arcs**, so that for any k-subset S of V, A contains exactly one of the $k!$ k-tuples whose entries belong to S. That is, T may be thought of as arising from an orientation (being a fixed permutation of vertices) of the hyperedges of the complete k-uniform hypergraph on n vertices. Clearly, a 2-hypertournament is merely a tournament.

As an example of a 3-hypertournament, let L have vertex set $V(L) = \{1, 2, 3, 4\}$ and arc set $A(L) = \{(1, 2, 3), (1, 2, 4), (1, 4, 3), (4, 3, 2)\}$. The four arcs of L are orientations of sets $\{1, 2, 3\}, \{1, 2, 4\}$, $\{1, 4, 3\}$ and $\{2, 3, 4\}$, respectively.

Hypertournaments have been studied by a number of authors (see, e.g., the papers [32] by Assous, [92, 93] by Barbut and Bialostocki, [117] by Bialostocki, [276] by Frankl, [374] by Gutin and Yeo, [552, 553] by Marshall, [599] by Pan, Zhou and Zhang and [759] by Zhou, Yao and Zhang). Reid [630, Section 8] describes several results on hypertournaments obtained by the above authors and poses some interesting problems on the topic. In particular, he raises the problem of extending the most important results on tournaments to hypertournaments.

In this section based on the results of Gutin and Yeo in [374] and Zhou, Yao and Zhang in [759], we give extensions of three of the most basic theorems on tournaments: every tournament has a Hamilton path (Rédei's theorem), every strong tournament has a Hamilton cycle (Camion's theorem), and Landau's theorem, Theorem 8.7.1, on out-degree sequences of tournaments. It turns out that every k-hypertournament on n ($> k$) vertices has a Hamilton path and every strong k-hypertournament on $n \geq k + 2 \geq 5$ vertices contains a Hamilton cycle. We also describe, for every $k \geq 3$, a strong

k-hypertournament on $k+1$ vertices which has no Hamilton cycle. We consider the complexity of the Hamilton cycle problem for k-hypertournaments and note that the problem remains polynomial time solvable when $k=3$ and becomes $\mathcal{NP}$-complete for every fixed integer $k \geq 4$. As follows from Theorem 11.3.4, deciding strong connectivity for hypertournaments is already $\mathcal{NP}$-complete. Interestingly enough, Landau's theorem and the Harary-Moser theorem, Theorem 8.7.2, on out-degree sequences of all tournaments and all strong tournaments have direct extension to hypertournaments.

Let $T=(V,A)$ denote a k-hypertournament T on n vertices. A **path** in T is a sequence $v_1a_1v_2a_2v_3\ldots v_{t-1}a_{t-1}v_t$ of distinct vertices $v_1,v_2,\ldots,v_t$, $t\geq 1$, and distinct arcs $a_1,\ldots,a_{t-1}$ such that v_i precedes v_{i+1} in a_i, $1\leq i\leq t-1$. A **cycle** in T is a sequence $v_1a_1v_2a_2v_3\ldots v_{t-1}a_{t-1}v_ta_tv_1$ of distinct vertices $v_1,v_2,\ldots,v_t$ and distinct arcs $a_1,\ldots,a_t$, $t\geq 1$, such that v_i precedes v_{i+1} in a_i, $1\leq i\leq t$ ($a_{t+1}=a_1$). The above definitions of a path and cycle in a hypertournament are similar to the corresponding definitions of a path and cycle in a hypergraph.

For a path or cycle Q, $V(Q)$ and $A(Q)$ denote the set of vertices (v_i's above) and the set of arcs (a_j's above), respectively. For a pair of vertices v_i and v_j of a path or cycle Q, $Q[v_i,v_j]$ denotes the subpath of Q from v_i to v_j (which can be empty). A path or cycle Q in T is **hamiltonian** if $V(Q)=V(T)$. The 3-hypergraph L considered in the beginning of this section has a Hamilton path $1,(1,2,3),2,(1,2,4),4,(1,4,3),3$. A hypertournament T is **hamiltonian** if it has a Hamilton cycle. A path from x to y is an *(x,y)-path.* A hypertournament T is **strong** if T has an (x,y)-path for every (ordered) pair x,y of distinct vertices in T. The hypertournament L is not strong, since there is no $(2,1)$-path in L. This, in particular, means that L is not hamiltonian.

We also consider paths and cycles in digraphs which will be denoted as sequences of the corresponding vertices.

The **out-degree** $d^+(v)$ of a vertex v in a hypertournament T is the number of arcs in T in which v is the last vertex. The **out-degree sequence** of $T=(V,A)$ is the non-decreasing sequence $s_1,s_2,\ldots,s_n$ of non-negative integers such that $\{s_1,s_2,\ldots,s_n\}=\{d^+(v):\ v\in V\}$. For a pair of distinct vertices x and y in T, $A_T(x,y)$ denotes the set of all arcs of T in which x precedes y. Clearly, for all distinct $x,y\in V(T)$,

$$|A_T(x,y)|+|A_T(y,x)|=\binom{n-2}{k-2}. \tag{11.6}$$

11.3.1 Out-Degree Sequences of Hypertournaments

It turns out that Landau's theorem on out-degree sequences of tournaments can be directly extended to hypertournaments. Similarly, one can extend the Harary-Moser theorem on out-degree sequences of strong tournaments. These extensions were proved by Zhou, Yao and Zhang [759].

Theorem 11.3.1 *[759] Given two non-negative integers n and k with $n \geq k \geq 2$, a non-decreasing sequence $s_1, s_2, \ldots, s_n$ of non-negative integers is the out-degree sequence of some k-hypertournament if and only if for each j, $k \leq j \leq n$,*

$$\sum_{i=1}^{j} s_i \geq \binom{j}{k},$$

with equality holding when $j = n$. □

Theorem 11.3.2 *[759] A sequence $s_1 \leq s_2 \leq \ldots \leq s_n$ of non-negative integers with $n > k \geq 2$ is the out-degree sequence of some strong k-hypertournament if and only if for each j, $k \leq j \leq n-1$,*

$$\sum_{i=1}^{j} s_i > \binom{j}{k}$$

and

$$\sum_{i=1}^{n} s_i = \binom{n}{k}.$$

□

11.3.2 Hamilton Paths

Assume, in this subsection, that $k \geq 2$. Clearly, no k-hypertournament with precisely $k(\geq 3)$ vertices has a Hamilton path. However, all other hypertournaments have Hamilton paths:

Theorem 11.3.3 *Every k-hypertournament with n $(> k)$ vertices contains a Hamilton path.*

Proof: Let $T = (V, A)$ be a k-hypertournament T on n vertices 1,2,…,n. We consider the cases $k = n-1$ and $k < n-1$ separately.

Case 1: $\boldsymbol{k = n-1}$. We proceed by induction on $k \geq 2$. Clearly, this theorem holds for $k = 2$. Hence, suppose that $k \geq 3$. Assume (by relabelling the vertices, if necessary) that T contains the arc $a = (23\ldots n)$. Let b be the arc of T whose vertices are $1, 2, \ldots, n-1$ (in some order). Consider the $(k-1)$-hypertournament $T' = (V', A')$ obtained from T by deleting the arc a, deleting n from the arcs in $A - \{a, b\}$, and finally deleting 1 from b. So, $V' = \{1, 2, \ldots, n-1\}$, $A' = \{e' : \ e' \text{ is } e \text{ without } n, \ e \in A - \{a, b\}\} \cup \{b'\}$, where b' is b without the vertex 1. By the induction hypothesis, T' has a Hamilton path $x_1 a'_1 x_2 a'_2 \ldots a'_{n-2} x_{n-1}$. This path corresponds to the path $Q = x_1 a_1 x_2 a_2 \ldots a_{n-2} x_{n-1}$ in T. Clearly, $\{x_1, \ldots, x_{n-1}\} = \{1, \ldots, n-1\}$ and $A - \{a_1, \ldots, a_{n-2}\}$ consists of the arc a and another arc c.

If $x_{n-1} \neq 1$, then Qan is a Hamilton path in T. Hence from now on assume that $x_{n-1} = 1$. Consider two subcases.

Subcase 1.1 $c \neq b$: If the last vertex of c is n, then Qcn is a Hamilton path in T. Otherwise, x_j is the last vertex of c for some $j \leq n-1$. If $j > 1$ we replace a_{j-1} by anc in Q in order to obtain a Hamilton path in T. If $j = 1$, then ncQ is a Hamilton path in T.

Subcase 1.2 $c = b$: If $c \neq (x_{n-1}x_{n-2} \dots x_1)$ so that x_i precedes x_{i+1}, for some i, $1 \leq i \leq n-2$, in c, then $P = Q[x_1, x_i]cQ[x_{i+1}, x_{n-1}]$ is a path in T. Since $a_i \neq b$, one can construct a Hamilton path in T from P as in Subcase 1.1. If $c = (x_{n-1}x_{n-2} \dots x_1)$, then $Q[x_2, x_{n-1}]cx_1an$ is a Hamilton path in T.

Case 2: $k < n - 1$. We proceed by induction on $n \geq 4$. The case $n = 4$ (and, hence, $k = 2$) is easy to verify (it also follows from Rédei's theorem). Therefore, suppose that $n \geq 5$. Consider the new k-hypertournament T'' obtained from T by deleting the vertex n along with all arcs in A containing n. T'' has a Hamilton path because of either Case 1 if $n = k - 2$ or the induction hypothesis, otherwise.

Let $P = x_1a_1x_2a_2 \dots a_{n-2}x_{n-1}$ be a Hamilton path in T''. If T has an arc $a \in A_T(x_{n-1}, n)$, then Pan is a Hamilton path in T. Suppose that $A_T(x_{n-1}, n) = \emptyset$. Then either $\cup_{l=1}^{n-1} A_T(x_l, n) = \emptyset$, or there is an i so that $\cup_{l=i+1}^{n-1} A_T(x_l, n) = \emptyset$ and T contains an arc b where x_i precedes n. In the first case, ncP is a Hamilton path in T, where c is an arc of T containing both x_1 and n. In the second case, $P[x_1, x_i]bndP[x_{i+1}, x_{n-1}]$ is a Hamilton path in T, where d is an arc of T containing both x_{i+1} and n and distinct from b. □

11.3.3 Hamilton Cycles

Clearly, every hamiltonian hypertournament is strong. However, for every $k \geq 3$, there exists a strong k-hypertournament with $n = k + 1$ vertices which is not hamiltonian. Indeed, let the $(n-1)$-hypertournament H_n have vertex set $\{x_1, \dots, x_n\}$ and arc set $\{a_1, a_2, \dots, a_n\}$, where $a_1 = (x_2x_3 \dots x_{n-2}x_nx_{n-1})$, $a_2 = 3$, $a_3 = (x_1x_2x_4x_5 \dots x_n)$, $a_4 = (x_2x_3x_1x_5x_6 \dots x_n)$, and

$$a_i = (x_1x_2 \dots x_{i-4}x_{i-3}x_{i-1}x_{i-2}x_{i+1}x_{i+2} \dots x_n) \text{ for } 5 \leq i \leq n.$$

The hypertournament H_n is strong (Exercise 11.35). However, H_n is not hamiltonian. To prove that, assume that H_n has a Hamilton cycle C. We will try to construct C starting from the vertex x_n. Since a_1 is the only arc which has a vertex that succeeds x_n, C has the form $x_na_1x_{n-1} \dots$. Since a_n is the only arc which has a vertex different from x_n that succeeds x_{n-1}, $C = x_na_1x_{n-1}a_nx_{n-2} \dots$. Continuing this process, we obtain that $C = x_na_1x_{n-1} \dots x_4a_5x_3 \dots$. The only arc where x_3 precedes x_1 or x_2 is a_4. Hence, $C = x_na_1x_{n-1} \dots x_4a_5x_3a_4x_1 \dots$. Now we need to include x_2, a_3 and a_2 into C. However, this is impossible because only one of the arcs a_3, a_2 contains x_2.

In the proof in [374] that every strong k-hypertournament with n vertices, where $3 \leq k \leq n-2$, is hamiltonian the following notion is of great importance. The **majority digraph** $D_{major}(H)$ of a k-hypertournament H with n vertices has the same vertex set V as H and, for every pair x, y of distinct vertices in V, the arc xy is in $D_{major}(H)$ if and only if $|A_H(x,y)| \geq |A_H(y,x)|$ (or, by (11.6), $|A_T(x,y)| \geq \frac{1}{2}\binom{n-2}{k-2}$). Obviously, $D_{major}(H)$ is a semicomplete digraph. Figure 11.10 shows the majority digraph $D_{major}(L)$ of the 3-hypertournament L with vertex set $V(L) = \{1,2,3,4\}$ and arc set $A(L) = \{(1,2,3),(1,2,4),(1,4,3),(4,3,2)\}$.

1 2

4 3

Figure 11.10 The majority graph $D_{major}(L)$ of L.

Since the proof of the following interesting result is rather lengthy, we do not provide it here.

Theorem 11.3.4 (Gutin and Yeo) *[374] Every strong k-hypertournament with n vertices, where $3 \leq k \leq n-2$, contains a Hamilton cycle.* □

We know that the Hamilton cycle problem for 2-hypertournaments, i.e. tournaments is polynomial time solvable (see Chapter 5). It turns out that the k-hypertournament hamiltonicity problem remains polynomial time solvable for $k = 3$, but becomes $\mathcal{NP}$-complete for every fixed $k \geq 4$.

Let $H = (V, A)$ be a k-hypertournament, $A = \{a_1, \ldots, a_m\}$. Associate with H the following arc-coloured directed multigraph $D(H)$: the vertex set of $D(H)$ is V; for distinct vertices $x, y \in V$, $D(H)$ has the arc xy of colour i if and only if $a_i \in A_H(x,y)$. Clearly, H contains a path from a vertex x to another vertex y if and only if $D(H)$ has a path P from x to y such that no two arcs in P have the same colour.

Theorem 11.3.5 *[374] The Hamilton cycle problem is solvable in polynomial time for the class of 3-hypertournaments.*

Proof: Let H be a 3-hypertournament. We may assume that $n \geq 5$, since the case when $n \leq 4$ can be checked in constant time. By Theorem 11.3.4, it suffices to prove that one can check the existence of a path, in H, from a vertex x to another vertex y in polynomial time. Construct the arc-coloured directed multigraph $D(H)$ as above. We prove that H has a path from x to

y if and only if $D(H)$ has some (x, y)-path. Clearly, if H has a path from x to y, then $D(H)$ contains such a path. Suppose that $D(H)$ has a path $Q = x_1 \ldots x_p$ from $x = x_1$ to $y = x_p$. If Q has no arcs of the same colour, then Q corresponds, in the obvious way, to an (x, y)-path of H. Suppose that Q contains arcs of the same colour. This means that there exist a subscript i and an integer j such that the arcs $x_{i-1}x_i$ and x_ix_{i+1} have the same colour j (these two are the only arcs of colour j which can be in Q). We can replace Q by the path $Q[x_1, x_{i-1}]Q[x_{i+1}, x_p]$. Continuing this process, we obtain a new path, in $D(H)$, from x to y without repetition of colours. The new path corresponds to an (x, y)-path in H. □

Theorem 11.3.6 (Gutin and Yeo) *[374] Let $k \geq 4$. The Hamilton cycle problem for k-hypertournaments is $\mathcal{NP}$-complete.* □

The proof of this theorem in [374] is considerably more difficult and lengthy. It reduces the 3-SAT problem into the Hamilton cycle problem for 4-hypertournaments.

11.4 Application: Alternating Hamilton Cycles in Genetics

In [200, 201] Dorninger considers Bennett's model (see Bennett's book [104] and the papers [423, 424] by Heslop-Harrison and Bennett) of chromosome arrangement in a cell of an eukaryotic organism. In [201], the case of even number, n, of chromosomes is studied. We consider here only this case as it is more interesting. Every individual chromosome consists of a long arm and a short arm, which are linked at the so-called centromere. At a certain stage of cell division, which is of interest to biologists, the arms of n chromosomes form an n-angle star whose internal points are the centromeres (see Figure 11.11) and external points created by the arms of 'adjacent' chromosomes. To find

Figure 11.11 Chromosome arrangement.

out the order of the centromeres, Bennett [104] suggested that the external points are formed by the most similar size arms. Bennett and Dorninger (see [201]) generalized the notion of similarity to so-called k-similarity and Dorninger [201] analyzed the consistency of this generalized notion. Let us consider the following graph-theoretic model of this biological system. Let s_i and l_i denote the short and long arm of chromosome number i. Let the chromosomes be labeled 1,2,…,n in such a way that s_i is longer than s_j if $i < j$, and let π be a permutation of $1, 2, \ldots, n$ such that $l_{\pi(i)}$ is longer than $l_{\pi(j)}$ if $i < j$.

We call two short arms s_i and s_j (long arms $l_{\pi(i)}$ and $l_{\pi(j)}$), $i \neq j$, ***k*-similar** if $|i - j| \leq k$. In this way, for $k = 1$, we obtain the original Bennett's notion of 'most similar size'. Let $G(n, k, \pi)$ be a 2-edge-coloured multigraph with vertex set $\{1, 2, \ldots, n\}$. The blue (red) subgraph $G_1(n, k, \pi)$ ($G_2(n, k, \pi)$) of $G(n, k, \pi)$ consists of edges pq ($p \neq q$) such that s_p and s_q (l_p and l_q) are k-similar. (See Figure 11.12.)

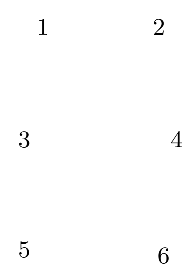


Figure 11.12 The 2-edge-coloured graph $G(6, 2, \tau)$, where $\tau(1) = 2,\ \tau(2) = 1,\ \tau(3) = 4,\ \tau(4) = 3,\ \tau(5) = 5,\ \tau(6) = 6$. The blue edges are shown by ordinary lines. The red edges are indicated by fat lines.

According to one of Bennett's assumptions, $G(n, k, \pi)$ has an alternating Hamilton cycle. Dorninger [201] analyzed when $G(n, k, \pi)$ has an alternating Hamilton cycle for every permutation π. Clearly, for $k = 1$, the 2-edge-coloured multigraph is a collection of $t \geq 1$ alternating cycles and, when $t \geq 2$, Bennett's assumption does not hold. Dorninger [201] proved that $G(n, 2, \pi)$ has an alternating Hamilton cycle for every π provided $n \leq 12$. He [201] also showed that for every $n \geq 14$ there exists a permutation π such that $G(n, 2, \pi)$ has no alternating Hamilton cycle. Yeo (private communication, April, 1999) proved that the alternating Hamilton cycle problem for the graphs $G(n, 2, \pi)$ is $\mathcal{NP}$-hard. Interestingly enough, $G(n, 3, \pi)$ contains an alternating Hamilton cycle for every permutation π. Thus, the notion of 3-similarity seems to be most consistent with Bennett's assumptions.

In the rest of this section we will prove the following two results:

Theorem 11.4.1 *[201] For every even positive integer $n \leq 12$ and every permutation π of 1,2,…,n, the 2-edge-coloured multigraph $G(n, 2, \pi)$ has an alternating Hamilton cycle.*

Theorem 11.4.2 (Dorninger) *[201] For every even positive integer n and every permutation π of 1,2,...,n, the 2-edge-coloured multigraph $G(n, 3, \pi)$ has an alternating Hamilton cycle.*

11.4.1 Proof of Theorem 11.4.1

In this subsection, which contains certain proofs suggested by Yeo (private communication, April, 1999), we consider multigraphs $G = G(n, 2, \pi)$. We recall that $V(G) = V(G_1) = V(G_2) = \{1, 2, \ldots, n\}$, $E(G_1) = \{ij : |i - j| \leq 2\}$, and $E(G_2) = \{\pi(i)\pi(j) : |i - j| \leq 2\}$ (see Figure 11.12). Clearly, every alternating cycle factor $\mathcal{F}$ of G is the union of a perfect matching F_1 of G_1 and a perfect matching F_2 of G_2. We write $\mathcal{F} = C(F_1, F_2)$.

Suppose that $e = ij$ and $f = pq$ are in F_1 and e and f belong to two distinct cycles X and Y of $\mathcal{F}$. Suppose also that $i < j$, $p < q$ and edges ip and jq are in G_1. If we delete e and f in F_1 and add edges ip and jq, we obtain a new perfect matching F_1' of G_1. Observe that $C(F_1', F_2)$ has one less cycle than $C(F_1, F_2)$ since the vertices of X and Y form a new alternating cycle Z. We call F_1' the **(e, f)-switch** of F_1; the operation to obtain F_1' from F_1 is a **switch** (or, the **(e, f)-switch**).

Let $S = \{\{2t - 1, 2t\} : t = 1, 2, \ldots, n/2\}$ and $L = \{\pi(2t - 1)\pi(2t) : t = 1, 2, \ldots, n/2\}$. Clearly, S and L are perfect matchings in G_1 and G_2, respectively.

Lemma 11.4.3 *Let $C(S, L)$ contain m cycles. There is a sequence of switches of edges in S, such that the resulting perfect matching F of G_1, has the property that $C(F, L)$ has at most $\lfloor (m + 1)/2 \rfloor$ cycles. Furthermore, given any cycle C_h in $C(S, L)$ we may choose F, such that all cycles in $C(F, L)$, except possibly C_h, have length at least 4.*

Proof: Let $C(S, L)$ consist of cycles $C_1, C_2, \ldots, C_m$. Let $e_i = \{2r_i - 1, 2r_i\}$ be an edge of C_i, such that r_i is minimum. Assume that the cycles $C_1, C_2, \ldots, C_m$ are labelled such that $1 = r_1 < r_2 < \ldots < r_m$. Define q_i to be the maximum number such that $\{2r_i - 1, 2r_i\}, \{2r_i + 1, 2r_i + 2\}, \ldots, \{2q_i - 1, 2q_i\}$ belong to C_i, for every $i = 1, 2, \ldots, m$. Observe that $1 = r_1 \leq q_1 < r_2 \leq q_2 < \ldots < r_m \leq q_m = n$.

Fix $h \in \{1, 2, \ldots, m\}$. We will now prove that by doing switches every cycle, except possibly C_h, can be merged with another cycle. We perform the switches recursively in the following way. While there is a cycle, C_i with $i < h$, which has not been merged to another cycle do the following: choose i to be the minimum such index and perform the $(\{2q_i - 1, 2q_i\}, \{2q_i + 1, 2q_i + 2\})$-switch. While there is some cycle, C_i with $i > h$, which has not been merged to another cycle do the following: choose i to be the maximum such index and perform the $(\{2r_i - 3, 2r_i - 2\}, \{2r_i - 1, 2r_i\})$-switch. Note that all the above switches use distinct edges.

Since every cycle, except possibly C_h, is merged to another cycle, we must have performed at least $\lfloor m/2 \rfloor$ merges. Therefore there are at most

$m - \lfloor m/2 \rfloor = \lfloor (m+1)/2 \rfloor$ cycles left, which proves the first part of the theorem. The second part follows immediately from the above construction. □

Theorem 11.4.1 follows from the next lemma.

Lemma 11.4.4 *If $C(S, L)$ has at most six cycles, then G has an alternating Hamilton cycle.*

Proof: By the previous lemma, the alternating cycle factor $C(F, L)$ has at most three cycles. Furthermore we may assume that all cycles in $C(F, L)$ have length at least 4, except possibly the cycle containing the vertex $\pi(1)$. If $C(F, L)$ consists of a unique cycle, then we are done. Assume that $C(F, L)$ has three or two cycles. Label them D_1, D_2, D_3 (or D_1, D_2) similarly to that in the proof of Lemma 11.4.3. Let $f_i = \pi(2r_i - 1)\pi(2r_i)$ be an edge of D_i, such that r_i is minimum. Assume that the cycles D_1, D_2, D_3 are labelled such that $1 = r_1 < r_2 < r_3$. Let $f'_i = \pi(2r_i - 3)\pi(2r_i - 2)$ for $i \geq 2$.

Note that all cycles except possibly D_1 have length at least 4. If $C(F, L)$ has two cycles (D_1 and D_2), then construct the (f'_2, f_2)-switch M of L. Clearly, $C(F, M)$ consists of a unique cycle. Assume that $C(F, L)$ has three cycles, D_1, D_2, D_3. Perform the (f'_2, f_2)-switch. If $f'_3 \neq f_2$ then perform the (f'_3, f_3)-switch, which gives the desired cycle. If $f'_3 = f_2$ then let $g = \pi(2j - 1)\pi(2j)$ be the edge of minimum $j > r_3$ which does not lie in D_3, and let $g' = \pi(2j - 3)\pi(2j - 2)$. Now perform the (g', g)-switch, which gives the desired cycle. □

11.4.2 Proof of Theorem 11.4.2

In this subsection, we follow [201]. We consider multigraphs $G = G(n, 3, \pi)$. We recall that $V(G) = V(G_1) = V(G_2) = \{1, 2, \ldots, n\}$, $E(G_1) = \{ij : |i - j| \leq 3\}$, and $E(G_2) = \{\pi(i)\pi(j) : \ |i - j| \leq 3\}$. We use the same notation as in the previous subsection, in particular, the notation $C_1, C_2, \ldots, C_m$ and $e_1, e_2, \ldots, e_m$ remain valid. Let G^k be the subgraph of G induced by the vertices of the cycles $C_1, C_2, \ldots, C_k$. Let $L^k = L \cap E(G_2^k)$.

We show that, for every $k \geq 1$, there is a perfect matching F^k of G_1^k such that $C(F^k, L^k)$ consists of a single cycle. Clearly, the assertion implies Theorem 11.4.2. Trivially, the assertion is true for $k = 1$. So let us assume that the assertion holds for every $i \leq k - 1$. Let $e_k = \{s + 1, s + 2\}$, where s is an appropriate even integer. Consider the following three cases.

Case 1: The edge $e = \{s, s - j\}$, where $j = 1$ or 2, is in F^{k-1}. Then, the desired F^k is the (e_k, e)-switch of $F^{k-1} + e_k$. Indeed, $C(F^k, L^k)$ consists of a single cycle.

Case 2: The edges $e' = \{s, s - 3\}$ and $e'' = \{s - 1, s - 2\}$ are in F^{k-1}. Let M_1 (M_2) be a perfect matching of G_1^{k-1} obtained from F^{k-1} by replacing edges e', e'' with $\{s, s - 1\}, \{s - 3, s - 2\}$ ($\{s, s - 2\}, \{s - 3, s - 1\}$).

Clearly, for some $i \in \{1, 2\}$, $C(M_i, L^{k-1})$ consists of a single cycle H. Since either $\{s, s-1\}$ or $\{s, s-2\}$ is in H, we can apply the transformation of Case 1 to the appropriate matching M_i.

Case 3: The edges $e' = \{s, s-3\}$ and $e'' = \{s-1, s-4\}$ are in F^{k-1}. Then $e = \{s-2, s-5\}$ must be in F^{k-1}. Let H be the single cycle of $C(F^{k-1}, L^{k-1})$. Consider the following two subcases.

Subcase 3.1: The vertices of e and e' are in the cyclic order $s-5, s-2, s-3, s$ in H. Replacing e and e' with $\{s-5, s-3\}$ and $\{s, s-2\}$, we obtain a perfect matching M of G_1^{k-1} such that $C(M, L^{k-1})$ consists of a single cycle. Since $\{s, s-2\} \in M$, we can apply the transformation of Case 1 to M.

Subcase 3.2: The vertices of e and e' are in the cyclic order $s-2, s-5, s-3, s$ in H. If e'' belongs to $H[s-5, s-3]$, then by replacing e, e', e'' with three edges, one of which is $\{s, s-1\}$, we obtain a perfect matching M of G_1^{k-1} such that $C(M, L^{k-1})$ consists of a single cycle. Since $\{s, s-1\} \in M$, we can apply the transformation of Case 1 to M. If e'' belongs to $H[s, s-2]$, then by replacing e, e', e'' with three edges, one of which is $\{s, s-2\}$, we obtain a perfect matching M of G_1^{k-1} such that $C(M, L^{k-1})$ consists of a single cycle. Since $\{s, s-2\} \in M$, we can apply the transformation of Case 1 to M. □

11.5 Exercises

11.1. Prove Proposition 11.1.1. Hint: use Häggkvist's transformation as well as Theorem 5.0.1, Proposition 9.2.1 and Theorem 9.2.3.

11.2. **(−)** Deduce from Theorem 11.1.2 that an undirected multigraph G has an eulerian trail if G is connected and each vertex of G is of even degree.

11.3. Prove that Pevzner's algorithm described after Theorem 11.1.2 is correct.

11.4. **(−)** Every eulerian digraph has a cycle (unless it is the trivial digraph with one vertex). Show that the corresponding claim is not valid for alternating trails and cycles in 2-edge-coloured graphs.

11.5. Let G be a connected 2-edge-coloured graph. Let $V(G) = X + Y$ such that $d_1(x) = d_2(x)$ for every $x \in X$, and $d_1(y) = d_2(y) - 1$ for every $y \in Y$. What is the minimum number of edge-disjoint alternating trails to cover $E(G)$?

11.6. Prove Corollary 11.1.7.

11.7. **Every bridgeless graph G has an M-alternating cycle for a given perfect matching M of G.** Let M be a perfect matching in a graph G. Using Theorem 11.1.6 prove that, if no edge of M is a bridge of G, then G has a cycle whose edges are taken alternatively from M and $G - M$ (Grossman and Häggkvist [335]).

11.8. (+) Let G be a 2-edge-coloured eulerian graph so that all monochromatic degrees are odd. Using Theorem 11.1.6 demonstrate that G has an alternating cycle (Grossman and Häggkvist [335]).

11.9. Prove Proposition 11.1.10.

11.10. Prove Proposition 11.1.15. Hint: see Exercise 1.30.

11.11. Prove Theorem 11.1.16 using the BD-correspondence and Corollary 5.6.3.

11.12. Deduce Theorem 11.1.16 from Theorem 11.1.17.

11.13. Show that the conditions of Theorem 11.1.25 are necessary. Hint: it is similar to the remark after Theorem 11.1.19.

11.14. Derive Theorem 11.1.19 from Theorem 11.1.25. Hint: you may use the DHM-construction.

11.15. (+) Prove Theorem 11.1.28.

11.16. (+) Prove Theorem 11.1.30.

11.17. Give a direct proof of Corollary 11.1.24 (Bang-Jensen, Gutin and Yeo [73]).

11.18. Prove Lemma 11.1.34.

11.19. Prove Theorem 11.1.38.

11.20. Prove Theorem 11.1.41.

11.21. Prove Theorem 11.1.42.

11.22. Check that G_6 introduced after Theorem 11.1.42 is colour-connected and has the alternating spanning cycle subgraph $1231 \cup 4564$, but does not contain a PC Hamilton cycle.

11.23. Using the definition of G_6 given after Theorem 11.1.42, construct, for every even n, a 3-edge-coloured complete graph on $n \geq 8$ vertices which is colour-connected and has a PC spanning cycle subgraph, but contains no PC Hamilton cycle.

11.24. Prove Proposition 11.1.43. Hint: consider the complete biorientation of a maximum spanning subgraph G of K_n^c such that no pair of edges in G is of the same colour. Apply Exercise 5.22 to see that $\overleftrightarrow{G}$ is hamiltonian.

11.25. (−) Prove that the alternating hamiltonian directed cycle problem is $\mathcal{NP}$-complete for bipartite 2-arc-coloured digraphs.

11.26. Prove Lemma 11.2.3.

11.27. (−) Prove Proposition 11.2.1.

11.28. Prove Lemma 11.2.7.

11.29. (+) Using the well-known inequality (see e.g. Feller's book [234, page 54])

$$\sqrt{2\pi}n^{n+1/2}e^{-n}e^{(12n+1)^{-1}} < n! < \sqrt{2\pi}n^{n+1/2}e^{-n}e^{(12n)^{-1}},$$

prove Lemma 11.2.8.

11.30. (+) Prove Lemma 11.2.9.

11.31. (−) Prove Proposition 11.2.6.

11.32. Let H be the 4-hypertournament with vertices $\{1,2,3,4,5\}$ and arcs

$$\{(2,3,4,5),(3,1,5,4),(2,1,4,5),(1,5,3,2),(1,2,3,4)\}.$$

Find a hamiltonian path in H.

11.33. Let H the 4-hypertournament defined above. Does H have a hamiltonian cycle?

11.34. Prove Theorem 11.2.12.

11.35. Prove that the hypertournament H_n introduced in the beginning of Subsection 11.3.3 is strong.

11.36. Prove directly that for every fixed $k \geq 4$ and n large enough every k-hypertournaments is traceable. Hint: use the fact that the majority digraph of T is semicomplete.

11.37. (+) Prove that every strong 3-hypertournament on $n \geq 5$ vertices is hamiltonian (Gutin and Yeo [374]).

11.38. Show that every hypertournament T has a 2-king, i.e. a vertex x such that for every $y \in V(T) - x$ there is an (x,y)-path of length at most two. Hint: see the hint for Exercise 11.36.

11.39. (−) Construct an alternating hamiltonian cycle in the 2-edge-coloured graph of Figure 11.12.

12. Additional Topics

The purpose of this chapter is to discuss briefly some topics that could not be covered in other chapters in the book and which we feel should still be mentioned. Depending on taste, several of these (and other topics which have been completely left out due to space limitations) could have taken up a whole chapter by themselves. Yet we think that our modest coverage will still show the flavour and potential usefulness of these topics. This applies in particular to the sections on matroids and heuristics for obtaining good solutions to $\mathcal{NP}$-hard problems.

12.1 Seymour's Second Neighbourhood Conjecture

Recall that for a vertex x in a digraph D, $N^{+2}(x)$ is the set of vertices of distance two from x. Seymour posed the following conjecture (see [187] and Problem 325, page 804 in volume 197/198 (1999) of *Discrete Mathematics*).

Conjecture 12.1.1 *Every oriented graph $D = (V, A)$ has a vertex x such that*

$$|N^+(x)| \leq |N^{+2}(x)|. \tag{12.1}$$

Note that, if we allow 2-cycles, then the conjecture is no longer true as can be seen by taking the complete digraph $\overset{\leftrightarrow}{K}_n$. Note also that, if the oriented graph has a vertex of out-degree zero, then this vertex satisfies the conjecture. This observation implies that it is sufficient to consider the conjecture for oriented graphs that are strongly connected.

The truth of Conjecture 12.1.1 in the case of tournaments was also conjectured by Dean [187]. This special case of the conjecture was proved by Fisher [237] using an analytic approach. Fisher's argument is non-trivial and involves the use of a probability distribution on the vertices along with Farkas' Lemma and several other tools. Moreover, Fisher's method does not explicitly identify a vertex which satisfies (12.1). Note that given any oriented graph D, such a vertex, or a proof that D is a counter-example to the conjecture, can be found in time $O(nm)$ (Exercise 12.1).

Below we give an elementary proof, due to Havet and Thomassé [407], of Conjecture 12.1.1 for the case of tournaments. The proof uses the concept of a median order of the vertex set of a tournament. A **median order** of a tournament T is an ordering $\mathcal{L} = v_1, v_2, \ldots, v_n$ of the vertices, such that the cardinality of the set of backwards arcs (namely arcs of the form v_iv_j, $i > j$) is minimum. In other words if H is an acyclic subdigraph of T whose size is maximum among all acyclic subdigraphs of T, then any acyclic ordering of H induces a median order on T.

By definition, if $\mathcal{L} = v_1, v_2, \ldots, v_n$ is a median order of $T = (V, A)$, then $A' = \{v_iv_j : i > j\}$ is a minimum feedback arc set in T (see Section 10.3). Hence, in the light of Conjecture 10.4.4, finding a median order of a tournament seems to be a difficult problem and the weighted version (where we seek an order which minimizes the total weight of the backwards arcs) is $\mathcal{NP}$-hard since it is easy to formulate the feedback arc set problem this way (Exercise 12.2).

The following relaxation of a median order, called a local median order in [407], is still a powerful tool as we shall see later. An ordering $\mathcal{L} = v_1, v_2, \ldots, v_n$ of the vertices of a tournament $T = (V, A)$ is a **local median order** if the following holds for all $1 \leq i \leq j \leq n$. (Here and below we use the notation $[v_i, v_j] = \{v_i, v_{i+1}, \ldots, v_j\}$ for all $1 \leq i \leq j \leq n$.)

$$|N^+(v_i) \cap [v_i, v_j]| \geq |N^-(v_i) \cap [v_i, v_j]| \text{ and} \tag{12.2}$$

$$|N^-(v_j) \cap [v_i, v_j]| \geq |N^+(v_j) \cap [v_i, v_j]|. \tag{12.3}$$

Note that, if (12.2) does not hold, then the number of forward arcs in

$$\mathcal{L}' = v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{j-1}, v_i, v_j, \ldots, v_n$$

is larger than in $\mathcal{L}$. Similarly if (12.3) does not hold, then we can obtain a better ordering (with respect to the number of forward arcs) by moving v_j just after v_i. Thus a local median order is precisely a local optimum, which cannot be improved by moving just one vertex in the ordering. Such an ordering can be found in polynomial time for any given digraph by using the 1-OPT procedure in Section 12.8 below.

The following is a direct consequence of the definition of a local median order:

Lemma 12.1.2 *Let $\mathcal{L} = v_1, v_2, \ldots, v_n$ be a local median order of a tournament T. Then for every $1 \leq i \leq j \leq n$ the ordering $\mathcal{L}_{ij} = v_i, v_{i+1}, \ldots, v_j$ is a local median order of $T\langle\{v_i, v_{i+1}, \ldots, v_j\}\rangle$.* $\square$

Lemma 12.1.2 provides us with a powerful inductive tool as we shall see below. Let T be a tournament and let $\mathcal{L} = v_1, v_2, \ldots, v_n$ be a local median order of T. We define a partition $G_{\mathcal{L}}, B_{\mathcal{L}}$ of $N^-(v_n)$ as follows:

$$G_{\mathcal{L}} = \{v_j : v_j \rightarrow v_n \text{ and there exists } i < j \text{ such that } v_n \rightarrow v_i \rightarrow v_j\};$$

$$B_{\mathcal{L}} = N^-(v_n) - G_{\mathcal{L}}.$$

The vertices of $G_{\mathcal{L}}$ are called **good** and those in $B_{\mathcal{L}}$ **bad** vertices. Note that $|N^{+2}(v_n)| \geq |G_{\mathcal{L}}|$. The following result by Havet and Thomassé implies that Conjecture 12.1.1 holds for tournaments.

Theorem 12.1.3 *[407] Let T be a tournament and let $\mathcal{L} = v_1, v_2, \ldots, v_n$ be a local median order of T. Then the vertex v_n has $|N^{+2}(v_n)| \geq |N^+(v_n)|$.*

Proof: Let $\mathcal{L} = v_1, v_2, \ldots, v_n$ be a local median order of T. We prove by induction on n that

$$|N^+(v_n)| \leq |G_{\mathcal{L}}|. \tag{12.4}$$

If $n = 1$ the claim is trivially true so suppose that $n > 1$. If $B_{\mathcal{L}} = \emptyset$ then we have $|G_{\mathcal{L}}| = |N^-(v_n)| \geq |N^+(v_n)|$, where the equality holds by the definition of the good vertices and the inequality holds by the definition of a local median order. Hence we may assume that there is a bad vertex. Choose i as small as possible so that v_i is bad. Define the sets $G^l_{\mathcal{L}}, G^h_{\mathcal{L}}, N^l, N^h$ as follows:

$$G^l_{\mathcal{L}} = G_{\mathcal{L}} \cap [v_1, v_i] \text{ and } G^h_{\mathcal{L}} = G_{\mathcal{L}} \cap [v_{i+1}, v_n],$$
$$N^l = N^+(v_n) \cap [v_1, v_i] \text{ and } N^h = N^+(v_n) \cap [v_{i+1}, v_n].$$

Note that, if a vertex is good with respect to the pair $(T\langle\{v_{i+1}, \ldots, v_n\}\rangle, \mathcal{L}^h)$, where $\mathcal{L}^h = v_{i+1}, \ldots, v_n$, then it is also good with respect to $(T, \mathcal{L})$. Hence, by the induction hypothesis (applied to $T\langle\{v_{i+1}, \ldots, v_n\}\rangle$ and the ordering $\mathcal{L}^h$), we have $|N^h| \leq |G^h_{\mathcal{L}}|$. The minimality of i implies that every vertex in $\{v_1, \ldots, v_{i-1}\}$ is either in $G^l_{\mathcal{L}}$ or N^l. Furthermore, since v_i is bad we have $N^l \subseteq N^+(v_i) \cap [v_1, v_{i-1}]$ and $N^-(v_i) \cap [v_1, v_{i-1}] \subseteq G^l_{\mathcal{L}}$. Now using (12.3) we obtain

$$|G^l_{\mathcal{L}}| \geq |N^-(v_i) \cap [v_1, v_{i-1}]| \geq |N^+(v_i) \cap [v_1, v_{i-1}]| \geq |N^l|.$$

Thus we have

$$|G_{\mathcal{L}}| = |G^l_{\mathcal{L}}| + |G^h_{\mathcal{L}}| \geq |N^l| + |N^h| = |N^+(v_n)|,$$

implying that (12.4) holds for all positive integers n. □

If a tournament has a vertex of out-degree zero, then this vertex satisfies (12.1) and the transitive tournament on n vertices shows that this vertex may be the only vertex satisfying (12.1). Using median orders Havet and Thomassé [407] proved that unless there is a vertex of out-degree zero, a tournament has at least two vertices satisfying (12.1).

Havet and Thomassé showed by an example that their method (just as Fisher's method [237]) will not suffice to prove Conjecture 12.1.1 in full. However, as an illustration of the power of median orders as a tool for proofs of results (on tournaments), Havet and Thomassé proved the following result. Recall that an oriented tree is an orientation of an undirected tree.

Theorem 12.1.4 *[407] Every tournament of order at least $\frac{7n-5}{2}$ contains every oriented tree on n vertices as a subdigraph.* □

This is a significant step towards proving the following conjecture due to Sumner (see [740]). Previous results on the conjecture (including a proof that every tournament on at least $(4+o(1))n$ vertices contains every oriented tree on n vertices) were obtained by Häggkvist and Thomason [390].

Conjecture 12.1.5 (Sumner) *Every tournament on at least $2n-2$ vertices contains every oriented tree on n vertices.*

12.2 Ordering the Vertices of a Digraph of Paired Comparisons

In this section we consider several methods for ordering the vertices of a weighted digraph. Even though all the methods we study can be applied to arbitrary weighted digraphs, we concentrate on so-called paired comparison digraphs (PCDs), a graph-theoretical model for the method of paired comparisons [183], which are defined in Subsection 12.2.1. In that subsection, we consider also the score method and the feedback set ordering method to order the vertices of PCDs. Limitations of these two methods imply the necessity to introduce and study other methods of ordering. In the main part of this section we consider three methods of ordering that are due to Kano and Sakamoto, and were introduced in 1983. These methods are described in Subsection 12.2.2; several results on these methods are given in the following subsections.

12.2.1 Paired Comparison Digraphs

The method of paired comparisons is an approach to ordering a group of objects. In the framework of this method, objects are considered in pairs, a pair at a time, and the decision is made of which of the two is better. This procedure is repeated with all or some other pairs. This method is normally applied when objects are characterized by many parameters and/or some parameters are unknown or vague (of non-numerical nature). The method of paired comparisons is usually carried out by a team of experts. In general, the experts will have different views and thus an object M will be favoured over an object N by some experts, while others will prefer N over M. (Notice that in general some pairs will not be compared at all.) Hence, the results of the use of the method of paired comparisons often have to be analyzed to find an 'average' ordering.

To carry out such an analysis, a paired comparison digraph D is initially constructed. The vertices of D correspond to the objects and, for an ordered pair x, y of vertices (i.e. objects) in D, the arc xy is in D if and only if some experts prefer y to x. The weight of xy is the fraction of the experts that favour x over y. Formally, following Kano and Sakamoto [472, 473], we introduce a digraph of paired comparisons as follows. Let $D = (V, A, \epsilon)$ be a weighted digraph in which every arc xy has a positive real weight $\epsilon(xy)$. A digraph D is called a **paired comparison digraph** (abbreviated to **PCD**) if D satisfies the following conditions:

(a) $0 < \epsilon(xy) \leq 1$ for every $xy \in A$;
(b) $\epsilon(xy) + \epsilon(yx) = 1$ if both xy and yx are arcs;
(c) $\epsilon(xy) = 1$ if $xy \in A$ but $yx \notin A$.

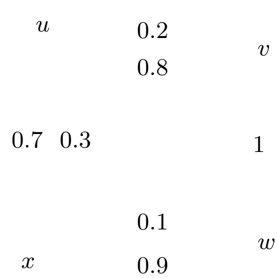


Figure 12.1 A paired comparison digraph H.

See Figure 12.1 for an example of a paired comparison digraph. An (unweighted) digraph $D = (V, A)$ can be viewed as a PCD by setting the weight of each arc of D as follows:

(i) $\epsilon(xy) = \epsilon(yx) = 0.5$ if $xy, yx \in A$;
(ii) $\epsilon(xy) = 1$ if $xy \in A$ but $yx \notin A$.

We call the PCD $D' = (V, A, \epsilon)$ with the weight function ϵ determined by (i) and (ii) the **uniform PCD corresponding to D**. The **positive (negative)** **score** of a vertex $x \in V$ is

$$\sigma^+(x) = \sum_{xy \in A} \epsilon(xy), \quad (\sigma^-(x) = \sum_{yx \in A} \epsilon(yx).)$$

In Figure 12.1, $\sigma^+(u) = 0.5$ and $\sigma^-(u) = 1.5$.

A PCD D is not always semicomplete (some pairs of vertices may not be compared). If D is a tournament, then usually the vertices of D are ordered according to their positive score, with the first vertex being of highest positive score. This approach, the **score method**, is justified by a series of natural axioms (see the paper [647] by Rubinstein). The score method can be naturally used for semicomplete PCDs. When a PCD is not semicomplete, the score method may produce results that are not justified from the practical point of view. For example, consider the digraph $R = (V, A)$ with $V = \{1, 2, \ldots, n\}$, $n \geq 5$, and $A = \{12, 13\} \cup \{41, 51, \ldots, n1\}$. Let $R' = (V, A, \epsilon)$ be the uniform PCD corresponding to R. Even though the positive score of the vertex 1 is maximum, it is against our intuition to order in R' the vertex 1 first (i.e. the winner). This raises the question of finding a method of ordering the vertices of an arbitrary PCD, which agrees with the score method for semicomplete PCDs.

In Subsection 10.3.3, we studied a method of ordering of the vertices of a weighted digraph $D = (V, A, \epsilon)$, the feedback set ordering (FSO). Recall that, for an ordering $\alpha = (v_1, v_2, \ldots, v_n)$, where $n = |V|$ and $\{v_1, v_2, \ldots, v_n\} = V$, an arc $(v_i, v_j) \in A$ is forward (backward) if $i < j$ $(i > j)$. In Figure 12.1, for the ordering $\beta = (u, v, w, x)$, uv, ux and wx are (all) forward arcs; vu and xw are backward arcs. An ordering $\alpha = (v_1, v_2, \ldots, v_n)$ can be viewed as a bijection from V to $\{1, 2, \ldots, n\}$. Thus, for a vertex $x \in V$, $\alpha(x) = i$ if $x = v_i$. An ordering α of V is **FSO-optimal** if the number of backward arcs is minimum. Let $OR(D)$ denote the set of all FSO-optimal orderings of V. In many cases, $OR(D)$ has more than one element. In these cases, the final objective is to calculate the **proper FSO-rank** of every vertex x of D, i.e.

$$\pi_{FSO}(x) = \frac{1}{|OR(D)|} \sum_{\alpha \in OR(D)} \alpha(x). \tag{12.5}$$

The final ordering is carried out according to the proper FSO-ranks; the best vertex has the smallest FSO-rank.

Although the FSO method is of definite importance for some applications, it does not agree with the score method for semicomplete digraphs: Let T be the digraph with vertices $1, 2, 3, 4, 5$, in which there is a pair of opposite arcs between any pair of distinct vertices except for the pairs $\{i, i+1\}$, $i = 1, 2, 3, 4$, and $\{1, 3\}$. Moreover, $i \rightarrow i+1$ for $i = 1, 2, 3, 4$ and $3 \rightarrow 1$. Let T' be the uniform PCD corresponding to the semicomplete digraph T. According to the score method, the set of optimal orderings is $\{(3, i, j, k, 5) : \{i, j, k\} = \{1, 2, 4\}\}$. This implies that the proper ranks of the vertices according to the score method (by an obvious analogue of (12.5)) are $\pi_S(3) = 1$, $\pi_S(1) = \pi_S(2) = \pi_S(4) = 3$ and $\pi_S(5) = 5$. At the same time, the orderings that are optimal according to FSO form the set $OR(T') = \{(1, 2, 3, 4, 5), (2, 3, 1, 4, 5), (3, 1, 2, 4, 5), (2, 3, 4, 5, 1), (3, 1, 4, 5, 2), (3, 4, 5, 1, 2)\}$. (To see this, first observe that the contribution from 2-cycles is independent on the ordering and hence can be ignored. Secondly, observe that in an FSO-optimal ordering, which actually has only one backward ordinary arc[1], 3 must be before 4 and 5, 4 before 5, and the vertices 1,2,3 must appear in this order, or as either 2,3,1 or 3,1,2.) By (12.5), we obtain that

$$\pi_{FSO}(3) < \pi_{FSO}(1) = \pi_{FSO}(2) < \pi_{FSO}(4) < \pi_{FSO}(5).$$

[1] Recall that an arc xy is ordinary if the opposite arc yx does not exist.

We leave it to the reader to construct other examples of semicomplete PCDs, for which FSO and the score method produce different results (Exercise 12.4).

12.2.2 The Kano-Sakamoto Methods of Ordering

In this subsection we describe three methods (forward, backward and mutual) of ordering introduced by Kano and Sakamoto [472, 473]. Notice that, for semicomplete digraphs, all these methods agree with the score method. In Subsection 12.2.3, we prove this important result. In Subsection 12.2.4, we characterize orderings that are optimal with respect to the mutual method. In Subsection 12.2.5, we study the complexity of the problems to find forward and backward optimal orderings as well as some ways to obtain polynomial algorithms for these problems restricted to semicomplete multipartite PCDs and PCDs close to them.

Although the reader may find examples of PCDs for which the methods of Kano and Sakamoto, especially the mutual one, produce counter-intuitive orderings, these methods seem to give adequate results for PCDs close to semicomplete, which are perhaps of the main interest for the method of paired comparisons.

Let $D = (V, A, \epsilon)$ be a PCD. Let x and y be a pair of distinct vertices in D and let α be an ordering of D. Then α_{xy} denotes an ordering of D as follows: $\alpha_{xy}(z) = \alpha(z)$ for every $z \notin \{x, y\}$, and $\alpha_{xy}(x) = \alpha(y)$, $\alpha_{xy}(y) = \alpha(x)$. The **length** of an arc $vu \in A$ is $\epsilon(vu)|\alpha(v) - \alpha(u)|$. The **forward (backward) length** $f_D(\alpha)$ ($b_D(\alpha)$) of α is the sum of the lengths of all forward (backward) arcs. The **mutual length** of α is $m_D(\alpha) = f_D(\alpha) - b_D(\alpha)$. In Figure 12.1, the ordering $\beta = (u, v, w, x)$ has forward length $f_H(\beta) = (0.2{\cdot}1 + 0.3{\cdot}3 + 0.9{\cdot}1) = 2$, backward length $b_H(\beta) = (0.8\cdot 1 + 1\cdot 1 + 0.1\cdot 1 + 0.7\cdot 3) = 4$, and mutual length $m_H(\beta) = -2$. Clearly, (x, w, v, u) is a better ordering (with respect to all three criteria) than β. Even $f_H(\beta_{uv}) = 2.3 > f_H(\beta)$.

An ordering α is **forward (backward, mutual) optimal** if the corresponding parameter $f_D(\alpha)$ ($b_D(\alpha)$, $m_D(\alpha)$) is maximum (minimum, maximum) over all orderings of D. The set of all forward (backward, mutual) optimal orderings of D is denoted by $FOR(D)$ ($BOR(D)$, $MOR(D)$). The final objective is to calculate the **proper forward rank** (**proper backward rank, proper mutual rank**) of every vertex x of D. They are obtained by replacing $OR(D)$ with $FOR(D)$ ($BOR(D)$,$MOR(D)$) in (12.5). Clearly, the best vertex of D has the lowest proper rank in each case. In Figure 12.1, $BOR(H) = \{(w, v, x, u), (w, x, v, u)\}$ (we will see how to find $BOR(D)$ for a semicomplete multipartite PCD in Subsection 12.2.5). Thus, $\pi_B(w) = 1, \pi_B(x) = \pi_B(v) = 2.5$ and $\pi_B(u) = 4$.

12.2.3 Orderings for Semicomplete PCDs

Lemma 12.2.1 *[472, 473] Let $K = (V, A, \epsilon)$ be a semicomplete PCD with n vertices, and let α be an ordering of V. Then*

$$f_K(\alpha) = \frac{1}{3}n(n^2-1) - \sum_{x \in V} \sigma^+(x)\alpha(x);$$
$$b_K(\alpha) = \sum_{x \in V} \sigma^+(x)\alpha(x) - \frac{1}{6}n(n^2-1).$$

Proof: The equality for $f_K(\alpha)$ can be proved by induction on n (Exercise 12.5). The equality for $b_K(\alpha)$ can be easily obtained from that for $f_K(\alpha)$ by using the fact that $f_K(\alpha) + b_K(\alpha) = n(n^2-1)/6$, the proof of which is left as Exercise 12.6. □

This lemma implies the following:

Theorem 12.2.2 *[472, 473] Let $K = (V, A, \epsilon)$ be a semicomplete PCD with n vertices, and let α be an ordering of V. Then $\alpha = (v_1, v_2, \ldots, v_n)$ is forward (backward) optimal if and only if $\sigma^+(v_i) \geq \sigma^+(v_{i+1})$ for every $i = 1, 2, \ldots, n-1$.*

Proof: Let $\alpha = (v_1, v_2, \ldots, v_n)$ be a forward (backward) optimal ordering. Suppose that $\sigma^+(v_i) < \sigma^+(v_{i+1})$ for some i. By Lemma 12.2.1,

$$f_K(\alpha) - f_K(\alpha_{v_i v_{i+1}}) = \sigma^+(v_i) - \sigma^+(v_{i+1}) < 0.$$

Hence, α is not forward optimal, a contradiction. Analogously, we can show that α is not backward optimal, a contradiction. So, we may conclude that $\sigma^+(v_i) \geq \sigma^+(v_{i+1})$ for every $i = 1, 2, \ldots, n-1$. On the other hand, let $\beta = (w_1, w_2, \ldots, w_n)$ be an ordering such that $\sigma^+(w_i) \geq \sigma^+(w_{i+1})$ for every $i = 1, 2, \ldots, n-1$. By the formula for $f_K(\alpha)$ in Lemma 12.2.1, $f_K(\alpha) = f_K(\beta)$. Hence, $\beta \in FOR(K)$. Analogously, we see that $\beta \in BOR(K)$. □

It is easy to see that this theorem allows one to compute the proper forward and backward ranks of a semicomplete PCD in polynomial time. Clearly, Theorem 12.2.2 is also valid for the mutual orderings of the semicomplete PCDs. However, for the mutual orderings, a more general assertion is true. We prove it in the next subsection.

For a vertex x of a PCD, let $\sigma^*(x) = \sigma^+(x) - \sigma^-(x)$. Since for a semicomplete PCD D of order n, $\sigma^+(x) + \sigma^-(x) = n-1$, we have that $\sigma^+(x) \geq \sigma^+(y)$ if and only if $\sigma^*(x) \geq \sigma^*(y)$. Therefore, Theorem 12.2.2 can be reformulated using σ^* instead of σ^+.

12.2.4 The Mutual Orderings

Kano and Sakamoto proved the following characterization of the mutual length of an ordering α:

Lemma 12.2.3 *[472] Let $D = (V, A, \epsilon)$ be a PCD and let α be an ordering of V. Then the mutual length of α satisfies*

$$m_D(\alpha) = -\sum_{x \in V} \sigma^*(x)\alpha(x).$$

Proof: Define

$$\bar{\epsilon}(u,v) = \begin{cases} \epsilon(u,v) & \text{if } (u,v) \text{ is an arc of } D \\ 0 & \text{otherwise (in particular, if } u = v). \end{cases}$$

Let $F(\alpha)$ ($B(\alpha)$) be the set of forward (backward) arcs for α. We have

$$\begin{aligned} m_D(\alpha) &= \sum_{(x,y)\in F(\alpha)} \epsilon(x,y)(\alpha(y)-\alpha(x)) - \sum_{(x,y)\in B(\alpha)} \epsilon(x,y)(\alpha(x)-\alpha(y)) \\ &= \sum_{(x,y)\in A} \epsilon(x,y)(\alpha(y)-\alpha(x)) \\ &= \sum_{y\in V}\left(\sum_{x\in V}\bar{\epsilon}(x,y)\alpha(y)\right) - \sum_{x\in V}\left(\sum_{y\in V}\bar{\epsilon}(x,y)\alpha(x)\right) \\ &= \sum_{y\in V}\sigma^-(y)\alpha(y) - \sum_{x\in V}\sigma^+(x)\alpha(x) = -\sum_{x\in V}\sigma^*(x)\alpha(x). \end{aligned}$$

□

Analogously to Theorem 12.2.2, but using the previous lemma instead of Lemma 12.2.1, we can prove the following:

Theorem 12.2.4 *[472] Let $D = (V, A, \epsilon)$ be a PCD with n vertices, and let α be an ordering of V. Then $\alpha = (v_1, v_2, \ldots, v_n)$ is mutual optimal if and only if $\sigma^*(v_i) \geq \sigma^*(v_{i+1})$ for every $i = 1, 2, \ldots, n-1$.* □

This theorem shows that the proper mutual ranks of vertices depend only on σ^*, not on the structure of a PCD. This indicates that perhaps mutual orderings are not sound for non-semicomplete PCDs.

12.2.5 Complexity and Algorithms for Forward and Backward Orderings

We saw in the previous subsection how to find a mutual optimal ordering (simply order according to the values of σ^*); this obviously can be done in polynomial time. The time complexity of the same problems for forward and backward optimal ordering are significantly more difficult (unless $\mathcal{P} = \mathcal{NP}$) as we see below.

Theorem 12.2.5 *[473] The problem of finding a backward optimal ordering of a PCD is $\mathcal{NP}$-hard.*

Proof: The following problem, called the optimal linear arrangement problem (OLAP), is $\mathcal{NP}$-completes see [303, page 200].

Instance: A graph $G = (V, E)$ and a positive integer k.

Question: Is there an ordering α of V so that

$$\sum_{\{x,y\}\in E} |\alpha(x) - \alpha(y)| \leq k.$$

Let $G = (V, E)$ be a graph and let $D = \overleftrightarrow{G}$ be the complete biorientation of G. Let also $\epsilon(xy) = 0.5$ for every $xy \in A(D)$. Then for every ordering α

$$\sum_{\{x,y\}\in E} |\alpha(x) - \alpha(y)| = 2f_D(\alpha).$$

Hence, the OLAP is polynomially reducible to the problem of finding a backward optimal ordering of the vertices of a PCD. □

A similar but slightly longer proof in [373] shows that the problem to find a forward optimal ordering of the vertices of a PCD is $\mathcal{NP}$-hard too (Exercise 12.7). This means that, in order to design polynomial algorithms to compute forward and backward optimal orderings, we need to restrict ourselves to special classes of PCDs. Since the method of paired comparisons is of main interest when a PCD is quite dense, it is useful to consider PCDs close to semicomplete. For semicomplete multipartite PCDs, characterizations of forward backward (forward) orderings were obtained by Kano [471] (Gutin and Yeo [373], respectively). In this subsection, we describe only the main result of [471]; the characterization in [373] is more complicated. Using the above-mentioned characterizations, Gutin [360] and Gutin and Yeo [373] constructed polynomial algorithms to find proper backward ranks and proper forward ranks, respectively, of the vertices of semicomplete multipartite PCDs. We will also discuss the method of multipartite completion (see [471, 473]), which allows one to find effectively all forward and backward optimal orderings in PCDs close to semicomplete multipartite PCDs.

Let $D = (V, A, \epsilon)$ be a semicomplete multipartite PCD and let α be an ordering of V. Then, for a vertex $x \in V$, we define $\psi(\alpha, x) = \sigma^+(x) + |\{y \in U : \alpha(y) > \alpha(x)\}|$, where U is the partite class of D containing x. The following lemma is proved in [471]; we give a much shorter proof adopted from [373].

Lemma 12.2.6 *Let α be an ordering of the vertices of a semicomplete multipartite PCD $D = (V, A, \epsilon)$, $n = |V|$. Then*

$$b_D(\alpha) = \sum_{x\in V} \psi(\alpha, x)\alpha(x) - \frac{1}{6}n(n^2 - 1).$$

Proof: For every partite set U of D, add the set of arcs $\{vw : v, w \in U, \alpha(w) > \alpha(v)\}$ (all of weight one) to A. The new PCD H is semicomplete. Observe that the positive score of a vertex x in H equals $\psi(\alpha, x)$. Now the formula of this lemma follows from the equality for $b_K(\alpha)$ in Lemma 12.2.1

and the fact that $b_D(\alpha) = b_H(\alpha)$ (which holds since all new arcs in H are forward). □

This lemma implies the following result (Exercise 12.9):

Lemma 12.2.7 *[471] Suppose that β is an ordering of the vertices of a semicomplete multipartite PCD D, and X and Y are distinct partite sets of D. We have the following:*

(a) If $x, y \in X$ and $m = \beta(y) - \beta(x) > 0$, then

$$b_D(\beta_{xy}) - b_D(\beta) = m(\sigma^+(x) - \sigma^+(y)).$$

(b) If $x \in X, y \in Y$, $m = \beta(y) - \beta(x) > 0$ and there is no vertex $z \in X \cup Y$ such that $\beta(x) < \beta(z) < \beta(y)$, then

$$b_D(\beta_{xy}) - b_D(\beta) = m(\psi(\beta, x) - \psi(\beta, y)).$$

□

Using this lemma one can prove the following (the actual proof is left as Exercise 12.10):

Theorem 12.2.8 *[471] Let D be a semicomplete multipartite PCD of order n. An ordering $\alpha = (v_1, v_2, \ldots, v_n)$ is backward optimal if and only if the following two conditions hold.*

(a) $\psi(\alpha, v_1) \geq \psi(\alpha, v_2) \geq \ldots \geq \psi(\alpha, v_n)$.
(b) For every pair x, y of vertices in the same partite set of D, $\alpha(x) < \alpha(y)$ implies $\sigma^+(x) \geq \sigma^+(y)$.

□

We illustrate this theorem by the semicomplete bipartite PCD H in Figure 12.1. We have $\sigma^+(u) = 0.5$, $\sigma^+(v) = \sigma^+(x) = 0.8$, $\sigma^+(w) = 1.9$. Let α be backward optimal. Then, by (b), $\alpha(w)$ is less than $\alpha(u)$ implying that $\psi(\alpha, w) = 2.9, \psi(\alpha, u) = 0.5$. Since the positive scores of v and x coincide, there are two backward optimal orderings α', α'' and $\psi(\alpha', v) = \psi(\alpha'', x) = 0.8$, $\psi(\alpha'', v) = \psi(\alpha', x) = 1.8$. By (a), $BOR(H) = \{(w, x, v, u), (w, v, x, u)\}$. Hence, $\pi_B(w) = 1$, $\pi_B(x) = \pi_B(v) = 2.5$ and $\pi_B(u) = 4$. Another example to illustrate this theorem is given in Exercise 12.11.

Applying Theorem 12.2.8, it is not difficult to construct a polynomial algorithm to find proper backward ranks of the vertices of a semicomplete multipartite PCD [360] (Exercise 12.12).

Let $\ell(D)$ be the backward length of a backward optimal ordering of a digraph D. Let D be a non-semicomplete multipartite PCD with partite sets $V_1, V_2, \ldots, V_k$. The semicomplete multipartite PCD obtained from D by adding exactly one arc between every pair of non-adjacent vertices from distinct partite sets is called a **multipartite completion** of D. Let $\mathcal{C}(D)$ be the set of multipartite completions of D. The significance of this set is given in the following theorem:

Theorem 12.2.9 *[473] Let D be a non-semicomplete multipartite PCD. Then*

$$\ell(D) = \min\{\ell(H) : \ H \in \mathcal{C}(D)\}.$$

Moreover, if $\mathcal{H} = \{H \in \mathcal{C}(D) : \ \ell(H) = \ell(D)\}$, then

$$BOR(D) = \cup_{H \in \mathcal{H}} BOR(H).$$

Proof: Exercise 12.13. □

Clearly, if the cardinality of $\mathcal{C}(D)$ is not large, this theorem allows one to list backward optimal orderings of D.

12.3 (k, l)-Kernels

Galeana-Sánchez and Li [293] introduced the concept of a (k, l)-kernel in a digraph. This concept generalizes several well-known notions of special independent sets of vertices such as a kernel and a quasi-kernel. In this section, we discuss (k, l)-kernels and their special important cases, kernels and quasi-kernels, and study some basic properties of kernels and quasi-kernels. The notion of a (k, l)-kernel has various applications, especially that of a $(2, 1)$-kernel.

Let k and l be integers with $k \geq 2$, $l \geq 1$, and let $D = (V, A)$ be a digraph. A set $J \subseteq V$ is a **(k, l)-kernel** of D if

(a) for every ordered pair x, y of distinct vertices in J we have $\mathrm{dist}(x, y) \geq k$,
(b) for each $z \in V - J$, there exists $x \in J$ such that $\mathrm{dist}(z, x) \leq l$.

A **kernel** is a $(2, 1)$-kernel and a **quasi-kernel** is a $(2, 2)$-kernel. Galeana-Sánchez and Li [293] proved some results which relate (k, l)-kernels in a digraph D to those in its line digraph. In particular, they proved the following:

Theorem 12.3.1 *Let D be a digraph with $\delta^-(D) \geq 1$. Then the number of $(k, 1)$-kernels in $L(D)$ is less than or equal to the number of $(k, 1)$-kernels in D.* □

12.3.1 Kernels

We start with an equivalent definition of a kernel. A set K of vertices in a digraph $D = (V, A)$ is a kernel if K is independent and the first closed neighbourhood of K, $N^-[K]$, is equal to V. This notion was introduced by von Neumann in [731]; kernels have found many applications, for instance in game theory (a kernel represents a set of winning positions, cf. [731] and Chapter 14 in the book by Berge [108]), in logic [109] and in list edge-colouring of graphs (see Section 12.4). Chvátal (see [303], p. 204) proved that the problem to verify whether a given digraph has a kernel is $\mathcal{NP}$-complete. Several sufficient conditions for the existence of a kernel have been proved. Many of these conditions can be trivially extended to **kernel-perfect** digraphs, i.e. digraphs for which every induced subdigraph has a kernel. The notion of kernel-perfect digraphs allows one to simplify certain proofs (due to the possibility of using induction, see the proof of Theorem 12.3.2) and is quite useful for applications (see Section 12.4).

Clearly, every symmetric digraph, i.e. digraph whose every arc belongs to a 2-cycle, is kernel-perfect (every maximal independent set is a kernel). It was proved by von Neumann and Morgenstern [731] that every acyclic digraph is kernel-perfect. Richardson [635] generalized this result as follows:

Theorem 12.3.2 *Every digraph with no odd cycle is kernel-perfect.*

The proof of Theorem 12.3.2, which we present here, is an adaptation of the one by Berge and Duchet [110]. A digraph which is not kernel-perfect is called **kernel-imperfect**. We say that a digraph D is **critical kernel-imperfect** if D is kernel-imperfect, but every proper induced subdigraph of D is kernel-perfect.

Lemma 12.3.3 *Every critical kernel-imperfect digraph is strong.*

Proof: Assume the converse and let $D = (V, A)$ be a non-strong critical kernel-imperfect digraph. Let T be a terminal strong component of D and let S_1 be a kernel of T. Since D has no kernel, the set $M = V - N^-[S_1]$ is non-empty. Hence the fact that D is critical kernel-imperfect implies that $D\langle M\rangle$ has a kernel S_2. The set $S_1 \cup S_2$ is independent since no arc goes from S_1 to S_2 (by the definition of a terminal strong component) and no arc goes from S_2 to S_1 (by the definition of M). Clearly, $N^-[S_1 \cup S_2] = V$. Hence, $S_1 \cup S_2$ is a kernel of D, a contradiction. □

Proof of Theorem 12.3.2: Let D be a kernel-imperfect digraph with no odd cycle and let D' be a critical kernel-imperfect subdigraph of D. By the lemma above, D' is strong. Since D' is strong and has no odd cycles, by Theorem 1.8.1, D' is bipartite. Let K be a partite set in D'. Since D' is strong, K is a kernel of D', a contradiction. □

This theorem has been strengthened in a number of papers. The conditions (a) and (b) of the following theorem are due to Duchet (see the papers by Berge [110]), and Galeana-Sánchez and Neumann-Lara [294], respectively). Galeana-Sánchez showed that for every $k \geq 2$, there are non-kernel-perfect digraphs for which every odd cycle has at least k chords [291].

Theorem 12.3.4 *A digraph D is kernel-perfect if at least one of the following conditions holds:*

(a) Every odd cycle has two arcs belonging to 2-cycles;
(b) Every odd cycle has two chords whose heads are consecutive vertices of the cycle.

□

There were other attempts to strengthen Richardson's Theorem 12.3.2. In particular, Duchet (see [132]) conjectured that every digraph D, which is not an odd cycle and which does not have a kernel, contains an arc e such that $D - e$ has no kernel either. Aparsin, Ferapontova and Gurvich [26] found a counterexample to this conjecture which we describe below. For an integer $n \geq 2$ and a set $W \subseteq \{1, 2, \ldots, n-1\}$, a **circular digraph** $C_n(W)$ is defined as follows: $V(C_n(W)) = \{1, 2, \ldots, n\}$ and

$$A(C_n(W)) = \{(i, i+j \pmod{n}) : \ 1 \leq i \leq n, j \in W\}.$$

In particular, $C_n(\{1, 2, \ldots, n-1\}) = \overleftrightarrow{K}_n$ and $C_n(\{1\}) = \vec{C}_n$.

Aparsin, Ferapontova and Gurvich proved that the circular digraph $C_{43}(\{1, 7, 8\})$ has no kernel, but after deletion of any arc in this digraph a kernel will appear. Observe that by the symmetry of $C_{43}(\{1, 7, 8\})$ one needs only to show that $C_{43}(\{1, 7, 8\}) - (1, 2)$, $C_{43}(\{1, 7, 8\}) - (1, 8)$ and $C_{43}(\{1, 7, 8\}) - (1, 9)$ have kernels. This task is left as Exercise 12.16. We note that $C_{43}(\{1, 7, 8\})$ is the only known counterexample to the Duchet conjecture; Gurvich (private communication, December 1999) suspects that there is an infinite such family of circular digraphs. It was also proved in [26] that $C_n(\{1, 7, 8\})$ has a kernel if and only if $n \equiv 0 \pmod{3}$ or $n \equiv 0 \pmod{29}$. The following problem seems quite natural:

Problem 12.3.5 *Characterize circular digraphs with kernels.*

A biorientation D of a graph G is called **normal**, if every subdigraph of D which is a semicomplete digraph has a kernel. An undirected graph G is **kernel-solvable** if every normal biorientation of G has a kernel. Boros and Gurvich [132] showed that a slight modification of the above conjecture of Duchet holds. They proved the following:

Theorem 12.3.6 *Let G be a connected non-kernel-solvable graph, which is not an odd cycle of length at least 5. Then there exists an edge e in G such that $G - e$ is not kernel-solvable either.* □

Berge and Duchet (see [543]) conjectured that a graph G is perfect[2] if and only if G is kernel-solvable. Boros and Gurvich [131] proved one direction of this conjecture, namely:

Theorem 12.3.7 *Every perfect graph is kernel-solvable.* □

[2] A graph G is perfect if, for every induced subgraph H of G, the chromatic number of H is equal to the order of the largest clique of H.

The two original proofs of Theorem 12.3.7 are quite involved and lengthy. Using the notion of a fractional kernel, Aharoni and Holzman [3] found a much shorter proof of Theorem 12.3.7. Many special cases of the above conjecture had been proved before, see [543] and references therein. In particular, Maffray [543] proved the following result:

Theorem 12.3.8 *A biorientation of a line graph is kernel-perfect if and only if it is normal.*

This result was extended to line multigraphs by Borodin, Kostochka and Woodall [130].

12.3.2 Quasi-Kernels

We start with an equivalent definition of a quasi-kernel. A set Q of vertices in a digraph $D = (V, A)$ is a quasi-kernel if Q is independent and the second closed in-neighbourhood of Q, $N^{-2}[Q]$, is equal to V. The two results on 2-kings (or, more precisely, 2-serfs) in tournaments mentioned in the beginning of Section 12.3.2 have been extended to quasi-kernels in arbitrary digraphs as follows. The first theorem is by Chvátal and Lovász [162] (see also [524]). It has a surprisingly short proof.

Theorem 12.3.9 *Every digraph D has a quasi-kernel.*

Proof: The proof is by induction on the order of D. The base case when the number of vertices is 1 is trivial. Let D be a digraph of order n and assume (as the induction hypothesis) that all digraphs with less than n vertices have a quasi-kernel. If D has a kernel, we are done. Assume D has no kernel. Let x be a vertex in D. Consider $D' = D - (x \cup N^-(x))$. By induction, D' has a quasi-kernel Q'. If $Q' \cup x$ is an independent set, then, clearly, this set is a quasi-kernel in D.

Suppose now that $Q' \cup x$ is not independent. Then there exists a vertex $z \in Q'$ which is adjacent to x. As $z \notin N^-(x)$, $x \rightarrow z$. Now it follows that Q' is a quasi-kernel in the whole digraph D. □

The second theorem is by Jacob and Meyniel [454].

Theorem 12.3.10 *If a digraph $D = (V, A)$ has no kernel, then D contains at least three quasi-kernels.*

Proof: By Theorem 12.3.9, D has a quasi-kernel Q_1. Since D has no kernel, we have $V \neq N^-[Q_1]$. Let Q_2 be a quasi-kernel of $D - N^-[Q_1]$. We will prove that $Q_2' = Q_2 \cup (Q_1 - N^-(Q_2))$ is a quasi-kernel of D. It is straightforward to see that Q_2' is independent and

$$V = (V - N^-[Q_1]) \cup N^-[Q_1 \cap N^-(Q_2)] \cup N^-[Q_1 - N^-(Q_2)].$$

By the definition of Q_2, every vertex of $V - N^-[Q_1]$ is the initial vertex of a path of length at most two terminating in Q_2. Since $N^-[Q_1 \cap N^-(Q_2)] \subseteq N^{-2}[Q_2]$, every vertex of $N^-[Q_1 \cap N^-(Q_2)]$ is the initial vertex of a path of length at most two terminating in Q_2. Since $N^-[Q_1 - N^-(Q_2)] \subseteq N^-[Q_1]$, a vertex of $N^-[Q_1 - N^-(Q_2)]$ either belongs to Q_1 or is the tail of an arc whose head is in $Q_1 - N^-(Q_2)$. Hence, Q_2 is a quasi-kernel.

Observe that $Q_1 \cap Q_2 = \emptyset$ and $Q_2 \neq \emptyset$. Thus, $Q'_2 \neq Q_1$.

As Q'_2 is not a kernel of D, we have $V \neq N^-[Q'_2]$. Let Q_3 be a quasi-kernel of $D - N^-[Q'_2]$ and let $Q'_3 = Q_3 \cup (Q'_2 - N^-(Q_3))$. As above, we can demonstrate that Q'_3 is a quasi-kernel distinct from Q'_2. It remains to show that $Q'_3 \neq Q_1$. Observe that $Q_3 \subseteq V - N^-[Q'_2]$ and $Q_1 \subseteq N^-[Q'_2]$. Thus, $Q_1 \cap Q_3 = \emptyset$. By this fact and since Q_3 is nonempty, we conclude that $Q'_3 \neq Q_1$. □

12.4 List Edge-Colourings of Complete Bipartite Graphs

The topic of this section may seem to have nothing to do with directed graphs, but as we will see, directed graphs have been a useful tool for solving the so-called Dinitz problem which we now describe. Our discussion in this section is inspired by the book [8] by Aigner and Ziegler and Galvin's paper [302].

An $n \times n$ matrix M over the integers $\{1, 2, \ldots, n\}$ is a **Latin square** (of size n) if no two entries in the same row and no two entries in the same column are equal. It is an easy exercise to show that for every integer $n \geq 1$ there exists a Latin square (Exercise 12.17).

A **proper edge-colouring** of an undirected graph $G = (V, E)$ is an assignment of integers to the edges in such a way that no two edges with a common end-vertex receive the same colour. The smallest k such that a graph G has a proper edge-colouring using only colours from the set $\{1, 2, \ldots, k\}$ is called the **chromatic index** of G. Thus it is easy to see that there is a 1-1 correspondence between the set of Latin squares of size n and the set of proper edge-colourings of the complete bipartite graph $K_{n,n}$ using colours $\{1, 2, \ldots, n\}$.

Proper edge-colourings are useful for various practical applications such as time table construction, see e.g. the book by Jensen and Toft [459]. In rest of this section we omit the word 'proper' since only proper edge-colourings will be considered.

In 1979 Dinitz raised the following problem (see e.g. [221, 222]): suppose we are given an $n \times n$ matrix whose (i, j) entry is a set $C(i, j)$ of n integers, $1 \leq i, j \leq n$, is it always possible to choose from each set $C(i, j)$ one element c_{ij} in such a way that the elements in each row are distinct and the elements in each column are distinct?

The Dinitz problem can be reformulated in terms of edge-colourings of complete bipartite graphs. Suppose that we are given, for each edge ij of the complete bipartite graph $K_{n,n}$, a set $C(i, j)$ of possible colours for that edge.

Does there always exist an edge-colouring of $K_{n,n}$ so that for each edge ij the colour c_{ij} of ij belongs to $C(i,j)$? In this formulation the Dinitz problem is just a special case of the more general **list colouring conjecture** (see e.g. the book by Jensen and Toft [459]) which states that, if a graph G has an edge-colouring with k colours, then no matter how we assign to each edge e of G a set C_e of k arbitrary colours, G has an edge colouring such that the colour of the edge e belongs to the set C_e for each $e \in E$. Such a colouring is called a **list edge-colouring** of G. An important step towards settling the Dinitz conjecture was made by Jansen [458] who proved that, if all lists have length $n+1$ (instead of n) then a solution always exists.

In order to apply results on kernel in digraphs we study the line graph of $K_{n,n}$. The definition of a line graph is analogous to that of a line digraph: $L(G)$ contains a vertex for each edge of G and two vertices in $L(G)$ are joined by an edge if and only if the corresponding edges have an end-vertex in common. It is easy to see that every list edge-colouring of $K_{n,n}$ corresponds to a list vertex colouring (in short a **list colouring**) of $L(K_{n,n})$ using the same sets (lists). Hence, in order to solve the Dinitz problem, it suffices to prove that no matter which sets $C_{11}, C_{12}, \ldots, C_{nn}$, each of size n, we associate with the n^2 vertices of $L(K_{n,n})$, there exists a proper vertex colouring of $L(K_{n,n})$ such that the colour of the vertex i is chosen from the corresponding set C_i.

Now we return to digraphs. The following lemma is attributed to Bondy, Boppana and Siegel in [15, Remark 2.4, p. 129] (see also [302]).

Lemma 12.4.1 *Let $D=(V,A)$ be a digraph and suppose that for each vertex $v \in V$ we are given a prescribed set $C(v)$ of colours satisfying $|C(v)| > d^+(v)$. If D is kernel perfect (i.e. every induced subdigraph of D has a kernel), then there exists a list colouring of $UG(D)$ which uses a colour from $C(v)$ for each $v \in V$.*

Proof: The proof is by induction on n, the case $n=1$ being trivially true. Fix a colour c which belongs to at least one of the sets $C(v)$, $v \in V$ and let $X(c) := \{v \in V | c \in C(v)\}$. By the assumption of the lemma the induced subdigraph $D\langle X(c)\rangle$ has a kernel Y. Now colour each vertex of V which belongs to Y by colour c (which is a proper choice by the definition of $X(c)$) and consider the digraph $D' = D - Y$ with colour sets $C'(v) = C(v) - \{c\}$. Notice that for each vertex $v \in X(c) - Y$ the out-degree of v in D' is at least one smaller than the out-degree of v in D and hence we have $|C'(v)| > d^+_{D'}(v)$ for all $v \in V(D')$. Furthermore, every vertex u that does not belong to $X(c)$ has $|C(u)| = |C'(u)|$. Thus, by the induction hypothesis, there is a list colouring of D' which uses a colour from $C'(v)$ for each $v \in V(D')$. Using that colouring along with the colour c for vertices in Y we achieve the desired colouring. □

From Lemma 12.4.1 we see that, if we can establish the existence of an orientation D of $L(K_{n,n})$ such that every induced subgraph of D has a kernel and $d_D^+(v) \leq n-1$ for each vertex v, then we have proved that $L(K_{n,n})$ has list chromatic number at most n as desired.

We show below that in order to obtain such an orientation we can use any n-edge-colouring of $K_{n,n}$ and orient appropriately. To prove the existence of a kernel in each induced subgraph we use the concept of stable matchings which we discuss below.

Below we assume that we are given a bipartite graph $B = (X \cup Y, E)$ and that for each vertex $u \in X \cup Y$ there is a fixed ordering $>_u$ on the neighbours of u. That is, $>_u$ induces an ordering $v_1 >_u v_2 >_u \ldots >_u v_{d_B(u)}$ on $N_B(u)$.

A matching M in $B = (X \cup Y, E)$ is **stable** with respect to the family of orderings $\{>_u | u \in X \cup Y\}$ if the following holds for all $uv \in E - M$: either $uy \in M$ for some y such that $y >_u v$ or $xv \in M$ for some x with $x >_v u$.

Stable matchings have an amusing real-life interpretation. Consider X as a set of men and Y as a set of women and let the existence of an edge $xy \in E$, $x \in X, y \in Y$ mean that person x and y might marry. As we saw in Theorem 3.11.2, given B we can determine in polynomial time the maximum number of men and women that can marry without anybody committing bigamy. However, in practice the fact that a man x and a woman y might marry does not mean that this particular choice is the optimal one for x or y. Hence, in a more realistic setting each person has a list of possible spouses and some ranking among these as to who would be the favourite choice down to the least wanted spouse (but still a possible choice). Now we see that this description corresponds to the orderings described above. Furthermore, stability of a given matching corresponds to saying that among the men and women that are paired for marriage there is no pair xy for which x prefers some other woman y' to y and at the same time woman y prefers some other man x' to x. So in some sense a stable matching corresponds to a situation where no pair is highly likely to split up.

The concept of stable matchings was introduced by Gale and Shapley who proved the following slightly surprising fact. We leave the proof as Exercise 12.18.

Theorem 12.4.2 *[290] For every bipartite graph $B = (X \cup Y, E)$ and every family of orderings $\{>_u | u \in X \cup Y\}$ which arises from a local linear ordering of the neighbours of each vertex in B, there exists a stable matching with respect to $\{>_u | u \in X \cup Y\}$.* □

In Exercise 12.19 the reader is asked to show by an example that it is not always true that there exists a maximum matching which is stable.

For more information about stable matchings see e.g. the papers [40, 41] by Balinski and Ratier. Now we are ready to describe Galvin's proof of the Dinitz conjecture.

Theorem 12.4.3 *[302] For every $n \geq 1$ the complete bipartite graph $K_{n,n}$ has list chromatic index n.*

Proof: Denote the vertices of $L(K_{n,n})$ by (i,j), $1 \le i,j \le n$, where (i,j) is adjacent to (i',j') if and only if $i = i'$ or $j = j'$, but not both. Let Q be any Latin square of size n (recall that this corresponds to a proper edge-colouring of $K_{n,n}$) and denote by Q_{ij} the ijth entry of Q. Let D_n be the oriented graph obtained from $L(K_{n,n})$ by orienting the edges as follows:

$(i,j) \rightarrow (i,j')$ if and only if $Q_{ij} < Q_{ij'}$ and
$(i,j) \rightarrow (i',j)$ if and only if $Q_{ij} > Q_{i'j}$ (see Figure 12.2).

1	2	3
2	3	1
3	1	2

Figure 12.2 The orientation of $L(K_{3,3})$ based on a Latin square of size 3.

It is easy to see that D is $(n-1)$-regular (Exercise 12.20). Thus, by Lemma 12.4.1 we just have to prove that every induced subdigraph of D has a kernel. To prove this we use Theorem 12.4.2.

Let D' be an arbitrary induced subdigraph of D and let $B = (X, Y, E)$ be the corresponding bipartite subgraph of $K_{n,n}$ induced by those edges for which the corresponding vertex (i,j) belongs to D'. For each vertex $i \in X$ we define an ordering $>_i$ of the neighbours of i in B by letting $j' >_i j$ whenever $(i,j) \rightarrow (i,j')$ in D. Similarly, for each $j \in Y$ we define the ordering $>_j$ of the neighbours of j in B by letting $i' >_j i$ whenever $(i,j) \rightarrow (i',j)$ in D.

According to Theorem 12.4.2 B has a stable matching M with respect to $\{>_u |u \in X \cup Y\}$. Since M is also a matching in $K_{n,n}$ the corresponding vertices are independent in D. Furthermore, it follows from the fact that M is stable with respect to $\{>_u |u \in X \cup Y\}$ that for every (i,j) such that $ij \notin M$, either there exist $j' \in Y$ such that $ij' \in M$ and $j' >_i j$ or there exists an $i' \in X$ such that $i'j \in M$ and $i' >_j i$. In the first case we have $(i,j) \rightarrow (i,j')$ and in the second case we have $(i,j) \rightarrow (i',j)$ in D. Thus we have shown that every vertex of D' which is not in M dominates a vertex in M. Hence M is a kernel and the proof is complete. $\square$

The idea of orienting $L(K_{n,n})$ as we did above is due to Maffray [543].

12.5 Homomorphisms – A Generalization of Colourings

Let D and H be digraphs. A mapping $f : V(D) \rightarrow V(H)$ is a **homomorphism** if it preserves arcs, that is, $xy \in A(D)$ implies $f(x)f(y) \in A(H)$. We will always write $f : D \rightarrow H$ or just $D \rightarrow H$ (when the actual homomorphism is not important). If there is no homomorphism from D to H, then we write $D \not\rightarrow H$. See an illustration in Figure 12.3. We say that G is **homomorphic** to H if $G \rightarrow H$. Similarly, for undirected graphs a homomorphism is an edge preserving map. To motivate what follows, we start our discussion from undirected graphs.

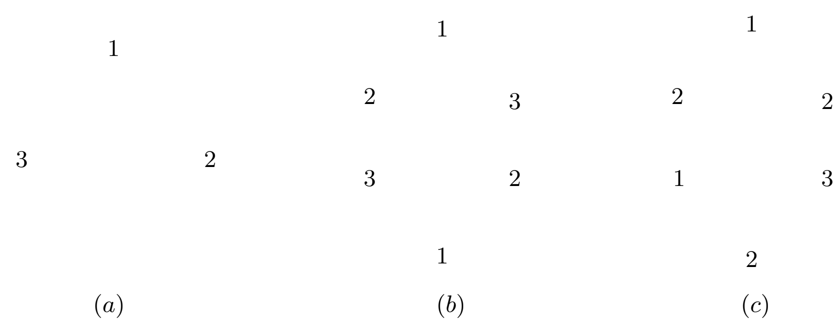


Figure 12.3 Illustrating the concept of a homomorphism; (a) A 3-cycle $\vec{C}_3$; (b) and (c) show digraphs with homomorphisms to $\vec{C}_3$ indicated by the labelling.

Recall that an undirected graph is k-colourable if we can assign numbers $1, 2, \ldots, k$ to its vertices such that adjacent vertices receive distinct numbers (colours). It is easy to see that an undirected graph G is k-colourable if and only if $G \rightarrow K_k$ (the complete graph on k vertices). Based on this observation we say that a (di)graph G is ***H*-colourable** for some (di)graph H if $G \rightarrow H$ and we call the mapping itself an ***H*-colouring** of G. Thus, if both G and H are given as part of the instance, the decision problem 'Is there a homomorphism of G to H?' properly includes k-colouring, and is therefore $\mathcal{NP}$-complete [474].

It is interesting to consider the same question when the graph H is fixed in advance. The ***H*-colouring problem** is formally defined as follows:

***H*-colouring**
Instance: A finite graph G.
Question:: Is there a homomorphism of G to H?

It is not difficult to see that a graph G has a homomorphism to a bipartite graph B if and only if G is 2-colourable (and hence is homomorphic to K_2). As we know this last question is the same as checking whether G is bipartite and hence easy (since this can be done using BFS or DFS). However in the case when the target graph (that is, the graph to which we want to map the given graph) H is not bipartite, the H-colouring problem is always difficult as shown by Hell and Nešetřil.

Theorem 12.5.1 *[412] If H is a fixed finite non-bipartite graph, then H-colouring is $\mathcal{NP}$-complete. If H is a fixed bipartite graph, then H-colouring is polynomial.* □

So, for undirected graphs the division between easy and hard problems is very clear: bipartite versus non-bipartite. For directed graphs the situation is much less clear. In the next pages we give some results and conjectures which illustrate the topic and interesting open problems.

First observe that, if $H \rightarrow H'$ for some induced subdigraph H' of H, then $D \rightarrow H$ if and only if $D \rightarrow H'$ (i.e. homomorphisms compose). Let H' be a subdigraph of H. A homomorphism $r : H \rightarrow H'$ is called a **retraction** if the restriction of r to H' is the identity map on H'. If there exists a retraction $H \rightarrow H'$, then H' is called a **retract** of H. A digraph is a **core** if and only if it has no proper retracts. The above observation shows that it suffices to study the H-colouring problem for those digraphs that are cores. Up to isomorphism every digraph has a unique core (see Exercise 12.21). Unfortunately deciding whether a digraph is indeed a core is a difficult problem.

Theorem 12.5.2 *[413] It is $\mathcal{NP}$-complete to decide whether a given input digraph is not a core*[3]. □

However for some classes of digraphs it is easy to tell whether they are cores or not. It is an easy exercise to show that every semicomplete digraph is a core (Exercise 12.22). It is slightly more difficult to characterize those semicomplete bipartite digraphs that are cores (Exercise 12.23).

Our first results deal with directed paths and cycles. The proof of the following easy observation by Maurer, Sudborough and Welzl is left as Exercise 12.25.

Proposition 12.5.3 *[556] There is a polynomial algorithm which decides if a given input digraph is homomorphic to the directed path $\vec{P}_k$.* □

When H is an arbitrary orientation of a path, Gutjahr Welzl and Woeginger proved that it is still polynomial (although much less trivial) to decide whether a given digraph is homomorphic to H.

Theorem 12.5.4 *[384] Let H be an arbitrary orientation of a path on k vertices. Then H-colouring is polynomial.* □

[3] Here the certificate showing that D is not a core is a mapping of D to a proper subdigraph of D.

Since homomorphisms compose it follows that, if $D \rightarrow H$ then every digraph which is homomorphic to D is also homomorphic to H. Thus one way of proving that D is not homomorphic to H would be to show a graph which is homomorphic to D but not to H. Using this approach Hell and Zhu [417] proved the following:

Theorem 12.5.5 *[417] Let D be a digraph on n vertices and P an oriented path on k vertices. Then $D \not\rightarrow P$ if and only if there exists an oriented path P' on at most $2^k n+1$ vertices such that $P' \rightarrow D$ and $P' \not\rightarrow P$.* □

It is easy to check whether $D \rightarrow \vec{C}_k$ holds for a given strong digraph $D = (V, A)$ and a given integer $k \geq 2$. Indeed, let the vertices of C_k be labelled $\{1, 2, \ldots, k\}$. Now to check whether $D \rightarrow \vec{C}_k$ we pick an arbitrary vertex $v \in V$ and map it to the vertex 1. After this the mapping of all other vertices in V fixed and it is easy to check whether this (unique) mapping is arc preserving. This can all be done in time $O(n+m)$ by using DFS from v to label and check whether each arc is preserved by the mapping at the same time. When D is not strongly connected it is a little more cumbersome to check whether $D \rightarrow C_k$, but it can still be done in time $O(n+m)$ (Exercise 12.26). Hence we have the following result ude to Maurer, Sudborough and Welzl:

Theorem 12.5.6 *[556] For every $k \geq 2$, $\vec{C}_k$-colouring is polynomial.* □

The following easy observation is merely a restatement of the definition of a homomorphism (recall that, by definition, digraphs have no loops):

Proposition 12.5.7 *Let D and H be digraphs. Then $D \rightarrow H$ if and only if there exists an extension $H_{ext} = H[\overline{K}_{a_1}, \overline{K}_{a_2}, \ldots, \overline{K}_{a_h}]$, $h = |V(H)|$ of H such that D is a subdigraph (not necessarily induced) of H_{ext}.* □

Let D be a digraph and C an oriented cycle of D. The **net length** of C is the absolute value of the difference between the number of forward arcs and the number of backward arcs with respect to an arbitrary fixed traversal of C (as an undirected cycle). Using Proposition 12.5.7 it is easy to prove the following characterization due to Häggkvist, Hell, Miller and Neuman-Lara of those digraphs which are homomorphic to a k-cycle (see also [556]):

Theorem 12.5.8 *[388] A digraph D is homomorphic to $\vec{C}_k$ if and only if the net length of every oriented cycle in D is divisible by k.*

Proof: Exercise 12.27. □

When H is an oriented cycle, the corresponding H-colouring problem may not be polynomial. Gutjahr showed in [383] that there are oriented cycles for which the corresponding H-colouring problem is $\mathcal{NP}$-complete. Hell and Zhu proved that, if H is an oriented cycle with net length different from zero, then a statement similar to Theorem 12.5.5 holds (necessity of these conditions is clear):

Theorem 12.5.9 *[418] Let C be an oriented cycle whose net length is not zero. A digraph D is homomorphic to C if and only if every oriented path homomorphic to D is also homomorphic to C, and the net length of every cycle of D is a multiple of the net length of C.* $\square$

It was shown in [384] by Gutjahr, Welzl and Woeginger and in [414] by Hell, Nešetřil and Zhu that the H-colouring problem may be $\mathcal{NP}$-complete even for orientations of trees. Hence classifying the complexity of the H-colouring problem for arbitrary digraphs seems almost hopeless.

When the target H has $\delta^0(H) > 0$ the picture seems clearer. If the core of H is a directed cycle, then H-colouring is polynomial by Theorem 12.5.6. In all other cases the problem seems to be difficult. In fact, the existence of two directed cycles in the core is often sufficient for the $\mathcal{NP}$-completeness of H-colouring as is illustrated by the next three results. The first result is an easy consequence of Theorem 12.5.1 (Exercise 12.28).

Theorem 12.5.10 *[412] Let H be the complete biorientation of an undirected graph G. If G is bipartite then H-colouring is polynomial and if G is not bipartite, then H-colouring is $\mathcal{NP}$-complete.* $\square$

The next two results due Bang-Jensen, Hell and MacGillivray, respectively, Bang-Jensen and Hell show that for some classes of digraphs, the number of cycles play an important role on the complexity of the H-colouring problem.

Theorem 12.5.11 *[77] Let H be a semicomplete digraph. If H has two or more directed cycles, then H-colouring is $\mathcal{NP}$-complete. If H has at most one directed cycle, then H-colouring is polynomial.* $\square$

Theorem 12.5.12 *[74] Let H be a semicomplete bipartite digraph which is a core. If H has two or more directed cycles, then H-colouring is $\mathcal{NP}$-complete. If H has at most one directed cycle, then H-colouring is polynomial.* $\square$

These results spurred further study [74, 85]. Based on the results in [74], Bang-Jensen and Hell made the following conjecture, which postulates a classification of the complexity of the H-colouring problem for all digraphs with $\delta^0(H) > 0$ and whose core is not a cycle. Note that a digraph H with $\delta^0(H) > 0$ is homomorphic to a directed cycle $\vec{C}_k$ if and only if its core is $\vec{C}_r$ for some r which is a multiple of k.

Conjecture 12.5.13 *[74] Let H be a digraph with $\delta^0(H) > 0$ and connected underlying graph. If H is homomorphic to a directed cycle, then H-colouring is polynomial. Otherwise H-colouring is $\mathcal{NP}$-complete.*

Since $\vec{C}_k$-colouring is polynomial as we mentioned above, the first statement is easy to see. Conjecture 12.5.13 has been verified for many classes of digraphs, see e.g. [74, 77, 78, 85, 383, 384, 412, 530, 531].

The main techniques for proving $\mathcal{NP}$-completeness H-colouring problems for directed graphs are described in [74, 77, 412]. These include the following two constructions both of which are due to Hell and Nešetřil [412]. We show how to use these tools below.

The indicator construction. Let I be a fixed digraph and let i, j be distinct vertices of I. The indicator construction (with respect to (I, i, j)) transforms a given digraph $H = (V, A)$ into the directed pseudograph $H^* = (V, A^*)$ where for every choice of (not necessarily distinct) $h, h' \in V$, the arc hh' is in A^* precisely when there exists a homomorphism $f : I \rightarrow H$ such that $f(i) = h$ and $f(j) = h'$. See Figure 12.4.

Lemma 12.5.14 *[412] If the H^*-colouring problem is $\mathcal{NP}$-complete, then so is the H-colouring problem.* □

j

i

(a) (b) (c)

Figure 12.4 Illustrating the indicator construction: (a) A digraph H; (b) An indicator I with special vertices i, j; (c) The result H^* of applying the indicator construction with respect to (I, i, j) to H. Undirected edges are used to indicate 2-cycles.

Note that H^* may have loops, in which case the H^*-colouring problem is trivial, since we can map every vertex to a vertex with a loop in H^*. Hence the construction is only useful if H^* has no loops. In this case H^* is always a digraph.

The sub-indicator construction. Let J be a fixed digraph with specified vertices $j, v_1, v_2, \ldots, v_t$. The sub-indicator construction with respect to $(J, j, v_1, v_2, \ldots, v_t)$ transforms a core $H = (V, A)$ with specified vertices $h_1, h_2, \ldots, h_t$ into the subdigraph $\tilde{H}$ of H which is induced by the vertex set $\tilde{V} \subseteq V$ where $\tilde{V}$ is defined as follows. Let W be the digraph obtained from the disjoint union of H and J by identifying v_i with h_i for $i = 1, 2, \ldots, t$. A vertex $v \in V$ belongs to $\tilde{V}$ if and only if there exists a retraction $f : W \rightarrow H$ which maps j to v. See Figure 12.5.

Lemma 12.5.15 *[412] Let H be a core. If the $\tilde{H}$-colouring problem is $\mathcal{NP}$-complete, then so is the H-colouring problem.* □

h j

v

(a) (b) (c)

Figure 12.5 Illustrating the sub-indicator construction; (a) a digraph H with a special vertex h; (b) the sub-indicator J with special vertices j, v; (c) the result $\tilde{H}$ of applying the sub-indicator construction with respect to (J, j, v) to (H, h).

To illustrate how to use the indicator and the sub-indicator construction, let us show that, if H is the digraph in Figure 12.4(a), then the H-colouring problem is $\mathcal{NP}$-complete. First apply the indicator construction with respect to the indicator shown in Figure 12.4(b) to H. This gives us the digraph H^* in Figure 12.4(c). By Lemma 12.5.14, H-colouring is $\mathcal{NP}$-complete if and only if H^*-colouring is $\mathcal{NP}$-complete. Now let J be the sub-indicator consisting of the complete biorientation of a 3-cycle with one vertex labelled j and an isolated vertex v_1. Let H' be the result of applying the sub-indicator construction with respect to (J, j, v_1) to H^*. Since v_1 is isolated, a vertex from H^* will be in H' precisely when it is itself on a complete biorientation of a 3-cycle in H^*. Hence H' is the complete biorientation of a 3-cycle. By Theorem 12.5.10 H'-colouring is $\mathcal{NP}$-complete and now we conclude by Lemma 12.5.15 that H^*-colouring and hence also H-colouring is $\mathcal{NP}$-complete.

Although the sub-indicator and the indicator constructions are very useful tools for proving the $\mathcal{NP}$-completeness of many H-colouring problems, there are digraphs H for which another approach such as a direct reduction from a different type of $\mathcal{NP}$-complete problem is needed. Such reductions are often from some variant of the satisfiability problem (see Section 1.10). The reader is asked to give such a reduction in Exercise 12.29.

For examples of other papers dealing with homomorphisms in digraphs see [135] by Brewster and MacGillivray, [416] by Hell, Zhou and Zhu, [590] by Nešetřil and Zhu, [680] by Sophena and [761, 762] by Zhou.

12.6 Other Measures of Independence in Digraphs

The definition of independence of vertex subsets in digraphs used in this book is by no means the only plausible definition of independence in digraphs. One may weaken the definition of independence in directed graphs in at least two other ways, both of which still generalize independence in undirected graphs.

(1) By considering induced subdigraphs which are acyclic. This gives rise to the **acyclic independence number**, $\alpha_{acyc}(D)$, which denotes the size of a maximum set of vertices X such that $D\langle X\rangle$ is acyclic.
(2) By considering induced subdigraphs which contain no 2-cycles. This gives rise to the **oriented independence number**, $\alpha_{or}(D)$, which denotes the size of a maximum set of vertices Y such that $D\langle Y\rangle$ is an oriented graph.

Both of these generalize the definition of independence in undirected graphs: if G is an undirected graph with independence number k then $\alpha_{acyc}(D) = \alpha_{or}(D) = k$, where D is the complete biorientation of G. Note that we always have

$$\alpha(D) \leq \alpha_{acyc}(D) \leq \alpha_{or}(D).$$

Furthermore, by our remark above, each of these parameters is at least as hard to calculate as $\alpha(D)$. In fact they seem much harder as they are $\mathcal{NP}$-hard already for tournaments, respectively semicomplete digraphs. The fact that $\alpha_{or}(D)$ is hard to calculate for semicomplete digraphs is left to the reader as Exercise 12.31. We prove below that calculating $\alpha_{acyc}(D)$ is $\mathcal{NP}$-hard even for tournaments. This result is due to Bang-Jensen and Thomassen and to Speckenmeyer.

Theorem 12.6.1 *[89, 681] The problem of finding a largest transitive subtournament in a tournament is $\mathcal{NP}$-hard.*

Proof: We show how to reduce the independent set problem for undirected graphs to our problem by a polynomial time reduction. This will imply the claim, since the independent set problem is $\mathcal{NP}$-hard, see e.g. [303]. Let $G = (V, E)$ be an undirected graph with vertex set $\{v_{1,0}, v_{2,0}, \ldots, v_{n,0}\}$. We form a tournament T as follows. We add, for each $i = 1, 2, \ldots, n$ a set of n+1 new vertices $\{v_{i,1}, v_{i,2}, \ldots, v_{i,n+1}\}$. Now T contains the directed arc $v_{i,k}v_{j,m}$ whenever $i > j$ or $i = j$ and $k > m$ unless $k = m = 0$ and $v_{i,0}, v_{j,0}$ are adjacent in G. In the last case T contains the arc $v_{j,0}v_{i,0}$. Now a vertex set S in G is a largest independent set if and only if $T - (V - S)$ is a largest transitive subtournament of T. $\square$

Jackson made the following conjecture:

Conjecture 12.6.2 *[453] Every digraph D with $\alpha_{or}(D) \leq \kappa(D)+1$ contains a hamiltonian path.*

As pointed out in Section 6.10.2, Conjecture 12.6.2 is not true if we replace $\alpha_{or}(D)$ by $\alpha(D)$.

A famous result due to Chvátal and Erdős [161] says that, if the vertex connectivity of an undirected graph G is at least as high as the size of a largest independent set, then G is hamiltonian. This is not true for digraphs, but as we pointed out in Proposition 3.11.12, at least there is a cycle factor in D if it is $\alpha(D)$-strong. Jackson proved that, if we consider the oriented independence number $\alpha_{or}(D)$, then an analogue of the Chvatal-Erdős theorem does exist.

Theorem 12.6.3 *[450] Let D be a digraph which is k-strong where $k = 2^{\alpha_{or}(D)}(\alpha_{or}(D)+2)!$. Then D has a hamiltonian cycle.* □

12.7 Matroids

In this section we give a very short introduction to matroids. The motivation for this is that algorithms for matroids are a useful tool for solving various graph theoretical problems. For an example of this we refer to Section 9.10 and Exercise 12.46. Unfortunately, due to lack of space we will not be able to describe in detail the algorithms for 2-matroid intersection and matroid partition (those are the ones used in the applications mentioned above). We refer the reader to the books [166] by Cook, Cunningham, Pulleyblank and Schrijver and [623] by Recski for detailed descriptions of these algorithms.

Definition 12.7.1 *Let S be a finite set and let $\mathcal{I}$ be a collection of subsets of S. The pair $M = (S, \mathcal{I})$ is a* **matroid** *if the following holds:*

(I1) $\emptyset \in \mathcal{I}$.
(I2) If $Y \in \mathcal{I}$ and $X \subseteq Y$, then $X \in \mathcal{I}$.
(I3) If $X, Y \in \mathcal{I}$ and $|X| < |Y|$, then there exists an element $y \in Y - X$ such that $X \cup \{y\} \in \mathcal{I}$.

Let $M = (S, \mathcal{I})$ be a matroid. A set $X \subseteq S$ such that $X \in \mathcal{I}$ is called **independent**. All other sets are **dependent**. A **base** of M is a maximal independent set. A **circuit** is a minimal dependent set. Let $\mathcal{B}$ denote the set of bases of M and $\mathcal{C}$ the set of circuits of M.

It follows directly from (I3) and the definition of a base that

$$\text{all bases of a matroid have the same size.} \tag{12.6}$$

Below we list some important properties of the bases of a matroid. (B1) follows from (I1). (B2) follows from (I3) and (B3) is left to the reader as Exercise 12.32.

Proposition 12.7.2 *Let $M = (S, \mathcal{I})$ be a matroid. The set $\mathcal{B}$ of bases of M satisfy the following:*

(B1) $\mathcal{B} \neq \emptyset$.
(B2) For all $B, B' \in \mathcal{B}$ *we have* $|B| = |B'|$.
(B3) Let $B, B' \in \mathcal{B}$. *For every* $b \in B$ *there exists an element* $b' \in B'$ *such that* $(B - b) \cup \{b'\} \in \mathcal{B}$. □

The other direction holds as well (see Exercise 12.35)

Proposition 12.7.3 *Let S be a finite set and $\mathcal{B}$ a collection of subsets of S which satisfies (B1)-(B3) above. Then there exists a matroid $M = (S, \mathcal{I})$ whose set of bases is precisely $\mathcal{B}$.* □

If $M = (S, \mathcal{I})$ is a matroid and $X \subseteq S$, then we say that a subset $Y \subseteq X$ is a **maximal independent subset of** X if $Y \in \mathcal{I}$ and $Y \subset Z \subseteq X$ implies $Z \notin \mathcal{I}$.

Lemma 12.7.4 *Let $M = (S, \mathcal{I})$ be a matroid and let $X \subseteq S$. All maximal independent subsets of X have the same size.*

Proof: Exercise 12.34. □

By Lemma 12.7.4, the following function is well-defined for all subsets of S.

$$r(X) = \max\{|Y| : Y \subseteq X \text{ and } Y \in \mathcal{I}\} \tag{12.7}$$

The **rank** of a matroid $M = (S, \mathcal{I})$ is the number $r(S)$, the size of a base in M.

Examples of matroids:

(1) Let $G = (V, E)$ be an undirected graph. Define $M(G)$ as $M(G) = (E, \mathcal{I})$, where $E' \in \mathcal{I}$ if and only if $G_{E'} = (V, E')$ has no cycle. Then $M(G)$ is a matroid (called the **circuit matroid of G**). To see this, it suffices to check (I3), since (I1),(I2) trivially hold. Let X, Y be subsets of E such that none of $G\langle X\rangle$ and $G\langle Y\rangle$ has a cycle and $|X| < |Y|$. It is easy to show that, if Z is independent in $M(G)$, then the number of connected components in $G\langle Z\rangle$ is $n - |Z|$, where n is the number of vertices in G. Thus $|X| < |Y|$ implies that the number of connected components of $G\langle X\rangle$ is larger than that of $G\langle Y\rangle$. Hence Y contains an edge y such that y joins two vertices which are in distinct components of $G\langle X\rangle$. This implies that $G\langle X \cup \{y\}\rangle$ is acyclic and hence $X \cup \{y\} \in \mathcal{I}$.
The bases of $M(G)$ are the (sets of edges of) maximal forests of G and a cycle of $M(G)$ is a fundamental cycle of G with respect to a maximal forest of G. The rank of $M(G)$ is $|V|$ minus the number of connected components of G.

(2) Let S be a set on n elements, and define $U_{n,k}$ for $k \leq n$ as follows: $U_{n,k} = (S, \{X \subseteq S : |X| \leq k\})$. This trivially gives a matroid called a **uniform matroid**. If $k = n$ we obtain a very special case in which all subsets are independent. This matroid is called the **free matroid** on n elements.

(3) Let $D = (V, A)$ be a digraph such that $\delta^-(D) > 0$ and define $\mathcal{B}$ as those subsets A' of A for which every vertex $v \in V$ has in-degree precisely one in $D\langle A' \rangle$. We show that $\mathcal{B}$ satisfies (B1)-(B3) of Proposition 12.7.2 and hence, by Proposition 12.7.3, $\mathcal{B}$ forms the set of bases of a matroid $M^-(D)$. Indeed, (B1) holds since $\delta^-(D) > 0$ and (B2) holds by the definition of $\mathcal{B}$. To see that (B3) is true consider sets $A', A'' \in \mathcal{B}$ and let $a' \in A'$. The arc a' enters a vertex x and in A'' there is exactly one arc a'' with head x. Now we see that $(A' - a') \cup \{a''\} \in \mathcal{B}$.
Similarly, if $\delta^+(D) > 0$, then we may define a matroid $M^+(D)$ whose bases are those subsets X of the arcs for which every vertex $v \in V$ has out-degree precisely one in $D\langle X \rangle$. This follows from the argument above by considering the converse of D.

The next result shows, in particular, that the rank function of a matroid is submodular. This is one of the reasons for the usefulness of matroids.

Proposition 12.7.5 *The rank function of $M = (S, \mathcal{I})$ satisfies the following:*

(R1) $0 \leq r(X) \leq |X|$ *for every* $X \in S$.
(R2) $X \subseteq Y$ *implies* $r(X) \leq r(Y)$.
(R3) For all $X, Y \subseteq S$: $r(X) + r(Y) \geq r(X \cap Y) + r(X \cup Y)$.

Proof: (R1) and (R2) follow from the definitions. To see that (R3) holds consider two subsets X, Y of S. We may assume that $X \neq Y$. Let A be a maximal independent subset of $X \cap Y$ and let B be an extension of A to a maximal independent subset of $X \cup Y$. Now using (R2) we have

$$\begin{aligned} r(X) + r(Y) &\geq |B \cap X| + |B \cap Y| \\ &= |B| + |A| \\ &= r(X \cup Y) + r(X \cap Y). \end{aligned} \tag{12.8}$$

□

12.7.1 The Dual of a Matroid

The **dual** of a matroid $M = (S, \mathcal{I})$ is the pair $M^* = (S, \mathcal{I}^*)$, where $\mathcal{I}^* = \{X \subseteq S : X \cap B = \emptyset \text{ for some base } B \text{ of M}\}$. In Exercise 12.37 the reader is asked to prove that M^* is a matroid. Note that the bases of M^* form precisely the set $\mathcal{B}^* = \{S - B : B \text{ is a base of } M\}$.

Proposition 12.7.6 *For any matroid M we have*

(i) $(M^*)^* = M$.
(ii) $r^*(X) = |X| + r(S - X) - r(S)$.

Proof: Exercise 12.38. □

A circuit in M^* is called a **cutset** or a **cocircuit** in M. It follows from the definition of M^* that a cocircuit of M is a minimal subset of S which has a non-empty intersection with all bases of M.

12.7.2 The Greedy Algorithm for Matroids

Let $M = (S, \mathcal{I})$ be a matroid. For every $X \in \mathcal{I}$ we define the set ext(X) by

$$\mathrm{ext}(X) = \{y \in S - X : X \cup \{y\} \in \mathcal{I}\}. \tag{12.9}$$

That is, ext(X) are precisely those elements y in $S - X$ such that y can be added to X without creating a dependent set.

Suppose we are given a weight function $w : S \rightarrow \mathcal{R}_+ \cup \{0\}$ on the elements of S. We let $w(X) = \sum_{x \in X} w(x)$. Our goal is to find an independent subset of S with maximum weight. Since $w(s) \geq 0$ for every $s \in S$ it follows that a maximum weight independent subset can always be assumed to be a base (using (I3), we may add extra elements of weight zero to X if X has maximum weight and is not a base). An **optimal base** is a base B such that $w(B) \geq w(B')$ for every $B' \in \mathcal{B}$.

The following simple algorithm $\mathcal{GA}$ is known as **the greedy algorithm for matroids**:

Input: A matroid $M = (S, \mathcal{I})$ and a weight function $w : S \rightarrow \mathcal{R}_+ \cup \{0\}$.
Output: an optimal base of M.

1. Let $X := \emptyset$;
2. If ext$(X) = \emptyset$ go to Step 5;
3. Choose an element $x \in$ ext(X) such that $w(x) = \max\{w(y) : y \in \mathrm{ext}(X)\}$;
4. Let $X := X \cup \{x\}$ and go to Step 2;
5. Return X;

Since the only maximal independent sets in M are bases, it follows that the greedy algorithm returns a base X of M. Such a base is called a **greedy base** of M. The following result due to Rado shows that the greedy algorithm works nicely for matroids:

Theorem 12.7.7 *[619] The greedy algorithm for matroids always finds an optimal base.*

Proof: Suppose there exists a matroid $M = (S, \mathcal{I})$ and weight function w such that the greedy algorithm does not find an optimal base of M. Let B_g be the greedy basis which is returned by the algorithm. By assumption, M has another base B such that

$$w(B_g) < w(B). \tag{12.10}$$

Since $\emptyset \subset B \cap B_g$ and $B_g \neq B$, there is a well-defined first iteration of the while loop in which $\mathcal{GA}$ chooses an element x which is not in B. Let A be the current independent subset found by the algorithm just before $\mathcal{GA}$ adds x (to A). Consider the independent set $A' = A \cup x$. By (I3), we can extend A' to a base B' of M by adding elements from B. It follows from this and (B2) that $B' = (B - y) \cup \{x\}$ for some $y \in B - B_g$. Since $A \cup y \subseteq B'$ we have $y \in \mathrm{ext}(A)$. Now it follows from that fact that $\mathcal{GA}$ chose x and not y when it extended A that we have $w(y) \leq w(x)$. However this means that

$$w(B_g) = w(B) - w(y) + w(x) \geq w(B),$$

contradicting (12.10). □

It can be shown that, if we have a collection $\mathcal{F}$ of subsets of a set S such that (I1) and (I2) hold, but (I3) does not, then there exists a non-negative real-valued weight function so that applying the algorithm $\mathcal{GA}$ to this collection of sets we never find an optimal basis (Exercise 12.39).

The reader who knows Kruskal's classical algorithm for finding a minimum weight spanning tree in a connected undirected graph G with weights on the arcs (see e.g. [169]) will have noticed the strong similarity between that algorithm and the algorithm $\mathcal{GA}$ above. In fact Kruskal's algorithm is precisely $\mathcal{GA}$ for the case when the input is the circuit matroid $M(G)$ of G.

12.7.3 Independence Oracles

What is a fast algorithm for matroids? How do we represent a matroid efficiently? These are important questions. In particular, it should be clear that in general it is infeasible to store information about a given matroid by a list of its independent sets. For example, if M is the uniform matroid $U_{n,k}$, we would have to store all subsets of size at most k of $\{1, 2, \dots, n\}$. On the other hand for $U_{n,k}$ it is very easy to decide whether a given subset of $\{1, 2, \dots, n\}$ is independent: simply calculate its size and check whether this is at most k. This illustrates that what is important is not having a list of all independent sets, but rather to be able to determine whether a given subset X of the ground set S is independent in M.

We shall assume that our matroids are always given in terms of the ground set S and a subroutine O_M which given $X \subseteq S$ decides whether X is independent in M or not. Such a subroutine O_M is called an **independence oracle** for $M = (S, \mathcal{I})$. We say that a matroid algorithm $\mathcal{A}$ for a matroid $M = (S, \mathcal{I})$

with independence oracle O_M is **fast** if the number of steps of $\mathcal{A}$ is polynomial in $|S|$ and any other inputs (such as a weight function), provided that we consider each call to O_M as taking constant time. With this assumption, the greedy algorithm is a fast matroid algorithm.

In order for a fast matroid algorithm to be useful in practice, we must be able to supply an independence oracle which works in polynomial time (and preferably very fast) In the case of Kruskal's algorithm above such an oracle exists, since a subset $X \subseteq E$ is independent in the circuit matroid of $G = (V, E)$ if and only if X induces a forest in G, something which can be checked in linear time by DFS, say (Exercise 12.41). Similarly, checking whether a subset is independent in $U_{n,k}$ can be done in linear time.

12.7.4 Union of Matroids

Let $M_i = (S, \mathcal{I}_i)$, $i = 1, 2, \ldots, k$ be matroids on the same ground set S. Define $\vee_{i=1}^k M_i = (S, \vee_{i=1}^k \mathcal{I}_i)$ as follows. A set $X \subseteq S$ is independent in $\vee_{i=1}^k M_i$ if and only if X can be decomposed as $X = X_1 \cup X_2 \cup \ldots \cup X_k$, where $X_i \in \mathcal{I}_i$ for $i = 1, 2, \ldots, k$. It is a non-trivial exercise (Exercise 12.42) to prove the following:

Proposition 12.7.8 *Let $M_i = (S, \mathcal{I}_i)$, $i = 1, 2, \ldots, k$ be matroids on the same ground set S. Then $\vee_{i=1}^k M_i$ is a matroid.* □

Note that, if X is independent in $\vee_{i=1}^k M_i$, then X has a partition into sets $X_1, X_2, \ldots, X_k$ such that X_i (which might be empty) is independent in M_i, $i = 1, 2, \ldots, k$. Thus deciding whether X is independent in $\vee_{i=1}^k M_i$ is equivalent to deciding whether X can be partitioned into k subsets $X_1, X_2, \ldots, X_k$ such that $X_i \in \mathcal{I}_i$ for $i = 1, 2, \ldots, k$.

The matroid partition problem: Let $M_i = (S, \mathcal{I}_i)$, $i = 1, 2, \ldots, k$ be matroids on the same ground set S and a subset $X \in S$. Does there exist subsets $X_1, X_2, \ldots, X_k$ of S such that $X = \cup_{i=1}^k X_i$ and $X_i \in \mathcal{I}_i$ for $i = 1, 2, \ldots, k$?

In Exercise 12.45 the goal is show that the question of deciding whether an undirected graph has k edge-disjoint spanning trees can be formulated as a matroid partition problem. Hence the following theorem implies the existence of a polynomial algorithm for deciding whether an undirected graph has k edge-disjoint spanning trees (see Exercise 12.46).

Theorem 12.7.9 *The matroid partition problem can be solved in polynomial time, provided we are given polynomial time realizable independence oracles for each of the matroids M_i, $i = 1, 2, \ldots, k$.* □

We refer the reader to Recski's book [623] for a description of a fast algorithm for the matroid partitioning problem. Note that, if $M = (S, \mathcal{I})$ is a matroid and X is a subset of S, then $M\langle X \rangle = (X, \mathcal{I}_X)$, where $\mathcal{I}_X = \{Y \in \mathcal{I} :$

$Y \subseteq X\}$ is also a matroid (Exercise 12.43). Hence, the matroid partitioning problem is equivalent to the problem of deciding whether the ground set S is independent in $\vee_{i=1}^{k} M_i$. This is the problem which is solved in [623].

12.7.5 Two Matroid Intersection

Another very useful topic on matroids is matroid intersection. By this we do not mean that, if M_1, M_2 are matroids on the same ground set S, then $M = (S, \mathcal{I}_1 \cap \mathcal{I}_2)$ is also a matroid. This is false as the reader can easily show by an example (Exercise 12.47). Instead we are interested in the following problem.

The matroid intersection problem: Given matroids $M_1 = (S, \mathcal{I}_1), M_2 = (S, \mathcal{I}_2)$ such that $r_1(S) = r_2(S)$. Find a maximum cardinality subset $T \subseteq S$ which is independent in each of M_1, M_2.

The next result shows that the matroid intersection problem and the matroid partition problem are closely related.

Theorem 12.7.10 *Let $M_1 = (S, \mathcal{I}_1), M_2 = (S, \mathcal{I}_2)$ be matroids on the same ground set S with $r_1(S) = r_2(S) = r$ and let $n = |S|$. There is a common base of M_1, M_2 if and only if $M_1 \vee M_2^* = U_{n,n}$.*

Proof: If X is a base of M_1 and of M_2, then $S - X$ is independent in M_2^* and hence $S = X \cup (S - X)$ is independent in $M_1 \vee M_2^*$, implying that this is the free matroid on $n = |S|$ elements.

To prove the converse, suppose S is independent in $M_1 \vee M_2^*$. Then S can be partitioned as $S = S_1 \cup S_2$ where $S_1 \in \mathcal{I}_1$, $S_2 \in \mathcal{I}_2^*$.

Now we obtain

$$
\begin{aligned}
|S| = |S_1| + |S_2| &= r_1(S_1) + r_2^*(S_2) \\
&\leq r_1(S) + r_2^*(S) \\
&= r + (|S| - r_2(S)) \qquad (12.11) \\
&= r + (|S| - r) \\
&= |S|.
\end{aligned}
$$

This implies that $r_1(S_1) = r$ and $r_2^*(S_2) = |S| - r_2(S)$. Thus S_1 is a base of M_1 and $S_2 (= S - S_1)$ is a base of M_2^*. Now we see that S_1 is a common base of M_1 and M_2. □

The following result is due to Edmonds:

Theorem 12.7.11 *[212] The matroid intersection problem can be solved in polynomial time, provided we are given polynomial time realizable independence oracles for M_1, M_2. Furthermore, under the same assumptions, one* *can find in polynomial time a maximum (or minimum) weight common independent subset with respect to any given real-valued weight function w on S.* □

For a description of a polynomial algorithm for (weighted) matroid intersection see e.g. [166, 623].

Matroid intersection is a very useful tool for modeling (and solving) many combinatorial optimization problems.

For instance the problem to find a minimum weight cycle factor in an arc weighted digraph can be formulated as a weighted two matroid intersection problem. Consider the intersection of the matroids $M^-(D), M^+(D)$ which were defined in the beginning of this section. There is a common base of these matroids if and only if D has a cycle factor and furthermore, the minimum weight of a common base equals the minimum weight of a cycle factor. Two more example are given in Section 9.10 and Exercise 12.48.

12.7.6 Intersections of Three or More Matroids

If we consider three or more matroids all on the same ground set and ask for a common base of these, then this problem contains quite a few difficult problems as special cases as we shall see below.

The k-matroid intersection problem: Given matroids $M_i = (S, \mathcal{I}_i)$, $i = 1, 2, \dots, k$ on the same ground set. Does there exist a set $X \subseteq S$ such that X is a base of M_i for $i = 1, 2, \dots, k$?

Theorem 12.7.12 *The k-matroid intersection problem is $\mathcal{NP}$-complete for $k \geq 3$.*

Proof: It suffices to prove the theorem for $k = 3$ since the proof can easily be extended to higher k by using several copies of the same matroid. We will prove that the $\mathcal{NP}$-complete problem of deciding the existence of a hamiltonian path which starts in a prescribed vertex u and ends in a prescribed vertex v in a digraph (see Exercise 6.3) can be reduced to the 3-matroid intersection problem in polynomial time.

Let $D = (V, A)$ be a digraph with specified vertices $u, v \in V$. Define $M_i = (S, \mathcal{I}_i)$, $i = 1, 2, 3$ as follows:

$S = A$;
$M_1 = M(UG[D])$;
$X \in \mathcal{I}_2$ if and only if there is no arc entering u in $D_X = (V, X)$ and every other vertex has at most one arc entering it in D_X.
$Y \in \mathcal{I}_3$ if and only if there is no arc leaving v in $D_Y = (V, Y)$ and every other vertex has at most one arc leaving it in D_Y.

We argued in Section 9.10 that $M_2 = (A, \mathcal{I}_2)$ is a matroid and similarly $M_3 = (A, \mathcal{I}_3)$ is seen to be a matroid. It is easy to see that D has a Hamilton path P from u to v if and only if M_1, M_2, M_3 have a common base (the arcs of a Hamilton path correspond to a common base of M_1, M_2, M_3).

Note that the reduction above is a polynomial one because given an instance $[D, u, v]$ of the hamiltonian path problem with prescribed initial and terminal vertices, we can easily extract the arc set of A and hence the ground set of the 3 matroids above. □

12.8 Finding Good Solutions to $\mathcal{NP}$-Hard Problems

In this book we have encountered many problems which are $\mathcal{NP}$-hard. Several of these such as the feedback arc set problem (denoted FAS below for convenience) are of significant practical interest. Part of our discussion below will focus on the feedback arc set problem, but most of the discussion is valid for the majority of $\mathcal{NP}$-hard problems we know of.

Clearly we could solve the FAS problem if we simply try all subsets of the arc set and take the smallest feedback arc set we find. Of course this would take exponential time and even for digraphs with at most 100 arcs this process would be extremely time consuming if not infeasible, even on the fastest computers available today and in the near future.

A better approach is to try to solve the problem at hand by a clever way of examining those among the set of all possible solutions which could be a candidate for an optimal solution. If we already know a feedback arc set with 20 arcs and we have a (preferably fast) way of detecting that among all subsets from a certain collection of subsets of arcs, no feedback arc set with less than 20 arcs exists, then we do not have to consider these subsets any more, since no optimal solution can be found here. This idea, which we will not describe in detail here, is one of the two main ingredients in a general method called **branch and bound**, (see e.g. the book [600] by Papadimitriou and Steiglitz). Branch and bound can be used to solve small instances of the FAS problem, but already for digraphs with 100 vertices it becomes very time consuming to find an optimal solution.

In the rest of this section we describe methods that do not give us any guarantee on the quality of the solution and sometimes not even on the running time of an implementation of the method. But experimental evidence suggests that in practice some of these so-called **heuristics** do give solutions which are close to the optimum solution. Furthermore, they often run very fast when implemented carefully on a PC. Such methods may not seem very interesting to the theoretician who may only consider methods that provably obtain the optimum or some approximation guarantee for the solution as worth studying. However, in practice the situation is entirely different: the engineer who has been asked to find a reasonable solution to an instance of the FAS problem, say, cannot really use this attitude. What (s)he needs is a way to get a good solution and some indication that this solution is better than a random solution and cannot be easily improved on (recall the discussion concerning the domination number of algorithms for the TSP problem in Section 6.12). Certainly such a solution will often be much better than one that could be found at hand by the engineer.

We start with a very simple method for finding a feedback arc set which is locally optimal. We assume that we are given a directed multigraph $D = (V, A)$ and an ordering $s = v_1, v_2, \ldots, v_n$ of V. Given this ordering we can easily determine the set of forward arcs A_f (as those arcs v_iv_j for which $i < j$) and clearly $A - A_f$ is a feedback arc set of D. Now suppose that there are indices i, j such that by deleting the vertex v_j and reinserting it between v_i and[4] v_{i+1} we obtain a smaller feedback arc set. The effect on the value of the feedback arc set can be calculated easily without reconsidering all arcs (Exercise 12.49).

By a **solution** s we mean an ordering of the vertices of D. The **value** $v(s)$ of a solution s is the number of backwards arcs with respect to s. We say that two solutions s, s' are **neighbours** if we can obtain one from the other by deleting one vertex and reinserting it somewhere else in the ordering of the remaining vertices. With respect to this definition of a neighbour of a solution s we can define the **neighbourhood** $N(s)$ of s as the set of solutions that are neighbours of s. Now we can describe a very simple heuristic which we call **1-OPT** for the FAS problem:

1-OPT
Input: A directed multigraph $D = (V, A)$;
Output: An ordering of V (for which the backwards arcs form a feedback arc set in D).

1. Start with a solution s corresponding to a random permutation of V;
2. If there exists a neighbour s' of s such that $v(s') < v(s)$; then take $s := s'$ as the new current solution and repeat this step;
3. Output the locally optimal solution s and halt.

It is easy to show (Exercise 12.50) that the 1-OPT algorithm will halt after finitely many steps with a solution that is locally optimal. Here **locally optimal** means that the number of backwards arcs cannot be decreased by moving a single vertex.

There are several other ways of defining sensible neighbourhoods of a solution to the FAS problem. For example, one could consider all solutions that can be obtained by interchanging the positions of two vertices in the given ordering (see Exercise 12.52 and Exercise 12.53). Experimental evidence found by Olsen [594] suggests that this last way of choosing the neighbourhood does not produce as high quality solutions as the one above.

Although 1-OPT produces solutions that are in general much better than a random choice, it only guarantees that the final solution found is locally

[4] We allow $i = n$ and $i = 0$ with the obvious meaning of v_{i+1} and v_0.

optimal. Furthermore, since a new solution is only taken if it improves the objective function, the algorithm cannot escape a local minimum.

This can be remedied somewhat by restarting the algorithm several times, each time starting from a new random permutation of the vertices. Since the algorithm is usually very fast it is possible to restart it many times (from different random solutions) and then take the best solution among the local optima which were found.

Another method to escape local minima would be to allow a neighbour s' of the current solution s with $v(s') > v(s)$ to be chosen with some positive probability. However, unless this probability decreases as the number of steps increases the method may never converge towards a local minimum.

This problem is handled in the next method which we briefly describe. In the method called **simulated annealing** the basic idea is to allow a neighbouring solution s' with $v(s') > v(s)$ to be chosen with a probability p which depends both on $\tau = v(s') - v(s)$ and the number of steps taken by the algorithm so far.

Below we describe the generic simulated annealing method for a minimization problem over the set S of possible solutions and with objective function f and neighbourhood structure N. Note that this is a **meta-heuristic**, i.e. it is a scheme that can be applied to many types of combinatorial optimization problems rather than just one specific problem.

Generic Simulated Annealing

1. Select an initial solution s_0;
2. Initialize control parameter t to a value t_0;
3. Select a reduction method M for the control parameter t;
4. Repeat $K(n)$ times:
5. Choose randomly a neighbour $s \in N(s_0)$;
6. Let $\tau := f(s) - f(s_0)$;
7. If $\tau \leq 0$ then $s_0 := s$
8. Else let $s_0 := s$ with probability $\exp(-\tau/t)$;
9. Let $t := M(t)$;
10. If the stopping condition is satisfied then return the best solution encountered and halt. Otherwise go to Step 4.

Although we did not write it above, it is understood that the algorithm also keeps track of the best solution found so far (note that this may not coincide with the current solution s_0.).

It is evident from the (loose) description above that any implementation of the method involves making several choices about how to perform the various steps. We discuss briefly the general idea below and refer to the survey [203] by Dowsland and the experimental evaluation of simulated annealing by Johnson, Aragon, McGeoch and Schevon described in [464] for more details. It is important to note that finding a good set of values/methods to implement the algorithm is by no means always a trivial task. Part of this process consists of **tuning** the parameters t_0,$K(n)$, the method M for decreasing t and the stopping criterion. This is done by performing a number of runs with all but one parameter fixed and then selecting values that look promising. After some stages of this process, one may arrive at a choice for the parameters which does not seem easy to improve (based on the test data used). See also Exercise 12.51. However, experimental evidence reported by Hansen [397] and Olsen [594] indicate that for a problem such as FAS it is not too hard to make a set of choices which will make the algorithm perform quite well.

The initial solution can be chosen arbitrarily or it may be a local optima found by 1-OPT, say. The control parameter t should be initialized so that in the beginning there is a fair chance that the algorithm will accept a neighbour with a higher f value than the current solution s_0. Normally this is done by starting from a random solution and then performing, say, 1000 steps of the algorithm while keeping track of the number of neighbours who are accepted as the new current solution[5]. The **initial acceptance rate** is the fraction of accepted solutions over the total number of neighbours tested (1000 above). Experiments reported in e.g. [464] suggest that acceptance rates in the interval $[0.3, 0.9]$ all work well (these experiments were not for the FAS problem, but the conclusion also seems to hold for FAS [594]).

Experiments show that the actual reduction method used to reduce t after every cycle of $K(n)$ steps is not as important as the rate at which t is reduced. This rate should be as slow as possible (that is, as time allows) [464]. In fact, whereas in general no theoretical guarantee exists for the quality of a solution found by local search heuristics such as 1-OPT, it can actually be shown (see e.g. the book [30] by Arts and Korst) that under ideal conditions (such as reducing the parameter t infinitely slowly, taking a very large number of steps for each value of t and using a neighbour structure that allows one to reach some optimal solution from an arbitrary solution) simulated annealing will in fact find an optimal solution. Of course such a result is only of theoretical interest, but the nice thing is that, since some of these results are based on Markov chains, the results suggest that the slower one reduces t and the higher $K(n)$ (as a function of the size of the neighbourhood), the better results one should obtain. This thesis seems to be true for several applications of simulated annealing (see e.g. [203, 464]).

It is common to use a simple geometric reduction method where we set $t := rt$ for some fixed number $0 < r < 1$ which is close to one. Experiments suggests that $r = 0.95$ is generally a good choice [464]. The number of steps $K(n)$ for each value assumed by t should be at least a linear function in the size of the neighbourhood of an arbitrary solution. Finally it is common to use as a stopping condition that there has been no improvement in the current solution for some number N of moves. Another possibility is to use the **current acceptance rate** (calculated similarly as the initial acceptance

[5] This includes those that have a better (or equal) value than the current solution as well as those that are worse, but are chosen in the probabilistic step 8.

rate by keeping track of the number of accepted moves over the last, say, 1000 steps) as a measure and stop when this rate gets below, say 1 percent. One may also decide to stop when the control parameter becomes smaller than a prescribed value t_s. Note that in the last case, the number of steps performed by the algorithm is always the same (for $K(n)$ and M fixed).

Due to space limitations we will not go into further details of the method. The interested reader is encouraged to work out the programming projects of Exercises 12.51 and 12.52. The success of the simulated annealing algorithm on various combinatorial optimization problems varies of course (and also depends strongly on the ingenuity of the persons who experiment with it, in particular in the tuning phase). For a problem like the linear ordering problem, the algorithm seems to perform very well. Hansen showed [397] that when applied to real-world instances of the linear ordering problem of sizes up to 75 vertices, the simulated annealing algorithm very often finds the optimal solution within a few minutes on a standard PC and the ones that were not optimal were within one percent of the optimal values.

For a very thorough discussion on how to tune simulated annealing algorithms as well as a comparison of simulated annealing with other methods on various combinatorial problems we refer to the experimental papers [464, 465] by Johnson, Aragon, McGeoch and Schevon. There are several other metaheuristics which work quite well for many types of combinatorial optimization problems. For a detailed discussion we refer the reader to the book [628] edited by Reeves.

12.9 Exercises

12.1. Show that given an oriented graph D one can check whether D satisfies Conjecture 12.1.1 in time $O(nm)$. Which representation of the oriented graph may we assume to obtain this complexity?

12.2. Show that finding a median order of an arc weighted tournament (that is, an order which minimizes the total weight of the backwards arcs) is $\mathcal{NP}$-hard by giving a polynomial reduction of the feedback arc set problem to this problem.

12.3. **(+)** Give a short and direct argument which shows that there exists a function $f(n)$ so that every tournament on $f(n)$ vertices contains every oriented tree on n vertices. Hint: consider removing a leaf from a tree and then applying induction.

12.4. Construct your own examples of semicomplete PCDs, for which FSO and the score method produce different results.

12.5. Prove by induction on n the formula for $f_K(\alpha)$ in Lemma 12.2.1.

12.6. Prove the following. Let $K = (V, A, \epsilon)$ be a complete PCD with n vertices, and let α be an ordering of K. Then $f_K(\alpha) + b_K(\alpha) = n(n^2 - 1)/6$. Hint: use induction on n.

12.7. Prove that the problem to find a forward optimal ordering of a PCD is $\mathcal{NP}$-hard (Gutin and Yeo [373]) .

12.8. (−) Formulate and prove a lemma for forward orderings analogous to Lemma 12.2.6.

12.9. Prove Lemma 12.2.7 using Lemma 12.2.6.

12.10. Prove Theorem 12.2.8.

12.11. (−) Compute the proper backward ranks of the vertices of the uniform PCD corresponding to the digraph D in Figure 12.6.

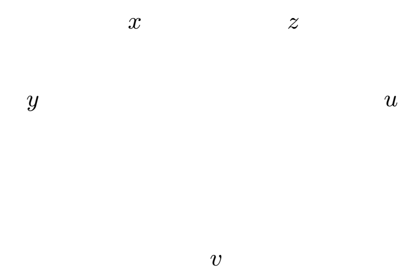


Figure 12.6 A semicomplete 3-partite digraph D.

12.12. Using Theorem 12.2.8 construct a polynomial algorithm to find proper backward ranks of the vertices of a semicomplete multipartite PCD.

12.13. Prove Theorem 12.2.9.

12.14. (−) Find the proper backward ranks of the vertices of the uniform PCD corresponding to the digraph $D - vz$, where the digraph D is depicted in Figure 12.6.

12.15. Give a direct proof that every acyclic digraphis kernel-perfect. Prove that an acyclic digraph has a unique kernel (von Neumann and Morgenstein [731]).

12.16. Prove that $C_{43}(\{1,7,8\}) - (1,2)$, $C_{43}(\{1,7,8\}) - (1,8)$ and $C_{43}(\{1,7,8\}) - (1,9)$ have kernels, where $C_{43}(\{1,7,8\})$ is a circular digraph.

12.17. (−) Give a construction of a Latin square of size n for each integer $n \geq 1$.

12.18. (+) Prove Theorem 12.4.2.

12.19. Construct a bipartite graph $B = (X \cup Y, E)$ with a family $\{>_u |u \in X \cup Y\}$ of orderings induced from local orderings of the neighbours of each vertex, such that no maximum matching of B is stable.

12.20. (−) Argue that the oriented graph D in the proof of Theorem 12.4.3 is $(n-1)$-regular.

12.21. (+) Prove that every digraph has a unique core (up to isomorphism).

12.22. (−) Prove that every semicomplete digraph is a core.

12.23. **Characterizing core semicomplete bipartite digraphs**. Prove the following theorem due to Bang-Jensen and Hell:

Theorem 12.9.1 *[74] Let B be a semicomplete bipartite digraph with vertex partition X, Y. Then B is a core if and only if*
(a) B is a 2-cycle, or
(b) For all $x, y \in X$ such that $x \neq y$, either $y \in N^{+2}(x)$ or $x \in N^{+2}(y)$ and for all $u, v \in Y$ such that $u \neq v$, either $u \in N^{+2}(v)$ or $v \in N^{+2}(u)$.

12.24. Show that there is a polynomial algorithm which transforms a given semicomplete bipartite digraph into its core. Hint: use Theorem 12.9.1.

12.25. Prove Proposition 12.5.3. Hint: first show that you can assume that the input digraph is acyclic and then use the acyclic ordering.

12.26. **A polynomial algorithm for $\vec{C}_k$-colouring**. Complete the description from the text to an $O(n+m)$ algorithm which, given an arbitrary digraph D of order n and size m, either finds a homomorphism $D \rightarrow \vec{C}_k$ or a proof that $D \not\rightarrow \vec{C}_k$.

12.27. Prove Theorem 12.5.8.

12.28. **(−)** Prove Theorem 12.5.10.

12.29. **(+) Reducing 3-SAT to an H-colouring problem**. Let H be the digraph in Figure 12.7(a) and let Y be the digraph in Figure 12.7(a).

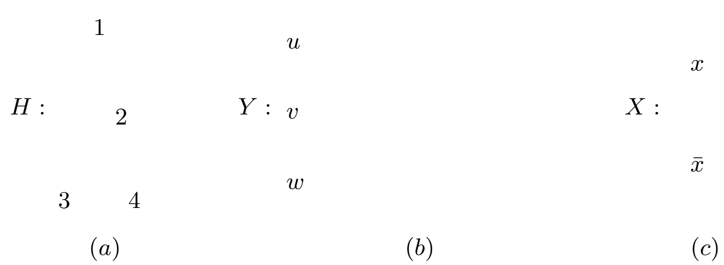


Figure 12.7 (a): the digraph H; (b) the digraph Y; the digraph X.

(i) Prove that for every H-colouring of Y, at least one of the vertices u, v, w is not mapped to 1.
(ii) Prove that every partial H-colouring of Y in which at most two of the vertices u, v, w is mapped to 1 can be extended to an H-colouring of Y.
(iii) **(−)** Prove that in every H-colouring of X, either x is coloured 1 and $\bar{x}$ is coloured 2 or vice versa.
(iv) **(+)** Use (i)-(iii) to construct in polynomial time for a given instance $\mathcal{F} = C_1 * C_2 * \ldots * C_m$ of 3-SAT a directed graph $D[\mathcal{F}]$ such that $D[\mathcal{F}] \rightarrow H$ if and only if $\mathcal{F}$ is satisfiable. Hint: use a copy of X for each variable and a copy of Y for each clause and piece together according to the formula $\mathcal{F}$.

12.30. **(+)** Prove, using a similar reduction to that outlined in Exercise 12.29 that, if H is the strong tournament on four vertices, then the H-colouring problem is $\mathcal{NP}$-complete.

12.31. Prove that calculating α_{or} is $\mathcal{NP}$-hard, even for semicomplete digraphs. Hint: reduce the independence number problem for undirected graphs to this problem.

12.32. Prove that (B3) holds for any matroid.

12.33. Is it true that $M = (S, \mathcal{I})$ is a matroid if and only if it satisfies (I1), (I2) and (I3'): All maximal elements of $\mathcal{I}$ have the same size?

12.34. $(-)$ Prove Lemma 12.7.4.

12.35. Prove Proposition 12.7.3.

12.36. **Circuit axioms for a matroid.** Prove the following Proposition. Hint: use (R3) and the fact that $C - x$ is independent for every circuit C and every $x \in C$.

Proposition 12.9.2 *Let $\mathcal{C}$ be the set of circuits of the matroid $M = (S, \mathcal{I})$. Then the following holds:*
(C1) If $C, C' \in \mathcal{C}$ and $C \subseteq C'$, then $C = C'$.
(C2) If $C, C' \in \mathcal{C}$, $C \neq C'$ and $u \in C \cap C'$, then there exists a circuit $Z \in \mathcal{C}$ such that $Z \subseteq C \cup C' - u$.

12.37. Prove that, if M is a matroid, then the dual M^* is also a matroid.

12.38. Prove Proposition 12.7.6.

12.39. **$(+)$ Fooling the greedy algorithm for families of subsets which are not matroids**. Suppose $\mathcal{F}$ is a collection of subsets of a set S which satisfies (I1), (I2), but not (I3). Construct a weight function w such that the algorithm $\mathcal{GA}$ will not find an optimal basis (Edmonds [213]).

12.40. $(+)$ Prove the following result:

Theorem 12.9.3 *Let $M = (S, \mathcal{I})$ satisfy (I1),(I2). The greedy algorithm $\mathcal{GA}$ finds an optimal base for M for every choice of non-negative real-valued weight function w on S if and only if M is a matroid.*

Hint: show that, if $A = \{a_1, \ldots, a_k\}$ and $B = \{b_1, \ldots, b_k, b_{k+1}\}$ both belong to $\mathcal{I}$, then one can choose a weight function w on the elements of S so that $\mathcal{GA}$ will always choose A as the first k elements and unless there is a $b_i \in B$ such that $A \cup \{b_i\} \in \mathcal{I}$, $\mathcal{GA}$ will not reach an optimal base.

12.41. Describe an $O(n + m)$ algorithm for deciding whether an undirected graph on n vertices and m edges has a cycle.

12.42. $(+)$ Prove Proposition 12.7.8. Hint: it suffices to prove the claim for two matroids. Consider a counterexample X, Y to (I3) with $X = X_1 \cup X_2$ and $Y = Y_1 \cup Y_2$, $X_1, Y_1 \in \mathcal{I}_1$, $X_2, Y_2 \in \mathcal{I}_2$ and $|X_1 \cap Y_2| + |X_2 \cap Y_1|$ is maximum.

12.43. Prove that $M\langle X \rangle$ defined in Section 12.7 is a matroid.

12.44. Let $D = (V, A)$ be a digraph with two vertices s, t such that $\lambda(s, t) \geq k$ for some k. Define $\mathcal{I}$ by $\mathcal{I} = \{X \subseteq A : \lambda_{D-X}(s, t) \geq k\}$. Show by an example that $(A, \mathcal{I})$ is not always a matroid. $(+)$ Can you characterize those digraphs for which $(A, \mathcal{I})$ is actually a matroid?

12.45. **$(+)$ Testing for k edge-disjoint spanning trees in graphs.** Show how to formulate the problem of deciding whether an undirected graph G has k edge-disjoint spanning trees as a matroid partition problem.

12.46. **(+) An algorithm for deciding the existence of k edge-disjoint spanning trees.** Use the formulation in Exercise 12.45 to derive a polynomial algorithm for deciding whether an undirected graph has k edge-disjoint spanning trees. Remember to justify that the needed oracles can be implemented as polynomial algorithms.

12.47. Give an example of two matroids M_1, M_2 on the same ground set S for which $M = (S, \mathcal{I}_1 \cap \mathcal{I}_2)$ is not a matroid.

12.48. **(+) Formulating the maximum (weight) matching problem for a bipartite graph as a (weighted) matroid intersection problem.**
(a) Show how to formulate the question of deciding the existence of a matching of size n in a bipartite graph $G = (U, V, E)$ on $2n$ vertices as a matroid intersection problem.
(b) Show how to solve the problem of finding a maximum weight matching of size n in the graph G above if we are given nonnegative weights on the edges of G.
(c) Argue that one can in fact find a maximum matching in any bipartite graph in polynomial time, using an algorithm for the matroid intersection problem.

12.49. Consider the 1-OPT method for the FAS problem. Describe how to determine, in linear time, the number of backwards arcs with respect to the ordering we obtain from $v_1, v_2, \ldots, v_n$ after removing one vertex from position j and reinserting it between v_i and v_{i+1}.

12.50. Prove that the 1-OPT algorithm applied to the feedback arc set problem will always halt. Then give a good bound on the number of steps taken by the algorithm.

12.51. **(+) Project: Implementing a simulated annealing algorithm for the feedback arc set problem**. The purpose of this project is to implement a version of simulated annealing which will allow one to obtain good solutions for moderately sized instances of the feedback arc set problem ($n \leq 500$). Use the details described in Section 12.8 along with the neighbourhood structure which we used in the 1-OPT algorithm. Perform test on various test data (such a randomly generated data and data for which a good feedback arc set is already known) in order to investigate the following issues[6]:
1. How much does the initial value of t (measured in terms of the resulting initial acceptance rate) influence the quality of the solution?
2. Is there a clear dependence of the value of the final solution on the value of the initial solution? Is is better to start from a good solution than a random one?
3. How important is it to decrease t slowly?
4. How many iterations should be performed between two consecutive reductions of t? Try to find a good estimate and see how it depends on the size of the input graph.
5. Try to combine the simulated annealing algorithm and 1-OPT by either rounding off a calculation by simulated annealing by an execution of 1-OPT, or by using 1-OPT at every step of the simulated annealing

[6] It is understood that as you vary one parameter, all other parameters are fixed at values which have either been found to be good experimentally already, or are as described in Section 12.8.

algorithm and using the value of resulting solution s' (based on the current solution s) as the (modified) objective function for the algorithm, i.e. take $f(s)$ to be the number of backwards arcs in the locally optimal solution s' and accept a new proposed neighbour of its f-value is better than $f(s)$, or it passes the test in Step 8 of the algorithm.

12.52. Instead of defining the neighbourhood of a solution s (an ordering of the vertices) to the FAS problem as we did in Section 12.8, we may also say that two solutions (orderings) s, s' are neighbours if we can obtain one from the other by interchanging the positions of two vertices v_i, v_j in the ordering. Try to work out Exercise 12.51 with this choice of neighbourhood instead and compare the results. Which neighbourhood choice would you think is the best and why? Carry out computation experiments to check this.

12.53. **Project: comparing various local search algorithms for the feedback arc set problem**. Consider the following heuristics for the FAS problem.
(a) 1-OPT.
(b) 2-OPT which uses the neighbourhood defined in Exercise 12.52 and swaps two vertices as long as there is a pair such that swapping these will improve the objective function.
(c) Steepest descent 1-OPT: Same as 1-OPT, except now we look at all neighbours of the current solution s and take the one whose objective function is the best if any has a lower value. Otherwise we stop.
(d) Steepest descent 2-OPT: Same as above, but for 2-OPT.
Implement each of these and compare them on various test data to see which one finds the best solution and compare their running times. Then try the same with probabilistic versions where the heuristics are restarted a number of times from random starting solutions.

References

1. A. Ádám. Problem. In *'Theory Graphs Applications', Proc. Coll. Smolenice*, pages 12–18, Czech. Acad. Sci. Publ., 1964.
2. A. Ádám. Bemerkungen zum graphentheoretischen Satze von I. Fidrich. *Acta Math. Acad. Sci. Hungar.*, 16:9–11, 1965.
3. R. Aharoni and R. Holzman. Fractional kernels in digraphs. *J. Combin. Theory Ser. B*, 73(1):1–6, 1998.
4. R. Aharoni and C. Thomassen. Infinite, highly connected digraphs with no two arc-disjoint spanning trees. *J. Graph Theory*, 13(1):71–74, 1989.
5. A.V. Aho, M.R. Garey, and J.D. Ullman. The transitive reduction of a directed graph. *SIAM J. Computing*, 1(2):131–137, 1972.
6. A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The design and analysis of computer algorithms*. Addison-Wesley Publishing Co., Reading, Mass.-London-Amsterdam, 1975.
7. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows*. Prentice Hall Inc., Englewood Cliffs, NJ, 1993. Theory, algorithms, and applications.
8. M. Aigner and G. Ziegler. *Proofs from the book*. Springer Verlag, Berlin Heidelberg New York, 1998.
9. A. Ainouche. An improvement of Fraisse's sufficient condition for hamiltonian graphs. *J. Graph Theory*, 16:529–543, 1992.
10. N. Alon. Disjoint directed cycles. *J. Combin. Theory Ser. B*, 68(2):167–178, 1996.
11. N. Alon and G. Gutin. Properly colored Hamilton cycles in edge colored complete graphs. *Random Structures and Algorithms*, 11:179–186, 1997.
12. N. Alon and N. Linial. Cycles of length 0 modulo k in directed graphs. *J. Combin. Theory Ser. B*, 47(1):114–119, 1989.
13. N. Alon, C. McDiarmid, and M. Molloy. Edge-disjoint cycles in regular directed graphs. *J. Graph Theory*, 22(3):231–237, 1996.
14. N. Alon and J.H. Spencer. *The probabilistic method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York, 1992. With an appendix by Paul Erdős, A Wiley-Interscience Publication.
15. N. Alon and M. Tarsi. Colourings and orientations of graphs. *Combinatorica*, 12:125–134, 1992.
16. N. Alon, R. Yuster, and U. Zwick. Color-coding: a new method for finding simple paths, cycles and other small subgraphs within large graphs. In *Proc. 26th Annual ACM Symp. Theory Computing*, pages 326–335, Montreal, Canada, 1994. ACM Press.
17. N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM*, 42:844–856, 1995.
18. N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17:209–223, 1997.

19. B. Alspach. Cycles of each length in regular tournaments. *Canad. Math. Bull.*, 10:283–285, 1967.
20. B. Alspach, J.-C. Bermond, and D. Sotteau. Decomposition into cycles. I. Hamilton decompositions. In *Cycles and rays (Montreal, PQ, 1987)*, pages 9–18. Kluwer Acad. Publ., Dordrecht, 1990.
21. B. Alspach and M. Rosenfeld. Realization of certain generalized paths in tournaments. *Discrete Math.*, 34:199–202, 1981.
22. B. Alspach and C. Tabib. A note on the number of 4-circuits in a tournament. In *Theory and practice of combinatorics*, volume 60 of *North-Holland Math. Stud.*, pages 13–19. North-Holland, Amsterdam, 1982.
23. H. Alt, N. Blum, K. Mehlhorn, and M. Paul. Computing a maximum cardinality matching in a bipartite graph in time $O(n^{1.5}\sqrt{m/\log n})$. *Inform. Process. Lett.*, 37(4):237–240, 1991.
24. D. Amar and A. Raspaud. Covering the vertices of a digraph by cycles of prescribed length. *Discrete Math.*, 87:111–118, 1991.
25. A. Andersson. Sublogarithmic Searching without Multiplications. In *Proceedings of the 36th Symposium on Foundations of Computer Science*, pages 655–663. IEEE Computer Society Press, 1995.
26. A. Apartsin, Ferapontova E., and V. Gurvich. A circular graph - counterexample to the Duchet kernel conjecture. *Discrete Math.*, 178:229–231, 1998.
27. K. Appel and W. Haken. Every planar map is four colorable. *Bull. Amer. Math. Soc.*, 82(5):711–712, 1976.
28. E.M. Arkin and C.H. Papadimitriou. On negative cycles in mixed graphs. *Operations Research Letters*, 4:113–116, 1985.
29. E.M. Arkin, C.H. Papadimitriou, and M. Yannakakis. Modularity of cycles and paths in graphs. *J. Assoc. Comput. Mach.*, 31:255–274, 1991.
30. E.H.L. Arts and J.H.M. Korst. *Simulated Annealing and Boltzmann Machines.* Wiley, Chicester, 1989.
31. A.A. Assad. Multicommodity network flows–A survey. *Networks*, 8:37–91, 1978.
32. R. Assous. Enchaînabilité et seuil de monomorphie des tournois n-aires. *Discrete Math.*, 62(2):119–125, 1986.
33. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation.* Springer-Verlag, Berlin, 1999.
34. J. N. Ayoub and I.T. Frisch. Optimally invulnerable directed communication networks. *IEEE Trans. Comm. Technol.*, 18:484–489, 1970.
35. L. Babel and G.J. Woeginger. Pseudo-Hamiltonian graphs. In *Graph-theoretic concepts in computer science (Berlin, 1997)*, pages 38–51. Springer, Berlin, 1997.
36. L. Baffi and R. Petreschi. Parallel maximal matching on minimal vertex series parallel digraphs. In *Algorithms, concurrency and knowledge (Pathumthani, 1995)*, pages 34–47. Springer, Berlin, 1995.
37. E. Balas and N. Simonetti. Linear time dynamic programming algorithms for some new classes of restricted travelling salesman problems. In *Proceedings of IPCO V*, volume 1084 of *Lecture Notes in Comput. Sci.*, pages 316–329. Springer Verlag, Berlin, 1996.
38. M.C. Balbuena, A. Carmona, J. Fàbrega, and M.A. Fiol. Connectivity of large bipartite digraphs and graphs. *Discrete Math.*, 174(1-3):3–17, 1997. Combinatorics (Rome and Montesilvano, 1994).
39. Y. Balcer and A.F. Veinott. Computing a graph's period quadratically by node condensation. *Discrete Math.*, 4:295–303, 1973.
40. M. Balinski and G. Ratier. On stable marriages and graphs, and strategy and polytopes. *SIAM Rev.*, 39(4):575–604, 1997.

41. M. Balinski and G. Ratier. Graphs and marriages. *Amer. Math. Monthly*, 105(5):430–445, 1998.
42. E. Bampis, P. Hell, Y. Manoussakis, and M. Rosenfeld. Finding an antidirected hamiltonian path starting with a forward arc from a given vertex in a tournament. *Lect. Notes Comp. Sci.*, 1120:67–73, 1996.
43. J. Bang-Jensen. On the 2-linkage problem for semicomplete digraphs. In *Graph theory in memory of G. A. Dirac (Sandbjerg, 1985)*, volume 41 of *Ann. Discrete Math.*, pages 23–37. North-Holland, Amsterdam, 1989.
44. J. Bang-Jensen. Locally semicomplete digraphs: a generalization of tournaments. *J. Graph Theory*, 14(3):371–390, 1990.
45. J. Bang-Jensen. A note on a special case of the 2-path problem for semicomplete digraphs. In *Graph theory, combinatorics, and applications, Vol. 1 (Kalamazoo, MI, 1988)*, Wiley-Intersci. Publ., pages 77–86. Wiley, New York, 1991.
46. J. Bang-Jensen. Edge-disjoint in- and out-branchings in tournaments and related path problems. *J. Combin. Theory Ser. B*, 51(1):1–23, 1991.
47. J. Bang-Jensen. On the structure of locally semicomplete digraphs. *Discrete Math.*, 100(1-3):243–265, 1992. Special volume to mark the centennial of Julius Petersen's "Die Theorie der regulären Graphs", Part I.
48. J. Bang-Jensen. Arc-local tournament digraphs: a generalization of tournaments and bipartite tournaments. Technical report 2, Department of Mathematics and Computer Science, Odense University, Denmark, 1993.
49. J. Bang-Jensen. A reformulation of Huang's structure theorem for local tournaments with some consequences. Technical report 13, Department of Mathematics and Computer Science, Odense University, Denmark, 1994.
50. J. Bang-Jensen. Digraphs with the path-merging property. *J. Graph Theory*, 20(2):255–265, 1995.
51. J. Bang-Jensen. Disjoint Paths with Prescribed Ends and Cycles through Given Arcs in Locally Semicomplete Digraphs and Quasi-Transitive Digraphs. Technical Report 22, Dept. of Math. and Comp. Sci., Odense University, Technical report, 1996.
52. J. Bang-Jensen. Linkages in locally semicomplete digraphs and quasi-transitive digraphs. *Discrete Math.*, 196(1-3):13–27, 1999.
53. J. Bang-Jensen, A. Frank, and B. Jackson. Preserving and increasing local edge-connectivity in mixed graphs. *SIAM J. Discrete Math.*, 8:155–178, 1995.
54. J. Bang-Jensen and Y. Guo. A note on vertex pancyclic oriented graphs. *J. Graph Theory*, 31:313–318, 1999.
55. J. Bang-Jensen, Y. Guo, G. Gutin, and L. Volkmann. A classification of locally semicomplete digraphs. *Discrete Math.*, 167/168:101–114, 1997. 15th British Combinatorial Conference (Stirling, 1995).
56. J. Bang-Jensen, Y. Guo, and L. Volkmann. Weakly Hamiltonian-connected locally semicomplete digraphs. *J. Graph Theory*, 21(2):163–172, 1996.
57. J. Bang-Jensen, Y. Guo, and A. Yeo. A new sufficient condition for a digraph to be Hamiltonian. *Discrete Applied Math.*, 95:61–72, 1999.
58. J. Bang-Jensen, Y. Guo, and A. Yeo. Complementary cycles containing prescribed vertices in tournaments. *Discrete Math.*, 214:77–87, 2000.
59. J. Bang-Jensen and G. Gutin. Longest paths and cycles in extended locally semicomplete digraphs. Technical Report 53, Department of Mathematics and Computer Science, Odense University, Denmark, 1993.
60. J. Bang-Jensen and G. Gutin. Paths, trees and cycles in tournaments. *Congr. Numer.*, 115:131–170, 1996. Surveys in graph theory (San Francisco, CA, 1995).
61. J. Bang-Jensen and G. Gutin. Alternating paths and cycles in edge-coloured multigraphs: a survey. *Discrete Math.*, 165-166:39–60, 1997.

62. J. Bang-Jensen and G. Gutin. Paths and cycles in extended and decomposable digraphs. *Discrete Math.*, 164(1-3):5–19, 1997. The Second Krakow Conference on Graph Theory (Zgorzelisko, 1994).
63. J. Bang-Jensen and G. Gutin. Vertex heaviest paths and cycles in quasi-transitive digraphs. *Discrete Math.*, 163(1-3):217–223, 1997.
64. J. Bang-Jensen and G. Gutin. Alternating cycles and trails in 2-edge-coloured complete multigraphs. *Discrete Math.*, 188:61–72, 1998.
65. J. Bang-Jensen and G. Gutin. Generalizations of tournaments: A survey. *Journal of Graph Theory*, 28:171–202, 1998.
66. J. Bang-Jensen and G. Gutin. On the complexity of hamiltonian path and cycle problems in certain classes of digraphs. *Discrete Applied Math.*, 95:41–60, 1999.
67. J. Bang-Jensen, G. Gutin, and J. Huang. Weakly Hamiltonian-connected ordinary multipartite tournaments. *Discrete Math.*, 138(1-3):63–74, 1995. 14th British Combinatorial Conference (Keele, 1993).
68. J. Bang-Jensen, G. Gutin, and J. Huang. A sufficient condition for a semicomplete multipartite digraph to be Hamiltonian. *Discrete Math.*, 161(1-3):1–12, 1996.
69. J. Bang-Jensen, G. Gutin, and H. Li. Sufficient conditions for a digraph to be Hamiltonian. *J. Graph Theory*, 22(2):181–187, 1996.
70. J. Bang-Jensen, G. Gutin, and A. Yeo. On k-strong and k-cyclic digraphs. *Discrete Math.*, 162(1-3):1–11, 1996.
71. J. Bang-Jensen, G. Gutin, and A. Yeo. Hamiltonian cycles avoiding prescribed arcs in tournaments. *Combin. Probab. Comput.*, 6(3):255–261, 1997.
72. J. Bang-Jensen, G. Gutin, and A. Yeo. A polynomial algorithm for the Hamiltonian cycle problem in semicomplete multipartite digraphs. *J. Graph Theory*, 29:111–132, 1998.
73. J. Bang-Jensen, G. Gutin, and A. Yeo. Properly coloured Hamiltonian paths in edge-coloured complete graphs. *Discrete Applied Math.*, 83:267–270, 1998.
74. J. Bang-Jensen and P. Hell. The effect of two cycles on the complexity of colourings by directed graphs. *Discrete Appl. Math.*, 26(1):1–23, 1990.
75. J. Bang-Jensen and P. Hell. Fast algorithms for finding Hamiltonian paths and cycles in in-tournament digraphs. *Discrete Appl. Math.*, 41(1):75–79, 1993.
76. J. Bang-Jensen, P. Hell, and J. Huang. Optimal recognition of local tournaments. *Congressus Numerantium*, 100:141–146, 1994.
77. J. Bang-Jensen, P. Hell, and G. MacGillivray. The complexity of colouring by semicomplete digraphs. *SIAM J. Discrete Math.*, 1(3):281–298, 1988.
78. J. Bang-Jensen, P. Hell, and G. MacGillivray. On the complexity of colouring by superdigraphs of bipartite graphs. *Discrete Math.*, 109(1-3):27–44, 1992. Algebraic graph theory (Leibnitz, 1989).
79. J. Bang-Jensen and J. Huang. Quasi-transitive digraphs. *J. Graph Theory*, 20(2):141–161, 1995.
80. J. Bang-Jensen and J. Huang. Kings in quasi-transitive digraphs. *Discrete Math.*, 185(1-3):19–27, 1998.
81. J. Bang-Jensen, J. Huang, and E. Prisner. In-tournament digraphs. *J. Combin. Theory Ser. B*, 59(2):267–287, 1993.
82. J. Bang-Jensen, J. Huang, and A. Yeo. Strongly connected spanning subgraphs with the minimum number of arcs in quasi-transitive digraphs. Technical Report 5, Department of Mathematics and Computer Science, Odense University, Denmark, 1999.
83. J. Bang-Jensen and T. Jordán. Spanning 2-strong subtournaments in 3-strong semicomplete digraphs . Unpublished manuscript, November 1995. Department of Mathematics and Computer Science, Odense University, Denmark.

84. J. Bang-Jensen and T. Jordán. Adding and reversing arcs in semicomplete digraphs. *Combin. Probab. Comput.*, 7(1):17–25, 1998.
85. J. Bang-Jensen and G. MacGillivray. Further effects of two directed cycles on the complexity of H-colouring. *J. Combin. Math. Combin. Comput.*, 10:33–50, 1991.
86. J. Bang-Jensen and Y. Manoussakis. Weakly Hamiltonian-connected vertices in bipartite tournaments. *J. Combin. Theory Ser. B*, 63(2):261–280, 1995.
87. J. Bang-Jensen, Y. Manoussakis, and C. Thomassen. A polynomial algorithm for Hamiltonian-connectedness in semicomplete digraphs. *J. Algorithms*, 13(1):114–127, 1992.
88. J Bang-Jensen and S. Poljak. Eulerian trails through a set of terminals in specific, unique and all orders. *Contemporary Mathematics*, 147:247–258, 1993.
89. J. Bang-Jensen and C. Thomassen. A polynomial algorithm for the 2-path problem for semicomplete digraphs. *SIAM J. Discrete Math.*, 5:366–376, 1992.
90. J. Bang-Jensen and A. Yeo. Strongly connected spanning subgraphs with the minimum number of arcs in semicomplete multipartite digraphs. Technical report 4, Department of Mathematics and Computer Science, Odense University, Denmark, 1999.
91. M. Bankfalvi and Zs. Bankfalvi. Alternating hamiltonian circuit in two-coloured complete graphs. In *Proceedings of Colloq. Tihany 1968*, pages 11–18. Academic Press, New York, 1968.
92. E. Barbut and A. Bialostocki. A generalization of rotational tournaments. *Discrete Math.*, 76(2):81–87, 1989.
93. E. Barbut and A. Bialostocki. On regular r-tournaments. *Ars Combin.*, 34:97–106, 1992.
94. O. Barr. Properly coloured Hamiltonian paths in edge-coloured complete graphs without monochromatic triangles. *ARS Combinatoria*, 50:316–318, 1998.
95. J.-P. Barthélémy, O. Hudry, G. Isaak, F.S. Roberts, and B. Tesman. The reversing number of a digraph. *Discrete Appl. Math.*, 60(1-3):39–76, 1995. ARIDAM VI and VII (New Brunswick, NJ, 1991/1992).
96. E.T. Baskoro, M. Miller, J. Plesník, and S. Znám. Digraphs of degree 3 and order close to the Moore bound. *J. Graph Theory*, 20:339–349, 1995.
97. G. Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph Drawing*. Prentice Hall, Englewood Cliffs, NJ, 1999.
98. J. Beck. On 3-chromatic hypergraphs. *Discrete Math.*, 24(2):127–137, 1978.
99. L.W. Beineke and C.H.C. Little. Cycles in bipartite tournaments. *J. Combin. Theory Ser. B*, 32(2):140–145, 1982.
100. L.W. Beineke and M.D. Plummer. On the 1-factors of a nonseparable graph. *J. Combin. Theory Ser. B*, 2:285–289, 1967.
101. L.W. Beineke and C.M. Zamfirescu. Connection digraphs and second order line graphs. *Discrete Math.*, 39:237–254, 1982.
102. R.E. Bellman. On a routing problem. *Quart. Appl. Math.*, 16:87–90, 1958.
103. A. Benkouar, Y. Manoussakis, V. Paschos, and R. Saad. On the complexity of finding alternating Hamiltonian and Eulerian cycles in edge-coloured graphs. In , volume 557 of *Lecture Notes in Comput. Sci.*, pages 190–198. Springer Verlag, Berlin, 1991.
104. M.D. Bennett. Nucleotypic basis of the spacial ordering of chromosomes in eucariotes and the implications of the order for genome and phenotypic variation. In *Genome Evolution*, pages 239–261. Academic Press, London, 1982.
105. C. Berge. *Graphs and Hypergraphs.* North-Holland Publishing Co., Amsterdam, 1976. Second edition.
106. C. Berge. Diperfect graphs. *Combinatorica*, 2(3):213–222, 1982.

107. C. Berge. Path partitions in directed graphs. In *Combinatorial Mathematics (Marseille-Luminy, 1981)*, pages 59–63. North-Holland, Amsterdam, 1983.
108. C. Berge. *Graphs*. North-Holland Publishing Co., Amsterdam, 1985. Second revised edition of part 1 of the 1973 English version.
109. C. Berge and Rao A.R. A combinatorial problem in logic. *Discrete Math.*, 17:23–26, 1977.
110. C. Berge and P. Duchet. Recent problems and results about kernels in directed graphs. *Discrete Math.*, 86(1-3):27–31, 1990.
111. K.A. Berman and X. Liu. Cycles through large degree vertices in digraphs: a generalization of Meyniel's theorem. *J. Combin. Theory, Ser. B*, 74:20–27, 1998.
112. J.-C. Bermond, A. Germa, M.-C. Heydemann, and D. Sotteau. Girth in digraphs. *J. Graph Theory*, 4(3):337–341, 1980.
113. J.-C. Bermond and P. Hell. On even factorizations and the chromatic index of the Kautz and de Bruijn digraphs. *J. Graph Theory*, 17:647–655, 1993.
114. J.-C. Bermond and C. Peyrat. De Bruijn and Kautz networks: A competitor for the hypercube? In *Hypercube and distributed computers, André, F. and Verjus, J. P. (eds.)*, pages 279–493. Elsevier, North-Holland, 1989.
115. J.-C. Bermond and C. Thomassen. Cycles in digraphs—a survey. *J. Graph Theory*, 5(1):1–43, 1981.
116. P. Bertolazzi, R.F. Cohen, G. Di Battista, R. Tamassia, and I.G. Tollis. How to draw a series-parallel digraph. *Internat. J. Comput. Geom. Appl.*, 4(4):385–402, 1994.
117. A. Bialostocki. An application of the Ramsey theorem to ordered r-tournaments. *Discrete Math.*, 61(2-3):325–328, 1986.
118. W. Bienia, L. Goddyn, P. Gvozdjak, A. Sebö, and M. Tarsi. Flows, view obstructions, and the lonely runner. *J. Combin. Theory Ser. B*, 72(1):1–9, 1998.
119. N.L. Biggs, E.K. Lloyd, and R.J. Wilson. *Graph Theory 1736-1936*. Clarendon Press, 1976.
120. F. Boesch and R. Tindell. Robbins's theorem for mixed multigraphs. *Amer. Math. Monthly*, 87(9):716–719, 1980.
121. B. Bollobás and P. Erdős. Alternating Hamiltonian cycles. *Israel J. Math.*, 23:126–131, 1976.
122. B. Bollobás, D.L. Goldsmith, and D.R. Woodall. Indestructive deletions of edges from graphs. *J. Combin. Theory Ser. B*, 30(3):263–275, 1981.
123. B. Bollobás and R. Häggkvist. Powers of Hamilton cycles in tournaments. *J. Combin. Theory Ser. B*, 50(2):309–318, 1990.
124. J.A. Bondy. Diconnected orientations and a conjecture of Las Vergnas. *J. London Math. Soc. (2)*, 14(2):277–282, 1976.
125. J.A. Bondy. A short proof of the Chen-Manalastas theorem. *Discrete Math.*, 146(1-3):289–292, 1995.
126. J.A. Bondy. Basic graph theory: paths and circuits. In *Handbook of combinatorics, Vol. 1, 2*, pages 3–110. Elsevier, Amsterdam, 1995.
127. J.A. Bondy and U.S.R. Murty. *Graph theory with applications*. American Elsevier Publishing Co., Inc., New York, 1976.
128. J.A. Bondy and C. Thomassen. A short proof of Meyniel's theorem. *Discrete Math.*, 19(2):195–197, 1977.
129. J.A. Bondy and A. Vince. Cycles in a graph whose lengths differ by one or two. *J. Graph Theory*, 27(1):11–15, 1998.
130. O.V. Borodin, A.V. Kostochka, and D.R. Woodall. On kernel-perfect orientations of line graphs. *Discrete Math.*, 191(1-3):45–49, 1998. Graph theory (Elgersburg, 1996).

131. E. Boros and V. Gurvich. Perfect graphs are kernel-solvable. *Discrete Math.*, 159:35–55, 1996.
132. E. Boros and V. Gurvich. A corrected version of the Duchet kernel conjecture. *Discrete Math.*, 179(1-3):231–233, 1998.
133. A. Brandstädt. *Graphen und Algorithmen.* B. G. Teubner, Stuttgart, 1994.
134. G. Brassard and P. Bratley. *Fundamentals of algorithmics.* Prentice Hall Inc., Englewood Cliffs, NJ, 1996.
135. R.C. Brewster and G. MacGillivray. The homomorphism factoring problem. *J. Combin. Math. Combin. Comput.*, 25:33–53, 1997.
136. W.G. Bridges and S. Toueg. On the impossibility of directed Moore graphs. *J. Combin. Theory, Ser. B*, 29:339–341, 1980.
137. R.E. Burkard, V.G. Deineko, and Woeginger G.J. The travelling salesman problem and the PQ-tree. In *Proceedings of IPCO V*, volume 1084 of *Lecture Notes in Comput. Sci.*, pages 490–504. Springer Verlag, Berlin, 1996.
138. R.G. Busacker and P.J. Gowen. A procedure for determining a family of minimal cost network flow patterns. Technical Report 15, ORO Tech. report, John Hopkins University, 1961.
139. L. Caccetta and R. Häggkvist. On minimal digraphs with given girth. *Congressus Numerantium*, 21:181–187, 1978.
140. P. Camion. Chemins et circuits hamiltoniens des graphes complets. *C. R. Acad. Sci. Paris*, 249:2151–2152, 1959.
141. Papadimitriou C.H. and M. Yannakakis. On limited nondeterminism and the complexity of the V-C dimension. In *Proc. 8th Annual Symp. Structure in Complexity Theory*, pages 12–18, San Diego, Ca., 1993.
142. G.J. Chang, F.K. Hwang, and L.D. Tong. The Hamiltonian property of the consecutive-3 digraphs. *Math. Comp. Mod.*, 25:83–88, 1997.
143. G.J. Chang, F.K. Hwang, and L.D. Tong. The consecutive-4 digraph are Hamiltonian. *J. Graph Theory*, 31:1–6, 1999.
144. G. Chartrand, D. Geller, and S. Hedetniemi. Graphs with forbidden subgraphs. *J. Comb. Theory Ser. B*, 10:12–41, 1971.
145. C.C. Chen and D.E. Daykin. Graphs with Hamiltonian cycles having adjacent lines of different colors. *J. Combin. Theory, Ser. B*, 21:135–139, 1976.
146. C.C. Chen and Jr. Manalastas, P. Every finite strongly connected digraph of stability 2 has a Hamiltonian path. *Discrete Math.*, 44(3):243–250, 1983.
147. G.-T. Chen, R.J. Gould, and H. Li. Partitioning Vertices of a Tournament into Independent Cycles. Technical report 1158, LRI, Universite Paris-sud, 1998.
148. E. Cheng and T. Jordán. Successive edge-connecitity augmentation problems. *Mathematical Programming, Series B*, 84:577–593, 1999.
149. J. Cheriyan and S.N. Maheshwari. Analysis of preflow push algorithms for maximum network flow. *SIAM J. Computing*, 18:1057–1086, 1989.
150. J. Cheriyan and J.H. Reif. Directed s-t numberings, rubber bands, and testing digraph k-vertex connectivity. *Combinatorica*, 14(4):435–451, 1994.
151. J. Cheriyan and R. Thurimella. Approximating minimum-size k-connected spanning subgraphs via matching (extended abstract). In *37th Annual Symposium on Foundations of Computer Science (Burlington, VT, 1996)*, pages 292–301. IEEE Comput. Soc. Press, Los Alamitos, CA, 1996.
152. J. Cheriyan and R. Thurimella. Approximating minimum size k-connected spanning subgraphs via matching. *Electronic Colloquium on Computational Complexity*, Report no. 25:36 pages, 1998.
153. B.V. Cherkassky and A.V. Goldberg. Negative-cycle detection algorithms. *Mathematical Programming*, 85:277–311, 1999.

154. B.V. Cherkassky, A.V. Goldberg, and T. Radzik. Shortest paths algorithms: theory and experimental evaluation. *Mathematical Programming*, 73:129–174, 1996.
155. A.G. Chetwynd and A.J.W. Hilton. Alternating Hamiltonian cycles in two colored complete bipartite graphs. *J. Graph Theory*, 16:153–158, 1992.
156. W.S. Chow, Y. Manoussakis, O. Megalakaki, M. Spyratos, and Zs. Tuza. Paths through fixed vertices in edge-colored graphs. *Math. Informatique et Sciences Humaines*, 127(32):49–58, 1994.
157. F.R.K. Chung, M.R. Garey, and R.E. Tarjan. Strongly connected orientations of mixed multigraphs. *Networks*, 15(4):477–484, 1985.
158. F.R.K. Chung, W. Goddard, and D.J. Kleitman. Even cycles in directed graphs. *SIAM J. Discrete Math.*, 7(3):474–483, 1994.
159. V. Chvátal. On Hamilton's ideals. *J. Combin. Theory, Ser. B*, 12:163–168, 1972.
160. V. Chvátal. *Linear programming.* W. H. Freeman and Company, 1983.
161. V. Chvátal and P. Erdős. A note on Hamiltonian circuits. *Discrete Math.*, 2:111–113, 1972.
162. V. Chvátal and L. Lovász. Every directed graph has a semi-kernel. *Lecture Notes in Mathematics*, 411:175, 1974.
163. V. Chvátal and E. Szemerédi. Short cycles in directed graphs. *J. Combin. Theory Ser. B*, 35(3):323–327, 1983.
164. V. Chvátal and C. Thomassen. Distances in orientations of graphs. *J. Combin. Theory Ser. B*, 24(1):61–75, 1978.
165. S.A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Ann. ACM Symp. on Theory of Computing*, pages 151–158, 1971.
166. W.J. Cook, W.H. Cunninghan, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization.* John Wiley & Sons, New York, 1998.
167. C. Cooper, A. Frieze, and M. Molloy. Hamilton cycles in random regular digraphs. *Combin. Probab. Comput.*, 3(1):39–49, 1994.
168. D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. In *Proceedings of the 19th Ann. ACM Symp. on Theory of Computation*, pages 1–6, ACM Press, 1987.
169. T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to algorithms.* The MIT Electrical Engineering and Computer Science Series. MIT Press, Cambridge, MA, 1990.
170. W.H. Cunningham and A. Frank. A primal-dual algorithm for submodular flows. *Math. Oper. Res.*, 10(2):251–262, 1985.
171. G. Dahl. Directed Steiner problems with connectivity constraints. *Discrete Appl. Math.*, 47:109–128, 1193.
172. M. Dalmazzo. Nombre d'arcs dans les graphes k-arc-fortement connexes minimaux. *C.R.Acad. Sci. Paris A*, 2853:341–344, 1977.
173. P. Dankelmann and L. Volkmann. Degree sequence conditions for maximally edge-connected graphs and digraphs. *J. Graph Theory*, 26(1):27–34, 1997.
174. S.K. Darbinyan. Cycles of any length in digraphs with large semidegrees. *Akad. Nauk Armyan. SSR Dokl.*, 75(4):147–152, 1982.
175. S.K. Darbinyan. Pancyclicity of digraphs with large semidegrees. *Akad. Nauk Armyan. SSR Dokl.*, 80(2):51–54, 1985.
176. S.K. Darbinyan. Pancyclicity of digraphs with the Meyniel condition. *Studia Sci. Math. Hungar.*, 20(1-4):95–117, 1985.
177. S.K. Darbinyan. A sufficient condition for the Hamiltonian property of digraphs with large semidegrees. *Akad. Nauk Armyan. SSR Dokl.*, 82(1):6–8, 1986.

178. S.K. Darbinyan. On the pancyclicity of digraphs with large semidegrees. *Akad. Nauk Armyan. SSR Dokl.*, 83(3):99–101, 1986.
179. S.K. Darbinyan. Hamiltonian and strongly Hamilton-connected digraphs. *Akad. Nauk Armyan. SSR Dokl.*, 91(1):3–6, 1990.
180. S.K. Darbinyan. On hamiltonian bypasses in digraphs satisfying Meyniel-like conditions (in Russian). *Math. Problems in Computer Science*, 20:7–19, 1998.
181. M. Darrah, Y.-P. Liu, and C.-Q. Zhang. Cycles of all lengths in arc-3-cyclic semicomplete digraphs. *Discrete Math.*, 173(1-3):23–33, 1997.
182. P. Das. Pan-alternating cyclic edge-partitioned graphs. *ARS Combinatoria*, 14:105–114, 1982.
183. H.A. David. *The method of paired comparisons.* Oxford Univ. Press, London, second edition, 1988.
184. D.E. Daykin. Graphs with cycles having adjacent lines of different colors. *J. Combin. Theory, Ser. B*, 20:149–152, 1976.
185. N.G. de Bruijn. A combinatorial problem. *Nederl. Akad. Wetensh. Proc.*, 49:758–764, 1946.
186. W.F. de la Vega. On the maximum cardinality of a consistent set of arcs in a random tournament. *J. Comb. Theory Ser. B*, 35:328–332, 1983.
187. N. Dean and B.J. Latka. Squaring the tournament—an open problem. *Congressus Numerantium*, 109:73–80, 1995.
188. V.G. Deineko and Woeginger G.J. A study of exponential neighbourhoods for the traveling salesman problem and the quadratic assignment problem. Technical Report Woe05, Tech. University Graz, Austria, 1997.
189. C. Delorme, O. Ordaz, and D. Quiroz. Tools for studying paths and cycles in digraphs. *Networks*, 31:125–148, 1998.
190. X. Deng, P. Hell, and J. Huang. Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. *SIAM J. Computing*, 25(2):390–403, 1996.
191. R. Diestel. *Graph theory.* Springer-Verlag, New York, 2000. 2nd edition, first edition 1997.
192. E.W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271, 1959.
193. R.P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51:161–166, 1950.
194. G. Ding, A. Schrijver, and P.D. Seymour. Disjoint paths in a planar graph – a general theorem. *SIAM J. Discrete Math.*, 5(1):112–116, 1992.
195. E.A. Dinic. An algorithm for the solution of the problem of maximal flow in a network with power estimation. *Dokl. Akad. Nauk SSSR*, 194:754–757, 1970.
196. E.A. Dinits and A.V. Karzanov. On the existence of two edge-disjoint chains in multi-graph connecting given pairs of its vertices. *Graph Theory Newsletters*, 8:2–3, 1979.
197. E.A. Dinits and A.V. Karzanov. On two integer flows of value 1. In *Combinatorial methods for network flow problems, A.V. Karzanov, ed.*, pages 127–137. Institute for System Studies, Moscow, 1979.
198. G.A. Dirac. Some theorems on abstract graphs. *Proc. London Math. Soc.*, 2(3):69–81, 1952.
199. A. Dolan and J. Aldous. *Networks and algorithms.* John Wiley & Sons Ltd., Chichester, 1993. An introductory approach.
200. D. Dorninger. On permutations of chromosomes. In *Contributions to General Algebra*, volume 5, pages 95–103. Teubner-Verlag, Stuttgart, 1987.
201. D. Dorninger. Hamiltonian circuits determining the order of chromosomes. *Discrete Applied Math.*, 50:159–168, 1994.

202. D. Dorninger and W. Timischl. Geometrical constraints on Bennett's predictions of chromosome order. *Heredity*, 58:321–325, 1987.
203. K. Dowsland. Simulated annealing. In *Modern heuristic techniques for combinatorial problems, C.R.Reeves ed.*, pages 20–69. McGraw-Hill Book Company, 1995.
204. D.-Z. Du, F. Cao, and D.F. Hsu. De Bruijn digraphs, and Kautz digraphs, and their generalizations. In *Combinatorial network theory, Du, D.-Z. and Hsu, D.F. (eds)*, pages 65–105. Kluwer Acad. Publ., Dordrecht, 1996.
205. D.-Z. Du and D.F. Hsu. On Hamiltonian consecutive-d digraphs. *Banach Center Publ.*, 25:47–55, 1989.
206. D.-Z. Du, D.F. Hsu, and F.K. Hwang. Hamiltonian property of d-consecutive digraphs. *Math. and Comput. Modeling*, 17:61–63, 1993.
207. D.Z. Du, Y.-D. Lyuu, and D.F. Hsu. Line digraph iterations and the spread concept—with application to graph theory, fault tolerance, and routing. In *Graph-theoretic concepts in computer science (Fischbachau, 1991)*, pages 169–179. Springer, Berlin, 1992.
208. I.S. Duff, A.M. Erisman, and J.K. Reid. *Direct methods for sparse matrices.* Oxford University Press, 1997.
209. R.J. Duffin. Topology of series-parallel networks. *J. Mathematical Analysis and Applications*, 10:303–318, 1965.
210. J. Edmonds. Paths, trees, and flowers. *Canad. J. Math.*, 17:449–467, 1965.
211. J. Edmonds. Optimum branchings. *J. Res. Nat. Bur. Standards Sect. B*, 71B:233–240, 1967.
212. J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and their Applications (Proc. Calgary Internat. Conf., Calgary, Alta., 1969)*, pages 69–87. Gordon and Breach, New York, 1970.
213. J. Edmonds. Matroids and the greedy algorithm. *Math. Programming*, 1:127–136, 1971.
214. J. Edmonds. Edge-disjoint branchings. In *Combinatorial Algorithms (B. Rustin, ed.)* , pages 91–96. Academic Press, New York, 1973.
215. J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. In *Studies in integer programming (Proc. Workshop, Bonn, 1975)*, pages 185–204. Ann. of Discrete Math., Vol. 1. North-Holland, Amsterdam, 1977.
216. J. Edmonds and R.M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19:248–264, 1972.
217. A. Ehrenfeucht, H.N. Gabow, R.M. McConnell, and S.J. Sullivan. An $O(n^2)$ divide-and-conquer algorithm for the prime tree decomposition of two-structures and modular decomposition of graphs. *J. Algorithms*, 16(2):283–294, 1994.
218. S. Enni. A note on mixed graphs and directed splitting off. *J. Graph Theory*, 27(4):213–221, 1998.
219. S. Enni. A 1-ST-edge-connectivity augmentation algorithm. *Mathematical Programming Ser. B*, 84:529–535, 1999.
220. P. Erdős. Graph theory and probability. *Canad. J. Math.*, 11:34–38, 1959.
221. P. Erdős. Some old and new problems in various branches of combinatorics. *Congressus Numerantium*, 23:19–37, 1979.
222. P. Erdős, A.L. Rubin, and R.W. Irwing. Choosability in graphs. *Congressus Numerantium*, 26:122–157, 1980.
223. P. Erdős and W.T. Trotter. When the Cartesian product of directed cycles is Hamiltonian. *J. Graph Theory*, 2:137–142, 1978.
224. A.H. Esfahanian and S.L. Hakimi. On computing the connectivities of graphs and digraphs. *Networks*, 14(2):355–366, 1984.

225. L. Euler. Solutio problematis ad geometriam situs pertinentis. *Comment. Academiae Sci. Petropolitanae*, 8:128–140, 1736.
226. L. Euler. Solutio problematis as geometriam situs pertinentis. *Commentarii Academiae Scientiarum Imperialis Petropolitanae*, 8:128–140, 1736.
227. G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.
228. S. Even. An algorithm for determining whether the connectivity of a graph is at least k. *SIAM J. Computing*, 4(3):393–396, 1975.
229. S. Even. *Graph algorithms.* Computer Science Press Inc., Woodland Hills, Calif., 1979. Computer Software Engineering Series.
230. S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Computing*, 5(4):691–703, 1976.
231. S. Even and O. Kariv. An $O(n^{2.5})$-algorithm for maximum matchings in general graphs. In *Proceedings of the 16th Annual Symp. Foundations of CS, Berkley, 1975*, pages 100–112. IEEE Comput. Soc. Press, Los Alamitos, CA, 1975.
232. S. Even and R.E. Tarjan. Network flow and testing graph connectivity. *SIAM J. Computing*, 4(4):507–518, 1975.
233. J. Feldman and M. Ruhl. The directed Steiner network problem is tractable for a constant number of terminals. In *Proceedings of the 40th annual Symposium on Foundations of Computer Science*, pages 299–308. IEEE Computer Society Press, 1999.
234. W. Feller. *An introduction to probability theory and its applications. Vol. I.* John Wiley & Sons Inc., New York, third edition, 1968.
235. J.F. Fink and L. Lesniak-Foster. Graphs for which every unilateral orientation is traceable. *ARS Combinatoria*, 9:113–118, 1980.
236. M.A. Fiol, J.L.A. Yebra, and I. Alegre. Line digraph iteration and the (d, k) digraph problem. *IEEE Transections on Computers*, C-33:400–403, 1984.
237. D.C. Fisher. Squaring a tournament: a proof of Dean's conjecture. *J. Graph Theory*, 23(1):43–48, 1996.
238. M.J. Fisher and A.R. Meyer. Boolean matrix multiplication and transitive closure. In *Proceedings of the 12th Ann. ACM Symp. on Switching and Automata Theory*, pages 129–131. ACM Press, 1971.
239. H. Fleischner. Eine gemeinsame Basis für die Theorie der Eulerschen Graphen und den Satz von Petersen. *Monatsh. Math.*, 81(4):267–278, 1976.
240. H. Fleischner. *Eulerian graphs and related topics. Part 1. Vol. 1.* North-Holland Publishing Co., Amsterdam, 1990.
241. H. Fleischner. *Eulerian graphs and related topics. Part 1. Vol. 2.* North-Holland Publishing Co., Amsterdam, 1991.
242. H. Fleischner, G. Sabidussi, and E. Wegner. Transforming eulerian trails. *Discrete Math.*, 109:103–116, 1992.
243. R.W. Floyd. Algorithm 97, shortest path. *Comm. ACM*, 5:345, 1962.
244. R. Forcade. Parity of paths and circuits in tournaments. *Discrete Math.*, 6:115–118, 1973.
245. Jr. Ford, L.R. Network flow theory. Technical Report P-923, The Rand Corp., 1956.
246. Jr. Ford, L.R. and D.R. Fulkerson. *Flows in networks.* Princeton University Press, Princeton, N.J., 1962.
247. S. Fortune, J.E. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10:111–121, 1980.
248. P. Fraigniaud and E. Lazard. Methods and problems of communication in usual networks. *Discrete Applied Math.*, 53:79–133, 1994.

249. P. Fraisse and C. Thomassen. Hamiltonian dicycles avoiding prescribed arcs in tournaments. *Graphs Combin.*, 3(3):239–250, 1987.
250. A. Frank. Kernel systems of directed graphs. *Acta Sci. Math. (Szeged)*, 41(1-2):63–76, 1979.
251. A. Frank. On the orientation of graphs. *J. Combin. Theory Ser. B*, 28(3):251–261, 1980.
252. A. Frank. On disjoint trees and arborescences. In *Algebraic methods in graph theory, Vol. I, II (Szeged, 1978)*, pages 159–169. North-Holland, Amsterdam, 1981.
253. A. Frank. A note on k-strongly connected orientations of an undirected graph. *Discrete Math.*, 39(1):103–104, 1982.
254. A. Frank. An algorithm for submodular functions on graphs. In *Bonn Workshop on Combinatorial Optimization (Bonn, 1980)*, volume 16 of *Ann. Discrete Math.*, pages 97–120. North-Holland, Amsterdam, 1982.
255. A. Frank. Finding feasible vectors of Edmonds-Giles polyhedra. *J. Combin. Theory Ser. B*, 36(3):221–239, 1984.
256. A. Frank. Submodular flows. In *Progress in combinatorial optimization (Waterloo, Ont., 1982)*, pages 147–165. Academic Press, Toronto, Ont., 1984.
257. A. Frank. On connectivity properties of Eulerian digraphs. In *Graph theory in memory of G. A. Dirac (Sandbjerg, 1985)*, volume 41 of *Ann. Discrete Math.*, pages 179–194. North-Holland, Amsterdam, 1989.
258. A. Frank. Augmenting graphs to meet edge-connectivity requirements. *SIAM J. Discrete Math.*, 5(1):25–53, 1992.
259. A. Frank. Applications of submodular functions. In *Surveys in combinatorics, 1993 (Keele)*, volume 187 of *London Math. Soc. Lecture Note Ser.*, pages 85–136. Cambridge Univ. Press, Cambridge, 1993.
260. A. Frank. Submodular functions in graph theory. *Discrete Math.*, 111(1-3):231–243, 1993. Graph theory and combinatorics (Marseille-Luminy, 1990).
261. A. Frank. Connectivity augmentation problems in network design. In *Mathematical Programming: State of the art (J. R. Birge, K. G. Murty eds.)*, pages 34–63. The University of Michigan, 1994.
262. A. Frank. Connectivity and network flows. In *Handbook of combinatorics, Vol. 1, 2*, pages 111–177. Elsevier, Amsterdam, 1995.
263. A. Frank. Orientations of graphs and submodular flows. *Congr. Numer.*, 113:111–142, 1996. Festschrift for C. St. J. A. Nash-Williams.
264. A. Frank. Applications of relaxed submodularity. *Documenta Mathematica*, Extra Vol. III:343–354, 1998.
265. A. Frank. Increasing the rooted-connectivity of a digraph by one. *Mathematical Programming, Series B*, 84:565–576, 1999.
266. A. Frank. Finding minimum edge-coverings of pairs of sets. manuscript 1998.
267. A. Frank and A. Gyárfás. Directed graphs and computer programs. In *Problémes Combinatoires et Théorie des Graphes*, Colloque Internationaux C.N.R.S., 260, pages 157–158, 1976.
268. A. Frank and A. Gyárfás. How to orient the edges of a graph? In *Combinatorics (Proc. Fifth Hungarian Colloq., Keszthely, 1976), Vol. I*, volume 18 of *Colloq. Math. Soc. János Bolyai*, pages 353–364. North-Holland, Amsterdam, 1978.
269. A. Frank, T. Ibaraki, and H. Nagamochi. On sparse subgraphs preserving connectivity properties. *J. Graph Theory*, 17(3):275–281, 1993.
270. A. Frank, T. Ibaraki, and H. Nagamochi. Two arc-disjoint paths in Eulerian digraphs. *SIAM J. Discrete Math.*, 11(4):557–589 (electronic), 1998.

271. A. Frank and T. Jordán. How to make a strongly connected digraph two-connected. In *Integer programming and combinatorial optimization (Copenhagen, 1995)*, volume 920 of *Lecture Notes in Comput. Sci.*, pages 414–425. Springer, Berlin, 1995.
272. A. Frank and T. Jordán. Minimal edge-coverings of pairs of sets. *J. Combin. Theory Ser. B*, 65(1):73–110, 1995.
273. A. Frank and T. Jordán. Directed vertex-connectivity augmentation. *Mathematical Programming, Series B*, 84:537–553, 1999.
274. A. Frank and É. Tardos. Generalized polymatroids and submodular flows. *Mathematical Programming Series B*, 42(3 (Ser. B)):489–563, 1988. Submodular optimization.
275. A. Frank and É. Tardos. An application of submodular flows. *Linear Algebra Appl.*, 114/115:329–348, 1989.
276. P. Frankl. What must be contained in every oriented k-uniform hypergraph. *Discrete Math.*, 62(3):311–313, 1986.
277. M.L. Fredman, J. Komlós, and E. Szemerédi. Stroring a sparse table with $O(1)$ worst case access time. *Journal of the ACM*, 31:538–544, 1984.
278. M.L. Fredman and R.E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. Assoc. Comput. Mach.*, 34(3):596–615, 1987.
279. S. Friedland. Every 7-regular digraph contains an even cycle. *J. Combin. Theory Ser. B*, 46(2):249–252, 1989.
280. S. Fujishige. *Submodular functions and optimization.* North-Holland Publishing Co., Amsterdam, 1991.
281. S. Fujishige, H. Röck, and U. Zimmermann. A strongly polynomial algorithm for minimum cost submodular flow problems. *Math. Oper. Res.*, 14(1):60–69, 1989.
282. D.R. Fulkerson. Networks, frames and blocking systems. In *Mathematics of the decision sciences, part I (eds: G.B.Danzig and A.F.Veinott)*, pages 303–334. Amer. Math. Soc., Providence, R.I., 1968.
283. D.R. Fulkerson. Packing rooted directed cuts in a weighted directed graph. *Math. Programming*, 6:1–13, 1974.
284. M. Funke and G. Reinelt. A polyhedral approach to the feedback vertex set problem. In *Integer programming and combinatorial optimization (Vancouver, BC, 1996)*, pages 445–459. Springer, Berlin, 1996.
285. Z. Főredi, P. Horak, C.M. Pareek, and X. Zhu. Minimal oriented graphs of diameter 2. *Graphs and Combinatorics*, 14:345–350, 1998.
286. M.E. Furman. Application of a method of fast multiplication of matrices in the problem of finding the transitive closure of a graph. *Sov. Math. Dokl.*, 11:1252, 1970.
287. H.N. Gabow. A matroid approach to finding edge connectivity and packing arborescences. *J. Comput. System Sci.*, 50(2):259–273, 1995. 23rd Symposium on the Theory of Computing (New Orleans, LA, 1991).
288. H.N. Gabow and T. Jordán. Bipartition constrained edge-splitting in directed graphs. *Discrete Applied Math.*, to appear.
289. D. Gale. A theorem on flows in networks. *Pacific Journal of Mathematics*, 7:1073–1082, 1957.
290. D. Gale and L.S. Shapley. College admissions and the stability of marriage. *Amer. Math. Monthly*, 69:9–15, 1962.
291. H. Galeana-Sánchez. A counterexample to a conjecture of Meyniel on kernel-perfect graphs. *Discrete Math.*, 41:105–107, 1982.
292. H. Galeana-Sánchez. A characterization of normal fraternally orientable perfect graphs. *Discrete Math.*, 169(1-3):221–225, 1997.

293. H. Galeana-Sánchez and X. Li. Semikernels and (k, l)-kernels in digraphs. *SIAM J. Discrete Math.*, 11(2):340–346, 1998.
294. H. Galeana-Sánchez and V. Neuman-Lara. On kernels and semikernels of digraphs. *Discrete Math.*, 48:67–76, 1984.
295. Z. Galil. Finding the vertex connectivity of graphs. *SIAM J. Computing*, 9(1):197–199, 1980.
296. T. Gallai. *Problem 15, in: M. Fiedler ed., Theory of Graphs and its Applications*, page 161. Czech. Akad. Sci. Prague, 1964.
297. T. Gallai. On directed paths and circuits. In *Theory of Graphs (Proc. Colloq., Tihany, 1966)*, pages 115–118. Academic Press, New York, 1968.
298. T. Gallai and A.N. Milgram. Verallgemeinerung eines graphentheoretischen Satzes von Rédei. *Acta Sci. Math. Szeged*, 21:181–186, 1960.
299. A. Galluccio and M. Loebl. Cycles of prescribed modularity in planar digraphs. *J. Algorithms*, 21(1):51–70, 1996.
300. A. Galluccio and M. Loebl. Even directed graphs in H-free digraphs. *J. Algorithms*, 27, 1996.
301. A. Galluccio and M. Loebl. (p, q)-odd digraphs. *J. Graph Theory*, 23(2):175–184, 1996.
302. F. Galvin. The list chromatic index of a bipartite multigraph. *Journal of Combin. Theory. Series B*, 63:153–158, 1995.
303. M.R. Garey and D.S. Johnson. *Computers and intractability.* W. H. Freeman and Co., San Francisco, Calif., 1979.
304. M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoret. Comput. Sci.*, 1(3):237–267, 1976.
305. M.R. Garey, D.S. Johnson, and R.E. Tarjan. The planar hamiltonian circuit problem is NP-complete. *SIAM J. Computing*, 5:704–714, 1976.
306. F. Gavril. Some NP-complete problems on graphs. In *Proc. 11th Conf. on Information Sciences and Systems*, pages 91–95, John Hopkins University, Baltimore, MD, 1977.
307. F. Gavril. Intersection graphs of proper subtrees of unicyclic graphs. *J. Graph Theory*, 18(6):615–627, 1994.
308. F. Gavril, V. Toledano Laredo, and D. de Werra. Chordless paths, odd holes, and kernels in graphs without m-obstructions. *J. Algorithms*, 17(2):207–221, 1994.
309. F. Gavril and J. Urrutia. Intersection graphs of concatenable subtrees of graphs. *Discrete Appl. Math.*, 52(2):195–209, 1994.
310. A.M.H. Gerards. Homomorphisms of graphs into odd cycles. *J. Graph Theory*, 12(1):73–83, 1988.
311. A.M.H. Gerards. *Graphs and polyhedra. Binary spaces and cutting planes.* Stichting Mathematisch Centrum Centrum voor Wiskunde en Informatica, Amsterdam, 1990.
312. A.M.H. Gerards. An orientation theorem for graphs. *J. Combin. Theory Ser. B*, 62(2):199–212, 1994.
313. A.M.H. Gerards and F.B. Shepherd. Strong orientations without even directed circuits. *Discrete Math.*, 188(1-3):111–125, 1998.
314. A. Ghouila-Houri. Diametre maximal d'un graphe fortement connexe. *C.R. Acad. Sci. Paris*, 250:254–256, 1960.
315. A. Ghouila-Houri. Une condition suffisante d'existence d'un circuit hamiltonien. *C.R. Acad. Sci. Paris*, 25:495–497, 1960.
316. A. Ghouila-Houri. Caractérisation des graphes non orientés dont on peut orienter les arëtes de manière à obtenir le graphe d'une relation d'ordre. *C. R. Acad. Sci. Paris*, 254:1370–1371, 1962.

317. P. Gibbons, R. Karp, V. Ramachandran, D. Soroker, and R. Tarjan. Transitive compaction in parallel via branchings. *J. Algorithms*, 12(1):110–125, 1991.
318. F. Glover. Ejection chains, reference structures, and alternating path algorithms for traveling salesman problem. University of Colorado-Boulder, April 1992.
319. F. Glover, G. Gutin, A. Yeo, and A. Zverovich. Construction heuristics and domination analysis for the asymmetric TSP. *European J. Oper. Res.*, to appear.
320. F. Glover and A.P. Punnen. The travelling salesman problem: new solvable cases and linkages with the development of approximation algorithms. *J. Oper. Res. Soc.*, 48:502–510, 1997.
321. W.D. Goddard, G. Kubicki, O.R. Oellermann, and S.L. Tian. On multipartite tournaments. *J. Combin. Theory Ser. B*, 52(2):284–300, 1991.
322. W.D. Goddard and O.R. Oellermann. On the cycle structure of multipartite tournaments. In *Graph Theory Combin. Appl. 1*, pages 525–533. Wiley, New York, 1991.
323. M.X. Goemans and Williamson D. Primal-dual approximation algorithms for feedback problems in planar graphs. *Combinatorica*, 18:37–59, 1998.
324. A.V. Goldberg and R.E. Tarjan. A new approach to the maximum-flow problem. In *Proceedings of the 18th ACM Symposium on the Theory of Computing*, pages 136–146. ACM Press, 1986.
325. A.V. Goldberg and R.E. Tarjan. A new approach to the maximum-flow problem. *J. Assoc. Comput. Mach.*, 35(4):921–940, 1988.
326. A.V. Goldberg and R.E. Tarjan. Finding minimum-cost circulations by canceling negative cycles. *J. Assoc. Comput. Mach.*, 36(4):873–886, 1989.
327. M.K. Goldberg. Some applications of the operation of contraction to strongly connected graphs. *Uspehi Mat. Nauk*, 20(5 (125)):203–205, 1965.
328. M.K. Goldberg. The diameter of a strongly connected graph. *Dokl. Akad. Nauk SSSR*, 170:767–769, 1966.
329. D.L. Goldsmith and A.T. White. On graphs with equal edge-connectivity and minimum degree. *Discrete Math.*, 23(1):31–36, 1978.
330. M.C. Golumbic. The complexity of comparability graph recognition and coloring. *Computing*, 18(3):199–208, 1977.
331. M.C. Golumbic. *Algorithmic graph theory and perfect graphs*. Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1980. With a foreword by Claude Berge, Computer Science and Applied Mathematics.
332. M. Gondran and M. Minoux. *Graphs and algorithms*. John Wiley & Sons Ltd., Chichester, 1984. Translated from the French by Steven Vajda, A Wiley-Interscience Publication.
333. A. Goralcikova and V. Koubek. A reduct-and-closure algorithm for graphs. In *Proceedings of the 8th Symp. on Math. Foundations of Computer Science*, volume 74 of *Lecture Notes in Computer Science*, pages 301–307. Springer-Verlag, Berlin, 1979.
334. È.Y. Grinberg. Examples of non-Ádám multigraphs. *Latv. Mat. Ezhegodnik*, 31:128–138, 253, 1988.
335. J.W. Grossman and R. Häggkvist. Alternating cycles in edge-partitioned graphs. *J. Combin. Theory Ser. B*, 34:77–81, 1983.
336. M. Grötschel and F. Harary. The graphs for which all strong orientations are hamiltonian. *J. Graph Theory*, 3:221–224, 1979.
337. M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Oper. Res.*, 32(6):1195–1220, 1984.
338. M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

339. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Springer-Verlag, Berlin, 1988.
340. B. Grünbaum. Antidirected Hamiltonian paths in tournaments. *J. Combin. Theory Ser. B*, 11:249–257, 1971.
341. Y. Guo. *Locally semicomplete digraphs*. PhD thesis, RWTH Aachen, Germany, 1995.
342. Y. Guo. Strongly Hamiltonian-connected locally semicomplete digraphs. *J. Graph Theory*, 22(1):65–73, 1996.
343. Y Guo. Path-connectivity in local tournaments. *Discrete Math.*, 167/168:353–372, 1997. 15th British Combinatorial Conference (Stirling, 1995).
344. Y. Guo. Spanning local tournaments in locally semicomplete digraphs. *Discrete Appl. Math.*, 79(1-3):119–125, 1997. 4th Twente Workshop on Graphs and Combinatorial Optimization (Enschede, 1995).
345. Y. Guo. Semicomplete Multipartite Digraphs: A Generalization of Tournaments. German Habilitation Thesis, RWTH Aachen, Germany 1998.
346. Y. Guo and J.H. Kwak. The cycle structure of regular multipartite tournaments. Aachen, submitted.
347. Y. Guo, A. Pinkernell, and L. Volkmann. On cycles through a given vertex in multipartite tournaments. *Discrete Math.*, 164(1-3):165–170, 1997. The Second Krakow Conference on Graph Theory (Zgorzelisko, 1994).
348. Y. Guo, M. Tewes, L. Volkmann, and A. Yeo. Sufficient conditions for semicomplete multipartite digraphs to be hamiltonian. *Discrete Math.*, to appear.
349. Y. Guo and L. Volkmann. Pancyclic locally semicomplete digraphs. Unpublished manuscript. 1992.
350. Y. Guo and L. Volkmann. Cycles in multipartite tournaments. *J. Combin. Theory Ser. B*, 62(2):363–366, 1994.
351. Y. Guo and L. Volkmann. On complementary cycles in locally semicomplete digraphs. *Discrete Math.*, 135(1-3):121–127, 1994.
352. Y. Guo and L. Volkmann. Locally semicomplete digraphs that are complementary m-pancyclic. *J. Graph Theory*, 21(2):121–136, 1996.
353. G. Gutin. Criterion for complete bipartite digraphs to be Hamiltonian. *Vestsī Akad. Navuk BSSR Ser. Fīz.-Mat. Navuk*, 1984(1):109–110, 1984.
354. G. Gutin. On an approach to solving the traveling salesman problem. In *Proceedings of the USSR Conference on System Research*, pages 184–185. Nauka, Moscow, 1984. (in Russian).
355. G. Gutin. Effective characterization of complete bipartite digraphs that have a Hamiltonian path. *Kibernetika (Kiev)*, 4:124–125, 1985.
356. G. Gutin. The radii of n-partite tournaments. *Mat. Zametki*, 40(3):414–417, 430, 1986.
357. G. Gutin. Finding the largest contour in a complete bipartite digraph. *Kibernetika*, (2):117–118, 1987.
358. G. Gutin. Characterization of complete n-partite digraphs that have a Hamiltonian path. *Kibernetika (Kiev)*, (1):107–108, 136, 1988.
359. G. Gutin. Characterization of vertex pancyclic partly oriented k-partite tournaments. *Vestsi Acad. Navuk BSSR Ser.Fiz.-Mat.*, (2):41–46, 1989.
360. G. Gutin. Determining the ranks of vertices in a complete multipartite graph of paired comparisons. *Automation and Remote Control*, (10):139–147, 1989.
361. G. Gutin. m-sources in complete multipartite digraphs. *Vestsī Akad. Navuk BSSR Ser. Fīz.-Mat. Navuk*, 5:101–106, 128, 1989.
362. G. Gutin. *Cycles and paths in directed graphs*. PhD thesis, School of Mathematics, Tel Aviv University, 1993.
363. G. Gutin. Finding a longest path in a complete multipartite digraph. *SIAM J. Discrete Math.*, 6:270–273, 1993.

364. G. Gutin. On cycles in multipartite tournaments. *J. Combin. Theory Ser. B*, 58(2):319–321, 1993.
365. G. Gutin. Polynomial algorithms for finding Hamiltonian paths and cycles in quasi-transitive digraphs. *Australas. J. Combin.*, 10:231–236, 1994.
366. G. Gutin. Minimizing and maximizing the diameter in orientations of graphs. *Graphs Combin.*, 10(3):225–230, 1994.
367. G. Gutin. Characterizations of vertex pancyclic and pancyclic ordinary complete multipartite digraphs. *Discrete Math.*, 141(1-3):153–162, 1995.
368. G. Gutin. Cycles and paths in semicomplete multipartite digraphs, theorems, and algorithms: a survey. *J. Graph Theory*, 19(4):481–505, 1995.
369. G. Gutin. Exponential neighbourhood local search for the travelling salesman problem. *Computers and Operations Research*, 26:313–320, 1999.
370. G. Gutin. Connected (g, f)-factors and supereulerian digraphs. *ARS Combinatoria*, 54:311–317, 2000.
371. G. Gutin, B. Sudakov, and A. Yeo. Note on alternating directed cycles . *Discrete Math.*, 191:101–107, 1998.
372. G. Gutin, M. Tewes, and A. Yeo. Longest paths in strong spanning oriented subgraphs of strong semicomplete multipartite digraphs. *Discrete Math.*, 222:269–274, 2000.
373. G. Gutin and A. Yeo. Ranking the vertices of a complete multipartite paired comparison digraph. *Discrete Appl. Math.*, 69(1-2):75–82, 1996.
374. G. Gutin and A. Yeo. Hamiltonian paths and cycles in hypertournaments. *J. Graph Theory*, 25(4):277–286, 1997.
375. G. Gutin and A. Yeo. Small diameter neighbourhood graphs for the traveling salesman problem: at most four moves from tour to tour. *Computers and Operations Research*, 26:321–327, 1999.
376. G. Gutin and A. Yeo. Kings in semicomplete multipartite digraphs. *J. Graph Theory*, 33:177–183, 2000.
377. G. Gutin and A. Yeo. Note on the path covering number of a semicomplete multipartite digraph. *J. Combin. Math. and Combin. Computing.*, 32:231–237, 2000.
378. G. Gutin and A. Yeo. Orientations of digraphs almost preserving diameter and the one-way street and gossip problems. Technical Report 4, Brunel University, 2000.
379. G. Gutin and A. Yeo. Quasi-hamiltonicity: a series of necessary conditions for a digraph to be hamiltonian. *J. Combin. Theory Ser. B*, 78:232–242, 2000.
380. G. Gutin and A. Yeo. Remarks on hamiltonian digraphs. submitted:4 pp., 2000.
381. G. Gutin and A. Yeo. Polynomial approximation algorithms for the TSP and the QAP with factorial domination number. *Discrete Applied Math.*, To appear.
382. G. Gutin and A. Yeo. Solution of a conjecture of Volkmann on the number of vertices in longest paths and cycles of strong semicomplete multipartite digraphs. *Graphs and Combinatorics*, to appear.
383. W. Gutjahr. *Graph Colourings*. PhD thesis, Freie Universität Berlin, Berlin, 1991.
384. W. Gutjahr, E. Welzl, and G. Woeginger. Polynomial graph-colorings. *Discrete Appl. Math.*, 35(1):29–45, 1992.
385. M. Habib, M. Morvan, and J.-X. Rampon. On the calculation of transitive reduction-closure of orders. *Discrete Math.*, 111:289–303, 1993.
386. R. Häggkvist. On F-Hamiltonian graphs. In *Graph Theory and Related Topics*, pages 219–231. Academic Press, New York, 1979.

387. R. Häggkvist. Hamilton cycles in oriented graphs. *Combin. Probab. Comput.*, 2(1):25–32, 1993.
388. R. Häggkvist, P. Hell, D.J. Miller, and V. Neumann Lara. On multiplicative graphs and the product conjecture. *Combinatorica*, 8(1):63–74, 1988.
389. R. Häggkvist and Y. Manoussakis. Cycles and paths in bipartite tournaments with spanning configurations. *Combinatorica*, 9(1):33–38, 1989.
390. R. Häggkvist and A. Thomason. Trees in tournaments. *Combinatorica*, 11(2):123–130, 1991.
391. R. Häggkvist and C. Thomassen. On pancyclic digraphs. *J. Combin. Theory Ser. B*, 20(1):20–40, 1976.
392. A. Hajnal, E.C. Milner, and E. Szemerédi. A cure for the telephone disease. *Canad. Math. Bull.*, 15:447–450, 1972.
393. P. Hall. On representation of subsets. *J. London Math. Soc.*, 10:26–30, 1935.
394. Y.O. Hamidoune. *Contribution a l' étude de la connectivité d'un graphe*. PhD thesis, 1980.
395. Y.O. Hamidoune. An application of connectivity theory in graphs to factorizations of elements in finite groups. *Europ. J. Combin.*, 2:349–355, 1981.
396. Y.O. Hamidoune. A note on minimal directed graphs with given girth. *J. Combin. Theory Ser. B*, 43:343–348, 1987.
397. S. Hansen. Heuristic and exact methods for solving the feedback arc set problem (Masters Thesis University of Southern Denmark, Odense), 1999.
398. J. Hao and J.B. Orlin. A faster algorithm for finding the minimum cut in a directed graph. *J. Algorithms*, 17(3):424–446, 1994. Third Annual ACM-SIAM Symposium on Discrete Algorithms (Orlando, FL, 1992).
399. F. Harary. The maximum connectivity of a graph. *Proc. Nat. Acad. Sci. U.S.A.*, 48:1142–1146, 1962.
400. F. Harary, J.A. Kabell, and F.R. McMorris. Bipartite intersection graphs. *Comment. Math. Univ. Carolin.*, 23:739–745, 1982.
401. F. Harary, J. Krarup, and A. Schwenk. Graphs suppressible to an edge. *Canadian Math. Bull.*, 15:201–204, 1971.
402. F. Harary and L. Moser. The theory of round robin tournaments. *Amer. Math. Mon.*, 73:231–246, 1966.
403. F. Harary and R.Z. Norman. Some properties of line digraphs. *Rend. Circ. Mat. Palermo*, 9(2):161–168, 1960.
404. F. Harary, R.Z. Norman, and D. Cartwright. *Structural Models*. John Wiley and Sons, 1965.
405. F. Havet. Finding an oriented hamiltonian path in a tournament. *submitted*, 1999.
406. F. Havet. Hamiltonian cycles in tournaments. *Submitted*, 1999.
407. F. Havet and S. Thomassé. Median orders of tournaments: a tool for the second neighbourhood problem and Sumner's conjecture. *submitted*, 1999.
408. F. Havet and S. Thomassé. Oriented hamiltonian paths in tournaments: a proof of Rosenfeld's conjecture. *J. Comb. Theory, Ser B*, 78:243–273, 2000.
409. S.M. Hedetniemi, S.T. Hedetniemi, and A. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18:129–134, 1988.
410. P. Hell, J. Bang-Jensen, and J. Huang. Local tournaments and proper circular arc graphs. In *Algorithms (Tokyo, 1990)*, volume 450 of *Lecture Notes in Comput. Sci.*, pages 101–108. Springer, Berlin, 1990.
411. P. Hell and J. Huang. Lexicographic orientation and representation algorithms for comparability graphs, proper circular arc graphs, and proper interval graphs. *J. Graph Theory*, 20(3):361–374, 1995.
412. P. Hell and J. Nešetřil. On the complexity of H-coloring. *J. Combin. Theory Ser. B*, 48(1):92–110, 1990.

413. P. Hell and J. Nešetřil. The core of a graph. *Discrete Math.*, 109(1-3):117–126, 1992. Algebraic graph theory (Leibnitz, 1989).
414. P. Hell, J. Nešetřil, and X. Zhu. Duality and polynomial testing of tree homomorphisms. *Trans. Amer. Math. Soc.*, 348(4):1281–1297, 1996.
415. P. Hell and M. Rosenfeld. The complexity of finding generalized paths in tournaments. *J. Algorithms*, 4(4):303–309, 1983.
416. P. Hell, H.S. Zhou, and X.D. Zhu. On homomorphisms to acyclic local tournaments. *J. Graph Theory*, 20(4):467–471, 1995.
417. P. Hell and X.D. Zhu. Homomorphisms to oriented paths. *Discrete Math.*, 132(1-3):107–114, 1994.
418. P. Hell and X.D. Zhu. The existence of homomorphisms to oriented cycles. *SIAM J. Discrete Math.*, 8(2):208–222, 1995.
419. R.L. Hemminger and L.W. Beineke. Line graphs and line digraphs. In *Selected Topics in Graph Theory (Beineke and Wilson, eds.)*, pages 271–305. Academic Press, London, 1978.
420. G.R.T. Hendry. Extending cycles in directed graphs. *J. Combin. Theory Ser. B*, 46(2):162–172, 1989.
421. G.R.T. Hendry. Extending cycles in graphs. *Discrete Math.*, 85(1):59–72, 1990.
422. M.R. Henzinger, S. Rao, and H.N. Gabow. Computing vertex connectivity: new bounds from old techniques. In *37th Annual Symposium on Foundations of Computer Science (Burlington, VT, 1996)*, pages 462–471. IEEE Comput. Soc. Press, Los Alamitos, CA, 1996.
423. J.S. Heslop-Harrison and M.D. Bennett. Prediction and analysis of spacial order in haploid chromosome complements. *Proc. Roy. Soc. London*, B:211–223, 1983.
424. J.S. Heslop-Harrison and M.D. Bennett. The spacial order of chromosomes in root-tip metaphases of Aegilops umbellulata. *Proc. Roy. Soc. London*, B:225–239, 1983.
425. C. Heuchenne. Sur une certaine correspondance entre graphs. *Bull. Soc. Roy. Sci. Liége*, 33:743–753, 1964.
426. M.-C. Heydemann and D. Sotteau. About some cyclic properties in digraphs. *J. Combin. Theory Ser. B*, 38(3):261–278, 1985.
427. M.-C. Heydemann, D. Sotteau, and C. Thomassen. Orientations of Hamiltonian cycles in digraphs. *Ars Combin.*, 14:3–8, 1982.
428. M.C. Heydemann. On cycles and paths in digraphs. *Discrete Math.*, 31:217–219, 1980.
429. A.J.W. Hilton. Alternating Hamiltonian circuits in edge-coloured bipartite graphs. *Discrete Applied Math.*, 35:271–273, 1992.
430. C.T. Hoang and B. Reed. A note on short cycles in digraphs. *Discrete Math.*, 66(1-2):103–107, 1987.
431. A.J. Hoffmann. Some recent applications of the theory of linear inequalities to extremal combinatorial analysis. In *Combinatorial Analysis, edited by R. Bellman and M. Hall*, pages 113–128. American Mathematical Society, Providence, RI, 1960.
432. J.E. Hopcroft and R.E. Tarjan. Efficient planarity testing. *Journal of the Association for Computing Machinery*, 21:549–568, 1974.
433. J. Hromkovič, R. Klasing, B. Monien, and R. Peine. Dissemination of information in interconnection networks (broadcasting & gossiping). In *Combinatorial network theory*, pages 125–212. Kluwer Acad. Publ., Dordrecht, 1996.
434. H.T. Hsu. An algorithm for finding a minimal equivalent graph of a digraph. *J. Assoc. Comput. Mach.*, 22:11–16, 1975.

435. J. Huang. *Tournament-like oriented graphs* . PhD thesis, School of Computing Science, Simon Fraser University, Burnaby B.C. Canada, 1992.
436. J. Huang. On the structure of local tournaments. *J. Combin. Theory Ser. B*, 63(2):200–221, 1995.
437. J. Huang. A note on spanning local tournaments in locally semicomplete digraphs. *Discrete Applied Math.*, 89:277–279, 1998.
438. J. Huang. Which digraphs are round? *Australasian J. Combinatorics*, 19:203–208, 1999.
439. F.K. Hwang. The Hamiltonian property of linear functions. *Oper. Res. Letters*, 6:125–127, 1987.
440. T. Ibaraki and S. Poljak. Weak three-linking in Eulerian digraphs. *SIAM J. Discrete Math.*, 4(1):84–98, 1991.
441. M. Imase and M. Itoh. Design to monimize a diameter on Building block networks. *IEEE Trans. on Computers*, C-30:439–443, 1981.
442. M. Imase and M. Itoh. Design for directed graphs with minimum diameter. *IEEE Trans. on Computers*, C-32:782–784, 1983.
443. M. Imase, I. Soneoka, and K. Okada. Connectivity of regular directed graphs with small diameter. *IEEE Trans. on Computers*, C-34:267–273, 1985.
444. M. Imase, I. Soneoka, and K. Okada. A fault tolerant processor interconnection network. *Systems and Computers in Japan*, 17:21–30, 1986.
445. M. Imori, M. Matsumoto, and H. Yamada. The line digraph of a regular and pancircular digraph is also regular and pan circular. *Graphs and Combinatorics*, 4:235–239, 1988.
446. G. Isaak. Tournaments as feedback arc sets. *Electronic J. Combinatorics*, 2:19pp, 1995.
447. S. Iwata, L. Fleischer, and S. Fujishige. A strongly polynomial time algorithm for minimizing submodular functions. manuscript 1999.
448. Carlier J. and P. Villon. A new heuristic for the travelling salesman problem. *RAIRO, Recherche Operationelle*, 24:245–253, 1990.
449. B. Jackson. Long paths and cycles in oriented graphs. *J. Graph Theory*, 5(2):145–157, 1981.
450. B. Jackson. A Chvátal-Erdős condition for Hamilton cycles in digraphs. *J. Combin. Theory Ser. B*, 43(3):245–252, 1987.
451. B. Jackson. Some remarks on arc-connectivity, vertex splitting, and orientation in graphs and digraphs. *J. Graph Theory*, 12(3):429–436, 1988.
452. B. Jackson and O. Ordaz. A Chvátal- Erdős condition for $(1,1)$-factors in digraphs. *Discrete Math.*, 57(1-2):199–201, 1985.
453. B. Jackson and O. Ordaz. Chvátal-Erdős conditions for paths and cycles in graphs and digraphs. A survey. *Discrete Math.*, 84(3):241–254, 1990.
454. H. Jacob and H. Meyniel. About quasi-kernels in a digraph. *Discrete Math.*, 154(1-3):279–280, 1996.
455. F. Jaeger. On nowhere-zero flows in multigraphs. *Congressus Numerantium*, 15:373–378, 1976.
456. F. Jaeger. On five-edge-colorings of cubic graphs and nowhere-zero flow problems. *Ars Combin.*, 20(B):229–244, 1985.
457. F. Jaeger. Nowhere-zero flow problems. In *Selected topics in graph theory, 3*, pages 71–95. Academic Press, San Diego, CA, 1988.
458. J. Janssen. The Dinitz problem solved for rectangles. *Bull. Amer. Math. Soc.*, 29:243–249, 1993.
459. T.R. Jensen and B. Toft. *Graph coloring problems*. John Wiley & Sons, Inc., New York, 1995. A Wiley-Interscience Publication.
460. W.S. Jewell. Optimal flow through networks. Technical Report 8, OR center, MIT, Cambridge, Mass., 1958.

461. J. Jirásek. On a certain class of multidigraphs, for which reversal of no arc decreases the number of their cycles. *Comment. Math. Univ. Carolinae*, 28:185–189, 1987.
462. J. Jirásek. Some remarks on Ádám's conjecture for simple directed graphs. *Discrete Math.*, 108:327–332, 1992.
463. D.B. Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the Association for Computing Machinery*, 24:1–13, 1977.
464. D.S. Johnson, C.R. Aragon, L. McGeoch, and C. Schevon. Optimization by simulated annealing: an experimental evaluation; part 1, Graph partitioning. *Operations Research*, 37:865–892, 1989.
465. D.S. Johnson, C.R. Aragon, L. McGeoch, and C. Schevon. Optimization by simulated annealing: an experimental evaluation; part 2, Graph coloring and number partitioning. *Operations Research*, 39:378–406, 1991.
466. D.S. Johnson and L.A. McGeoch. The traveling salesman problem: A case study in local optimization. In E.H.L. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 215–310. John Wiley &Sons, New York, 1997.
467. T. Jordán. Increasing the vertex-connectivity in directed graphs. In *Algorithms—ESA '93 (Bad Honnef, 1993)*, volume 726 of *Lecture Notes in Comput. Sci.*, pages 236–247. Springer, Berlin, 1993.
468. T. Jordán. *Connectivity augmentation problems in Graphs*. PhD thesis, Department of Computer Science, Eötvös University, Budapest, 1994.
469. T. Jordán. On the optimal vertex-connectivity augmentation. *J. Combin. Theory Ser. B*, 63:8–20, 1995.
470. H.A. Jung. Eine Verallgemeinerung des n-fachen zusammenhangs für Graphen. *Math. Ann.*, 187:95–103, 1970.
471. M. Kano. Ranking the vertices of an r-partite paired comparison digraph. *Discrete Appl. Math.*, 17(3):245–253, 1987.
472. M. Kano and A. Sakamoto. Ranking the vertices of a weighted digraph using the length of forward arcs. *Networks*, 13(1):143–151, 1983.
473. M. Kano and A. Sakamoto. Ranking the vertices of a paired comparison digraph. *SIAM J. Algebraic Discrete Methods*, 6(1):79–92, 1985.
474. R.M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103. Plenum, New York, 1972.
475. A.V. Karzanov. The problem of finding the maximal flow in a network by the method of preflows. *Dokl. Akad. Nauk SSSR*, 215:49–52, 1974.
476. J.G. Kemeny and J.L. Snell. *Finite Markov Chains*. Springer-Verlag, New York, 1976.
477. A. Kemnitz and B. Greger. A forbidden subdigraph condition implying an oriented graph to be Hamiltonian. *Congr. Numer.*, 130:127–131, 1998.
478. S. Khuller, B. Raghavachari, and N. Young. Approximating the minimum equivalent digraph. *SIAM J. Computing*, 24(4):859–872, 1995.
479. S. Khuller, B. Raghavachari, and N. Young. On strongly connected digraphs with bounded cycle length. *Discrete Appl. Math.*, 69(3):281–289, 1996.
480. M. Klein. A primal method for minimum cost flows with applications to the assignment and transportation problems. *Management Science*, 14:205–220, 1967.
481. D.E. Knuth. *The art of computer programming. Vol. 1: Fundamental algorithms*. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont, 1968. Second printing.
482. W. Kocay and D. Stone. An algorithm for balanced flows. *J. Combin. Math. Combin. Comput.*, 19:3–31, 1995.

483. K.-M. Koh. Even circuits in directed graphs and Lovasz's conjecture. *Bull. Malaysian Math. Soc.*, 7(3):47–52, 1976.
484. K.M. Koh and B.P. Tan. The diameters of a graph and its orientations. Technical report, Department of Mathematics, National University of Singapore, 1992.
485. K.M. Koh and B.P. Tan. Kings in multipartite tournaments. *Discrete Math.*, 147:171–183, 1995.
486. K.M. Koh and B.P. Tan. Number of 4-kings in bipartite tournaments with no 3-kings. *Discrete Math.*, 154(1-3):281–287, 1996.
487. K.M. Koh and B.P. Tan. The diameter of an orientation of a complete multipartite graph. *Discrete Math.*, 149(1-3):131–139, 1996.
488. K.M. Koh and B.P. Tan. The minimum diameter of orientations of complete multipartite graphs. *Graphs Combin.*, 12(4):333–339, 1996.
489. K.M. Koh and B.P. Tan. The number of kings in a multipartite tournament. *Discrete Math.*, 167/168:411–418, 1997.
490. K.M. Koh and E.G. Tay. On optimal orientations of Cartesian products of even cycles and paths. *Networks*, 30(1):1–7, 1997.
491. K.M. Koh and E.G. Tay. Optimal orientations of products of paths and cycles. *Discrete Appl. Math.*, 78(1-3):163–174, 1997.
492. K.M. Koh and E.G. Tay. On optimal orientations of Cartesian products of graphs (I). *Discrete Math.*, 190:115–136, 1998.
493. K.M. Koh and E.G. Tay. On optimal orientations of Cartesian products of trees. *Graphs and Combinatorics*, to appear, 1999.
494. K.M. Koh and E.G. Tay. On optimal orientations of Cartesian products with a bipartite graph. *Discrete Applied Math.*, 98:103–120, 1999.
495. K.M. Koh and E.G. Tay. On optimal orientations of Cartesian products of graphs (II): complete graphs and even cycles. *Discrete Math.*, 211:75–102, 2000.
496. K.M. Koh and E.G. Tay. On optimal orientations of G vertex-multiplications. *Discrete Math.*, 219:153–171, 2000.
497. D. König. Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre. *Math. Ann.*, 77:454–465, 1916.
498. D. König. Graphs and matrices (in Hungarian). *Mat. Fiz. Lapok*, 38:116–119, 1931.
499. D. König. *Theorie der endlichen und unendlichen Graphen*. Akademische Verlagsgesellschaft, 1936.
500. J-C. König, D.W. Krumme, and E. Lazard. Diameter preserving orientation of the torus. *Networks*, 32:1–11, 1998.
501. A. Kotzig. On the theory of finite graphs with a linear factor II. *Math. Fyz. Čazopis*, 9:73–91, 1959.
502. A. Kotzig. Moves without forbidden transitions in a graph. *Math. Fyz. Čazopis*, 18:76–80, 1968.
503. A. Kotzig. The decomposition of a directed graph into quadratic factors consisting of cycles. *Acta Fac. Rerum Natur. Univ. Comenian. Math. Publ.*, 22:27–29, 1969.
504. D.W. Krumme, G. Cybenko, and K.N. Venkataraman. Gossiping in minimal time. *SIAM J. Computing*, 21(1):111–139, 1992.
505. H.W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.
506. A. Kunzmann and Wunderlich H.J. An analytical approach to the partial scan problem. *J. Electronic Testing: Theory and Applications*, 1:163–174, 1990.
507. C. Kuratowski. Sur le probléme des courbes gauches en topologie. *Fund. Math.*, 15:271–283, 1930.

508. H.G. Landau. On dominance relations and the structure of animal societies III. The condition for a score structure. *Bull. Math. Biophys.*, 15:143–148, 1953.
509. E.L. Lawler. *Combinatorial Optimization: Networks and Mathroids*. Holt, Rinehart and Winston, New York, 1976.
510. E.L. Lawler. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Ann. Discrete Math.*, 2:75–90, 1978. Algorithmic aspects of combinatorics (Conf., Vancouver Island, B.C., 1976).
511. E.L. Lawler, J.K. Lenstra, A.H.G. Rinooykan, and D.B. Shmoys, editors. *Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley & Sons, New York, 1985.
512. C.E. Leiserson and J.B Saxe. Retiming synchronous circuitry. *Algoritmica*, 6:5–35, 1991.
513. L.A. Levin. Universal sorting problems. *Problems of Information Transmission*, 9:265–266, 1973.
514. M. Lewin. On maximal circuits in directed graphs. *J. Combin. Theory, Ser. B*, 18:175–179, 1975.
515. N. Linial, L. Lovász, and A. Wigderson. Rubber bands, convex embeddings and graph connectivity. *Combinatorica*, 8(1):91–102, 1988.
516. C. Little, K. Teo, and H. Wang. On a conjecture on directed cycles in a directed bipartite graph. *Graphs Combin.*, 13(3):267–273, 1997.
517. J. Liu and H.S. Zhou. Graphs and digraphs with given girth and connectivity. *Discrete Math.*, 132(1-3):387–390, 1994.
518. X. Liu and D.B. West. Line digraphs and coreflexive vertex sets. *Discrete Math.*, 188(1-3):269–277, 1998.
519. L. Lovász. Connectivity in digraphs. *J. Combin. Theory Ser. B*, 15:174–177, 1973.
520. L. Lovász. Coverings and coloring of hypergraphs. volume 8, pages 3–12, Winnipeg, Man., 1973. Utilitas Math.
521. L. Lovász. On two min–max theorems in graph theory. *J. Combin. Theory Ser. B*, 21:26–30, 1976.
522. L. Lovász. *Combinatorial problems and exercises*. North-Holland Publishing Co., Amsterdam, 1979.
523. L. Lovász. Connectivity algorithms using rubber bands. In *Foundations of software technology and theoretical computer science (New Delhi, 1986)*, pages 394–411. Springer, Berlin, 1986.
524. L. Lovász. *Combinatorial problems and exercises*. North-Holland Publishing Co., Amsterdam, second edition, 1993.
525. L. Lovász and M.D. Plummer. *Matching theory*. North-Holland Publishing Co., Amsterdam, 1986. Annals of Discrete Mathematics, 29.
526. C.L. Lucchesi. *A minimax equality for directed graphs*. PhD thesis, University of Waterloo, Ontario, Canada, 1976.
527. C.L. Lucchesi and M.C.M.T. Giglio. On the connection between the undirected and the acyclic directed two disjoint paths problem. *Ars Combin.*, 47:191–200, 1997.
528. C.L. Lucchesi and D.H. Younger. A minimax theorem for directed graphs. *J. London Math. Soc. (2)*, 17(3):369–374, 1978.
529. J.F. Lynch. The equivalence of theorem proving and the interconnection problem. *(ACM) SIGDA Newsletter*, 5(3):31–36, 1975.
530. G. MacGillivray. *The complexity of generalized colouring*. PhD thesis, Simon Fraser University, Burnaby, B.C. Canada, 1990.
531. G. MacGillivray. On the complexity of colouring by vertex-transitive and arc-transitive digraphs. *SIAM J. Discrete Math.*, 4(3):397–408, 1991.

532. W. Mader. Minimale n-fach kantenzusammenhängende Graphen. *Math. Ann.*, 191:21–28, 1971.
533. W. Mader. Ecken vom Grad n in minimalen n-fach zusammenhängenden Graphen. *Arch. Math. (Basel)*, 23:219–224, 1972.
534. W. Mader. 1-factoren von Graphen. *Math. Ann.*, 201:269–282, 1973.
535. W. Mader. Ecken vom Innen- und Aussengrad n in minimal n-fach kantenzusammenhängenden Digraphen. *Arch. Math. (Basel)*, 25:107–112, 1974.
536. W. Mader. A reduction method for edge-connectivity in graphs. *Ann. Discrete Math.*, 3:145–164, 1978. Advances in graph theory (Cambridge Combinatorial Conf., Trinity College, Cambridge, 1977).
537. W. Mader. Konstruktion aller n-fach kantenzusammenhängenden Digraphen. *European J. Combin.*, 3(1):63–67, 1982.
538. W. Mader. Minimal n-fach zusammenhängende Digraphen. *J. Combin. Theory Ser. B*, 38(2):102–117, 1985.
539. W. Mader. Ecken von kleinem Grad in kritisch n-fach zusammenhängenden Digraphen. *J. Combin. Theory Ser. B*, 53(2):260–272, 1991.
540. W. Mader. On vertices of degree n in minimally n-connected graphs and digraphs. In *Combinatorics, Paul Erdős is eighty, Vol. 2 (Keszthely, 1993)*, pages 423–449. János Bolyai Math. Soc., Budapest, 1996.
541. W. Mader. On vertices of out-degree n in minimally n-connected digraphs. manuscript 1999.
542. H. Maehara. A digraph represented by a family of boxes or spheres. *J. Graph Theory*, 8(3):431–439, 1984.
543. F. Maffray. Kernels in perfect line graphs. *Journal of Combin. Theory. Series B*, 55:1–8, 1992.
544. V.M. Malhotra, M.P. Kumar, and S.N. Maheshwari. An $O(n^3)$ algorithm for finding maximum flows in networks. *Information Processing Letters*, 7:277–278, 1978.
545. Y. Manoussakis. k-linked and k-cyclic digraphs. *J. Combin. Theory Ser. B*, 48(2):216–226, 1990.
546. Y. Manoussakis. A linear-time algorithm for finding Hamiltonian cycles in tournaments. *Discrete Appl. Math.*, 36(2):199–201, 1992.
547. Y. Manoussakis. Directed Hamiltonian graphs. *J. Graph Theory*, 16(1):51–59, 1992.
548. Y. Manoussakis. Alternating paths in edge-coloured complete graphs. *Discrete Applied Math.*, 56:297–309, 1995.
549. Y. Manoussakis, M. Spyratos, Zs. Tuza, and M. Voigt. Minimal colorings for properly colored subgraphs. *Graphs and Combinatorics*, 12:345–360, 1996.
550. D.A. Marcus. Spanning subgraphs of k-connected digraphs. *J. Combin. Theory Ser. B*, 30(1):21–31, 1981.
551. D.A. Marcus. Directed cycles with chords. *J. Graph Theory*, 31(1):17–28, 1999.
552. S. Marshall. On the existence of k-tournaments with given automorphism group. *Discrete Math.*, 152(1-3):259–268, 1996.
553. S. Marshall. Representing posets with k-tournaments. *Order*, 13(2):147–158, 1996.
554. S. Martello and P. Toth. Finding a minimum equivalent graph of a digraph. *Networks*, 12:89–100, 1982.
555. T. Masuzawa, K. Hagihara, and N. Tokura. An optimal time algorithm for the k-vertex-connectivity unweighted augmentation problem for rooted directed trees. *Discrete Appl. Math.*, 17(1-2):67–105, 1987.
556. H.A. Maurer, J.H. Sudborough, and E. Welzl. On the complexity of the general coloring problem. *Inform. and Control*, 51(2):128–145, 1981.

557. D. May. The next generation transputers and beyond. In *Distributed Memory Computing*, volume 487 of *Lecture Notes in Comput. Sci.*, pages 7–22. Springer Verlag, Berlin, 1991.
558. J.E. McCanna. Orientations of the n-cube with minimum diameter. *Discrete Math.*, 68(2-3):309–310, 1988.
559. W. McCuaig. Intercyclic digraphs. In *Graph structure theory (Seattle, WA, 1991)*, volume 147 of *Contemp. Math.*, pages 203–245. Amer. Math. Soc., Providence, RI, 1993.
560. C. McDiarmid. Probability. In *Graph connections. Relationships between graph theory and other areas of mathematics (eds: Beineke, L.W. et al.)*, pages 194–207. Oxford University Press, Oxford, UK, 1997.
561. K. Mehlhorn. *Data Structures and Algorithms 2: Graph Algorithms and NP-completeness.* Springer-Verlag, Berlin, 1984.
562. K. Menger. Zur allgemeinen Kurventheorie. *Fund. Math.*, 10:96–115, 1927.
563. A. Metzlar. Disjoint paths in acyclic digraphs. *J. Combin. Theory Ser. B*, 57(2):228–238, 1993.
564. H. Meyniel. Une condition suffisante d'existence d'un circuit hamiltonien dans un graphe orienté. *J. Combin. Theory Ser. B*, 14:137–147, 1973.
565. M. Miller. Digraph covering and its application to two optimizatio problems for digraphs. *Australas. J. Combinatorics*, 3:151–164, 1991.
566. M. Miller and I. Fris. Maximum order digraphs for diameter 2 or degree 2. In *Pullman Volume of Graphs and Matrices*, volume 139 of *Lecture Notes in Pure and Applied Math.*, pages 269–278. Pullman, New York, 1992.
567. G.J. Minty. A theorem on n-colouring the points of a linear graph. *American Mathematical Monthly*, 69:623–624, 1962.
568. C.L. Monma and J.B. Sidney. A general algorithm for optimal job sequencing with series-parallel constraints. *Math. of Operations Research*, 4:215–224, 1977.
569. J.W. Moon. Solution to problem 463. *Math. Mag.*, 35:189, 1962.
570. J.W. Moon. On subtournaments of a tournament. *Canad. Math. Bull.*, 9:297–301, 1966.
571. J.W. Moon. *Topics on tournaments.* Holt, Rinehart and Winston, New York, 1968.
572. E.F. Moore. The shortest path through a maze. In *Proc. of the Int. Symp. on the Theory of Switching*, pages 285–292. Harward University Press, 1959.
573. A.O. Morris. *Linear Algebra: an Introduction.* Van Nostrand Reinhold, Workingham, England, second edition edition, 1982.
574. M. Morvan and L. Viennot. Parallel comparability graph recognition and modular decomposition. In *STACS 96 (Grenoble, 1996)*, pages 169–180. Springer, Berlin, 1996.
575. H.M. Mulder. Julius Petersen's theory of regular graphs. *Discrete Math.*, 100:157–175, 1992.
576. H. Müller. Recognizing interval digraphs and interval bigraphs in polynomial time. *Discrete Appl. Math.*, 78:189–205, 1997.
577. J.H. Muller and J. Spinrad. Incremental modular decomposition. *J. Assoc. Comput. Mach.*, 36(1):1–19, 1989.
578. K.G. Murty. *Network programming.* Prentice Hall Inc., Englewood Cliffs, NJ, 1992.
579. H. Nagamochi and T. Ibaraki. A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7(5-6):583–596, 1992.
580. H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM J. Discrete Math.*, 5(1):54–66, 1992.

581. H. Nagamochi and T. Ibaraki. Deterministic $\widetilde{O}(nm)$ time edge-splitting in undirected graphs. *J. Comb. Optim.*, 1(1):5–46, 1997.
582. H. Nagamochi and T. Ibaraki. A note on minimizing submodular functions. *Inform. Process. Lett.*, 67(5):239–244, 1998.
583. C.St.J.A. Nash-Williams. On orientations, connectivity and odd-vertex-pairings in finite graphs. *Canad. J. Math.*, 12:555–567, 1960.
584. C.St.J.A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *J. London Math. Soc.*, 36:445–450, 1961.
585. C.St.J.A. Nash-Williams. Decomposition of finite graphs into forests. *J. London Math. Soc.*, 39:12, 1964.
586. C.St.J.A. Nash-Williams. Problem 47. In *Proceedings of Colloq. Tihany 1966*, page 366. Academic Press, 1968.
587. C.St.J.A. Nash-Williams. Hamilton circuits in graphs and digraphs. In *The many facets of graph theory*, Springer Verlag Lecture Notes 110, pages 237–243. Springer Verlag, 1969.
588. C.St.J.A. Nash-Williams. Hamilton circuits. In *Studies in Graph Theory Part II*, Studies in Mathematics 12, pages 301–360. M.A.A., Washington, 1975.
589. M. Natu and S.-C. Fang. The point-to-point connection problem–analysis and algorithms. *Discrete Applied Math.*, 78:207–226, 1997.
590. J. Nešetřil and X.D. Zhu. Path homomorphisms. *Math. Proc. Cambridge Philos. Soc.*, 120(2):207–220, 1996.
591. L.L. Ng. Hamiltonian decomposition of complete regular multipartite digraphs. *Discrete Math.*, 177(1-3):279–285, 1997.
592. L.L. Ng. Hamiltonian decomposition of lexicographic products of digraphs. *J. Combin. Theory Ser. B*, 73(2):119–129, 1998.
593. T. Nishimura. Short cycles in digraphs. *Discrete Math.*, 72(1-3):295–298, 1988.
594. C. Olsen. Heuristics for combinatorial optimization problems (course project in Danish University of Southern Denmark, Odense), 1998.
595. O. Ore. *Theory of graphs*. American Mathematical Society, Providence, R.I., 1962. American Mathematical Society Colloquium Publications, Vol. XXXVIII.
596. Erdős P. and G. Szekeres. A combinatorial problem in geometry. *Compositio Math.*, 2:463–470, 1935.
597. M. Overbeck-Larisch. Hamiltonian paths in oriented graphs. *J. Combin. Theory Ser. B*, 21(1):76–80, 1976.
598. M. Overbeck-Larisch. A theorem on pancyclic-oriented graphs. *J. Combin. Theory Ser. B*, 23(2-3):168–173, 1977.
599. L. Pan, G. Zhou, and K.-M. Zhang. A note on hypertournaments. *Chinese Science Bull.*, 44:1460–1463, 1999.
600. C.H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1982.
601. L.E. Penn and D. Witte. When the Cartesian product of two directed cycles is hypohamiltonian. *J. Graph Theory*, 7:441–443, 1983.
602. Y. Perl and Y. Shiloach. Finding two disjoint paths between two pairs of vertices in a graph. *J. Assoc. Comput. Machin.*, 25:1–9, 1978.
603. J. Petersen. Die Theorie der regulären graphs. *Acta Math.*, 15:193–220, 1891.
604. V. Petrović. Kings in bipartite tournaments. *Discrete Math.*, 173(1-3):187–196, 1997.
605. V. Petrović and C. Thomassen. Kings in k-partite tournaments. *Discrete Math.*, 98(3):237–238, 1991.
606. P.A. Pevzner. DNA physical mapping and alternating eulerian cycles in colored graphs. *Algorithmica*, 13, 1995.

607. J. Plesník. Remarks on diameters of orientations of graphs. *Acta Math. Univ. Comenian.*, 46/47:225–236 (1986), 1985.
608. J. Plesník. Remarks on diameters of orientations of graphs. *Acta Math. Univ. Comenian.*, 46/47:225–236 (1986), 1985.
609. J. Plesník and S. Znám. Strongly geodetic directed graphs. *Acta F. R. N. Univ. Comen.-Mathe.*, 29:29–34, 1975.
610. L. Pósa. A theorem concerning Hamiltonian lines. *Publ. Math. Inst. Hungar. Acad. Sci.*, 7:225–226, 1962.
611. C.N. Potts and S.L. van de Velde. Dynasearch – iterative local improvement by dynamic programming: Part I, the traveling salesman problem. Univ. Southampton, UK, November 1995.
612. E. Prisner. *Familien zusammenhängender Teilgraphen eines Graphen und ihre Durchschnittsgraphen.* PhD thesis, University of Hamburg, 1988.
613. E. Prisner. *Graph dynamics.* Longman, Harlow, 1995.
614. E. Prisner. Line graphs and generalizations—a survey. *Congressus Numerantium*, 116:193–229, 1996. Surveys in graph theory (San Francisco, CA, 1995).
615. A. Punnen and S. Kabadi. Domination analysis of some heuristics for the asymmetric traveling salesman problem. April 1999.
616. A.P. Punnen. The traveling salesman problem: new polynomial approximation algorithms and domination analysis. University of New Brunswick-St. John, December 1996.
617. M. Queyranne. Minimizing symmetric submodular functions. *Math. Programming*, 82(1-2, Ser. B):3–12, 1998. Networks and matroids; Sequencing and scheduling.
618. J. Radhakrishnan and A. Srinivasan. Improved bounds and algorithms for hypergraph 2-colouring. *Random Structures and Algorithms*, 16:4–32, 2000.
619. R. Rado. Note on independence functions. *Proc. London Math. Soc.*, 7:300–320, 1957.
620. V. Ramachandran. A minimax arc theorem for reducible flow graphs. *SIAM J. Discrete Math.*, 3:554–560, 1990.
621. F.P. Ramsey. On a problem of formal logic. *Proc. London Math. Soc.*, 30:264–286, 1930.
622. J. Rattner. The new age of supercomputing. In *Distributed Memory Computing*, volume 487 of *Lecture Notes in Comput. Sci.*, pages 1–6. Springer Verlag, Berlin, 1991.
623. A. Recski. *Matroid theory and its applications in electric network theory and in statics.* Springer-Verlag, Berlin, 1989.
624. S. M. Reddy, D.K. Pradhan, and J. G. Kuhl. Directed graphs with minimal diameter and maximal connectivity. Tech. rep., School of Engineering Oakland Univ., 1980.
625. L. Rédei. Ein kombinatorischer Satz. *Acta Litt. Szeged*, 7:39–43, 1934.
626. B. Reed, N. Robertson, P.D. Seymour, and R. Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996.
627. B.A. Reed and F.B. Shepherd. The Gallai-Younger conjecture for planar graphs. *Combinatorica*, 16(4):555–566, 1996.
628. C.R. Reeves (editor). *Modern heuristic techniques for combinatorial problems.* McGraw-Hill Book Company, 1995.
629. K.B. Reid. Two complementary circuits in two-connected tournaments. In *Cycles in graphs (Burnaby, B.C., 1982)*, volume 115 of *North-Holland Math. Stud.*, pages 321–334. North-Holland, Amsterdam, 1985.
630. K.B. Reid. Tournaments: scores, kings, generalizations and special topics. *Congressus Numerantium*, 115:171–211, 1996. Surveys in graph theory (San Francisco, CA, 1995).

631. G. Reinelt. *The linear ordering problem: algorithms and applications.* Heldermann Verlag, Berlin, 1985.
632. G. Reinelt. *The Travelling Salesman: Computational Solutions for TSP Applications.* Springer-Verlag, Berlin, 1994.
633. F. Rendl. Quadratic assignment problems on series-parallel digraphs. *Z. Oper. Res. Ser. A-B*, 30(3):A161–A173, 1986.
634. P.I. Richards. Precedence constraints and arrow diagrams. *SIAM Rev.*, 9:548–553, 1967.
635. M. Richardson. Solution of irreflective relations. *Ann. Math.*, 58:573–580, 1953.
636. M.B. Richey, R.G. Parker, and R.L. Rardin. An efficiently solvable case of the minimum weight equivalent subgraph problem. *Networks*, 15(2):217–228, 1985.
637. H.E. Robbins. A theorem on graphs with an application to a problem on traffic control. *American Mathematical Monthly*, 46:281–283, 1939.
638. F.S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs I: Large grids. *SIAM J. Discrete Math.*, 1:199–222, 1988.
639. F.S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs II: Two east-west avenues or north-south streets. *Networks*, 19:221–233, 1989.
640. F.S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs III: Three east-west avenues or north-south streets. *Networks*, 22:109–143, 1992.
641. F.S. Roberts and Y. Xu. On the optimal strongly connected orientations of city street graphs IV: Four east-west avenues or north-south streets. *Discrete Applied Math.*, 49:331–356, 1994.
642. N. Robertson and P.D. Seymour. Graph minors. XIII: The disjoint paths problem. *J. Comb. Theory, Ser. B*, 63:65–110, 1995.
643. N. Robertson, P.D. Seymour, and R. Thomas. Permanents, Pfaffian orientations, and even directed circuits. *Ann. Math.*, 150:929–975, 1999.
644. M. Rosenfeld. Antidirected Hamiltonian paths in tournaments. *J. Comb. Theory, Ser. B*, 12:93–99, 1971.
645. M. Rosenfeld. Antidirected Hamiltonian cycles in tournaments. *J. Comb. Theory, Ser. B*, 16:234–242, 1974.
646. B. Roy. Nombre chromatique et plus longs chemins d'un graphe. *Rev. Française Informat. Recherche Opérationnelle*, 1(5):129–132, 1967.
647. A. Rubinstein. Ranking the participants in tournaments. *SIAM J. Applied Math.*, 38:108–111, 1980.
648. R. Saad. Finding a longest alternating cycle in a 2-edge-coloured complete graph is in RP. *Combinatorics, Probability and Computing*, 5:297–306, 1996.
649. M.R. Samathan and D.K. Pradhan. The de Bruijn multiprocessor network: a versatile parallel processing and sorting network for VLSI. *IEEE Trans. on Computers*, C-38:567–581, 1989.
650. B.K. Sanyal and M.K. Sen. New characterization of digraphs represented by intervals. *J. Graph Theory*, 22:297–303, 1996.
651. V.I. Sarvanov and N.N. Doroshko. The approximate solution of the traveling salesman problem by a local algorithm that searches neighborhoods of exponential cardinality in quadratic time. In *Software: Algorithms and Programs*, volume 31, pages 8–11. Math. Institute of the Belorussian Acad. Sci., Minsk, 1981. (in Russian).
652. V.I. Sarvanov and N.N. Doroshko. The approximate solution of the traveling salesman problem by a local algorithm with scanning neighborhoods of

factorial cardinality in cubic time. In *Software: Algorithms and Programs*, volume 31, pages 11–13. Math. Institute of the Belorussian Acad. Sci., Minsk, 1981. (in Russian).
653. J.P. Schmidt and A. Siegel. The spatial complexity of oblivious k-probe hash functions. *SIAM J. Computing*, 19(5):775–786, 1990.
654. C.-P. Schnorr. Bottlenecks and edge connectivity in unsymmetrical networks. *SIAM J. Computing*, 8(2):265–274, 1979.
655. A. Schrijver. Total dual integrality from directed graphs, crossing families, and sub- and supermodular functions. In *Progress in combinatorial optimization (Waterloo, Ont., 1982)*, pages 315–361. Academic Press, Toronto, Ont., 1984.
656. A. Schrijver. A group-theoretical approach to disjoint paths in directed graphs. *CWI Quarterly*, 6(3):257–266, 1993.
657. A. Schrijver. Finding k disjoint paths in a directed planar graph. *SIAM J. Computing*, 23(4):780–788, 1994.
658. A. Schrijver. Paths in graphs and curves on surfaces. In *Joseph, A. (ed.) et al., First European congress of mathematics (ECM), Paris, France, July 6-10, 1992. Volume II: Invited lectures (Part 2). Basel: Birkhaeuser.[ISBN 3-7643-2799-5/hbk]*, pages 381–406, 1994.
659. A. Schrijver. Polyhedral combinatorics. In *Handbook of combinatorics, Vol. 1, 2*, pages 1649–1704. Elsevier, Amsterdam, 1995.
660. A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. manuscript 1999.
661. M. Sen, S. Das, A.B. Roy, and D.B. West. Interval digraphs: an analogue of interval graphs. *J. Graph Theory*, 13(2):189–202, 1989.
662. P.D. Seymour. Disjoint paths in graphs. *Discrete Math.*, 29:293–309, 1980.
663. P.D. Seymour. Nowhere-zero 6-flows. *J. Combin. Theory Ser. B*, 30(2):130–135, 1981.
664. P.D. Seymour. Nowhere-zero flows. In *Handbook of combinatorics, Vol. 1, 2*, pages 289–299. Elsevier, Amsterdam, 1995. Appendix: Colouring, stable sets and perfect graphs.
665. P.D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.
666. P.D. Seymour. Packing circuits in Eulerian digraphs. *Combinatorica*, 16(2):223–231, 1996.
667. P.D. Seymour and C. Thomassen. Characterization of even directed graphs. *J. Combin. Theory Ser. B*, 42(1):36–45, 1987.
668. J. Shearer. A property of the colored complete graph. *Discrete Math.*, 25:175–178, 1979.
669. Y. Shiloach. Edge-disjoint branching in directed multigraphs. *Inform. Process. Lett.*, 8(1):24–27, 1979.
670. Y. Shiloach. A polynomial solution to the undirected two paths problem. *J. Assoc. Comput. Mach.*, 27:445–456, 1980.
671. D. Shmoys. Cut problems and their application to divide and conquer. In *Approximation algorithms for NP-hard problems, D. Hochbaum ed.*, pages 192–235. PWS publishing company, Boston, 1997.
672. K. Simon. An improved algorithm for transitive closure on acyclic digraphs. *Theoret. Comput. Sci.*, 58:325–346, 1988.
673. K. Simon. Finding a minimal transitive reduction in a strongly connected digraph within linear time. In *Graph-theoretic concepts in Computer Science (Kerkrade, 1989)*, Lecture Notes in Computer Science, pages 245–259. Springer-Verlag, Berlin, 1990.
674. S.S. Skiena. *The Algorithm Design Manual*. Springer-Verlag, New York, 1997.

675. D.J. Skrien. A relationship between triangulated graphs, comparability graphs, proper interval graphs, proper circular-arc graphs, and nested interval graphs. *J. Graph Theory*, 6(3):309–316, 1982.
676. L. Šoltés. Orientations of graphs minimizing the radius or the diameter. *Math. Slovaca*, 36(3):289–296, 1986.
677. Z.M. Song. Complementary cycles in bipartite tournaments. *J. Nanjing Inst. Tech.*, 18:32–38, 1988.
678. Z.M. Song. Complementary cycles of all lengths in tournaments. *J. Combin. Theory Ser. B*, 57(1):18–25, 1993.
679. Z.M. Song. Pancyclic oriented graphs. *J. Graph Theory*, 18(5):461–468, 1994.
680. E. Sopena. The chromatic number of oriented graphs. *J. Graph Theory*, 25(3):191–205, 1997.
681. E. Speckenmeyer. On feedback problems in digraphs. In *Proceedings of the 15th WG* , volume 411 of *Lecture Notes in Comput. Sci.*, pages 218–231. Springer, Berlin, 1989.
682. G. Steiner. A compact labeling scheme for series-parallel graphs. *Discrete Appl. Math.*, 11(3):281–297, 1985.
683. H.J. Straight. The existence of certain type of semi-walks in tournaments. *Congressus Numerantium.*, 29:901–908, 1980.
684. X.Y. Su. Paths, cycles, and arc-connectivity in digraphs. *J. Graph Theory*, 19(3):339–351, 1995.
685. J. Szigeti and Z. Tuza. Generalized colorings and avoidable orientations. *Discuss. Math. Graph Theory*, 17(1):137–145, 1997.
686. J.L. Szwarcfiter. On minimum cuts of cycles and maximum disjoint cycles. In *Graphs and algorithms (Boulder, CO, 1987)*, volume 89 of *Contemp. Math.*, pages 153–166. Amer. Math. Soc., Providence, RI, 1989.
687. É. Tardos. A strongly polynomial minimum cost circulation algorithm. *Combinatorica*, 5(3):247–255, 1985.
688. R.E. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Computing*, 1(2):146–160, 1972.
689. R.E. Tarjan. Finding optimum branchings. *Networks*, 7(1):25–35, 1977.
690. R.E. Tarjan. *Data structures and network algorithms.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa., 1983.
691. E.G. Tay. *Optimal orientations of graphs.* PhD thesis, National University of Singapore, Department of Mathematics, 1999.
692. M. Tewes. *In-tournaments and Semicomplete Multipartite Digraphs.* PhD thesis, Lehrstuhl II für Mathematik, RWTH Aachen, 1998.
693. M. Tewes and L. Volkmann. The ratio of the longest cycle and longest path in semicomplete multipartite digraphs. RTWH Aachen.
694. A. Thomason. Paths and cycles in tournaments. *Trans. Amer. Math. Soc.*, 296(1):167–180, 1986.
695. S. Thomassé. Covering a strong digraph by $\alpha - 1$ disjoint paths. A proof of Las Vergnas's conjecture. manuscript 2000.
696. C. Thomassen. An Ore-type condition implying a digraph to be pancyclic. *Discrete Math.*, 19(1):85–92, 1977.
697. C. Thomassen. 2-linked graphs. *European J. Combinatorics*, 1:371–378, 1980.
698. C. Thomassen. Hamiltonian-connected tournaments. *J. Combin. Theory Ser. B*, 28(2):142–163, 1980.
699. C. Thomassen. Edge-disjoint Hamiltonian paths and cycles in tournaments. *Proc. London Math. Soc. (3)*, 45(1):151–168, 1982.
700. C. Thomassen. Disjoint cycles in digraphs. *Combinatorica*, 3(3-4):393–396, 1983.

701. C. Thomassen. Connectivity in tournaments. In *Graph theory and combinatorics (Cambridge, 1983)*, pages 305–313. Academic Press, London, 1984.
702. C. Thomassen. Even cycles in directed graphs. *European J. Combin.*, 6(1):85–89, 1985.
703. C. Thomassen. Hamilton circuits in regular tournaments. In *Cycles in graphs (Burnaby, B.C., 1982)*, volume 115 of *North-Holland Math. Stud.*, pages 159–162. North-Holland, Amsterdam, 1985.
704. C. Thomassen. The 2-linkage problem for acyclic digraphs. *Discrete Math.*, 55(1):73–87, 1985.
705. C. Thomassen. Sign-nonsingular matrices and even cycles in directed graphs. *Linear Algebra Appl.*, 75:27–41, 1986.
706. C. Thomassen. Counterexamples to Adám's conjecture on arc reversals in directed graphs. *J. Combin. Theory Ser. B*, 42(1):128–130, 1987.
707. C. Thomassen. Paths, circuits and subdivisions. In *Selected topics in graph theory Vol. 3*, pages 97–131. Academic Press Limited, 1988.
708. C. Thomassen. Configurations in graphs of large minimum degree, connectivity, or chromatic number. *Annals of the New York Academy of Sciences*, 555:402–412, 1989.
709. C. Thomassen. Whitney's 2-switching theorem, cycle spaces, and arc mappings of directed graphs. *J. Combin. Theory Ser. B*, 46(3):257–291, 1989.
710. C. Thomassen. Highly connected non-2-linked digraphs. *Combinatorica*, 11(4):393–395, 1991.
711. C. Thomassen. The even cycle problem for directed graphs. *J. Amer. Math. Soc.*, 5(2):217–229, 1992.
712. C. Thomassen. The even cycle problem for planar digraphs. *J. Algorithms*, 15:61–75, 1993.
713. C. Thomassen. Directed cycles with two chords and strong spanning directed subgraphs with few arcs. *J. Combin. Theory Ser. B*, 66(1):24–33, 1996.
714. C. Thomassen. Parity, cycle space and K_4-subdivisions in graphs. In *Surveys in Combinatorics, J.D. Lamb and D.A. Preece, eds.*, London Math. Soc. Lect. Note Series, 267, pages 223–237. Cambridge Univ. Press, Cambridge, UK, 1999.
715. M. Thorup. On RAM priority queues. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 59–67, New York, 1996. ACM Press.
716. M. Thorup. Undirected single-source shortest paths with positive integer weights in linear time. *Journal of the Association for Computing Machinery*, 46:362–394, 1999.
717. K. Thulasiraman and M.N.S. Swamy. *Graphs: theory and algorithms.* John Wiley & Sons Inc., New York, 1992.
718. F. Tian, Z.S. Wu, and C.Q. Zhang. Cycles of each length in tournaments. *J. Combin. Theory Ser. B*, 33(3):245–255, 1982.
719. T.W. Tillson. A Hamiltonian decomposition of $\overset{*}{K}_{2m}$, $2m \geq 8$. *J. Combin. Theory Ser. B*, 29(1):68–74, 1980.
720. W.T. Tutte. A contribution to the theory of chromatic polynomials. *Canadian J. Math.*, 6:80–91, 1954.
721. W.T. Tutte. A theorem on planar graphs. *Trans. Amer. Math. Soc.*, 82:99–116, 1956.
722. W.T. Tutte. On the problem of decomposing a graph into n connected factors. *J. London Math. Soc.*, 36:221–230, 1961.
723. Z. Tuza. Graph coloring in linear time. *J. Combin. Theory Ser. B*, 55(2):236–243, 1992.
724. Z. Tuza. Characterization of $(m, 1)$-transitive and $(3, 2)$-transitive semicomplete directed graphs. *Discrete Math.*, 135(1-3):335–347, 1994.

725. J. Urrutia and F. Gavril. An algorithm for fraternal orientation of graphs. *Inform. Process. Lett.*, 41(5):271–274, 1992.
726. J. Valdes, R.E. Tarjan, and E.L. Lawler. The recognition of series parallel digraphs. *SIAM J. Computing*, 11(2):298–313, 1982.
727. L.M. Vitaver. Determination of minimal coloring of vertices of a graph by means of Boolean powers of the incidence matrix. *Dokl. Akad. Nauk SSSR*, 147:758–759, 1962.
728. L. Volkmann. Cycles in multipartite tournaments: results and problems. *Discrete Math.*, to appear.
729. L. Volkmann. Longest paths in semicomplete multipartite digraphs. *Discrete Math.*, 199:279–284, 1999.
730. L. Volkmann. Spanning multipartite tournaments of semicomplete multipartite digraphs. *ARS Combinatoria*, to appear.
731. J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behaviour*. Princeton University Press, Princeton, 1944.
732. C. Wang, E.L. Floyd, and M.L. Soffa. Feedback vertex sets and cyclically reducible graphs. *J. ACM*, 32:296–313, 1985.
733. H. Wang, C. Little, and K. Teo. Partition of a directed bipartite graph into two directed cycles. *Discrete Math.*, 160(1-3):283–289, 1996.
734. S. Warshall. A theorem on boolean matrices. *J. ACM*, 9:11–12, 1962.
735. D.B. West. *Introduction to graph theory*. Prentice-Hall, 1996.
736. D.B. West. Short proofs for interval digraphs. *Discrete Math.*, 178(1-3):287–292, 1998.
737. R.J. Wilson. An Eulerian trail through Königsberg. *J. Graph Theory*, 8:265–275, 1986.
738. D. Woodall. Improper colourings of graphs. In *Graph Colourings*, volume 218 of *Pitman Research Notes in Math. Series*, pages 45–63. Longman, 1990.
739. D.R. Woodall. Sufficient conditions for cycles in digraphs. *Proc. London Math. Soc.*, 24:739–755, 1972.
740. N.C. Wormald. Subtrees of large tournaments. *Lecture Notes in Mathematics*, 1036:417–419, 1983.
741. Z.S. Wu, K.M. Zhang, and Y. Zou. A necessary and sufficient condition for arc-pancyclicity of tournaments. *Sci. Sinica Ser. A*, 25:249–254, 1982.
742. J.M. Xu. A sufficient condition for equality of arc-connectivity and minimum degree of a digraph. *Discrete Math.*, 133(1-3):315–318, 1994.
743. A. Yeo. A note on alternating cycles in edge-coloured graphs. *J. Combin. Theory Ser. B*, 69:222–225, 1997.
744. A. Yeo. One-diregular subgraphs in semicomplete multipartite digraphs. *J. Graph Theory*, 24(2):175–185, 1997.
745. A. Yeo. *Semicomplete Multipartite Digraphs*. PhD thesis, Department of Mathematics and Computer Science, Odense University, Denmark, 1998.
746. A. Yeo. A polynomial time algorithm for finding a cycle covering a given set of vertices in a semicomplete multipartite digraph. *J. Algorithms*, 33(1):124–139, 1999.
747. A. Yeo. Diregular c-partite tournaments are vertex-pancyclic when $c \geq 5$. *J. Graph Theory*, 32:137–152, 1999.
748. A. Yeo. How close to regular must a semicomplete multipartite digraph be to secure Hamiltonicity? *Graphs Combin.*, 15:481–493, 1999.
749. A. Yeo. Large diregular 4-partite tournaments are vertex-pancyclic. manuscript 1999.
750. D.H. Younger. Graphs with interlinked directed circuits,. In *Proceedings of the Midwest symposium on circuit theory 2*, pages XVI 2.1 – XVI 2.7, 1973.
751. D.H. Younger. Integer flows. *J. Graph Theory*, 7(3):349–357, 1983.

752. D.A. Youngs. Minimal orientations of colour critical graphs. *Combinatorica*, 15(2):289–295, 1995.
753. N. Zadeh. Theoretical efficiency of the Edmonds-Karp algorithm for computing maximal flows. *J. Assoc. Comput. Mach.*, 19:184–192, 1972.
754. C.Q. Zhang. Every regular tournament has two arc-disjoint hamiltonian cycles. *J. Qufu Normal College*, Special Issue Oper. Res.:70–81, 1980.
755. K.-M. Zhang. Vertex even-pancyclicity in bipartite tournaments. *J. Nanjing Univ. Math. Biquarterly*, 1:85–88, 1981.
756. K.M. Zhang, Y. Manoussakis, and Z.M. Song. Complementary cycles containing a fixed arc in diregular bipartite tournaments. *Discrete Math.*, 133(1-3):325–328, 1994.
757. K.M. Zhang and J.-Z. Wang. Complementary cycles containing a fixed arc and a fixed vertex in bipartite tournaments. *Ars Combin.*, 35:265–269, 1993.
758. L.-C. Zhao and J.-H. Meng. A sufficient condition for hamiltonian cycles in digraphs. *Ars Combinatoria*, 32:335–338, 1991.
759. G. Zhou, T. Yao, and K.-M. Zhang. On score sequences of k-hypertournaments. *Europ. J. Combin.*, to appear, 2000.
760. G. Zhou and K. Zhang. A sufficient condition for a semicomplete multipartite digraph to be hamiltonian. *Australas. J. Combin.*, 19:231–234, 1999.
761. H. Zhou. Multiplicativity of acyclic digraphs. *Discrete Math.*, 176(1-3):263–271, 1997.
762. H.S. Zhou. Characterization of the homomorphic preimages of certain oriented cycles. *SIAM J. Discrete Math.*, 6(1):87–99, 1993.

Symbol Index

To shorten and unify notation, in this index we use the following convention:
B denotes a bipartite (di)graph.
C, C_i denote cycles (directed, undirected, edge-coloured, oriented).
D, D_i denote digraphs, directed multigraphs and directed pseudographs.
G, G_i denote undirected graphs and undirected multigraphs.
H denotes a hypergraph.
M denotes a mixed graph or a matroid.
P, P_i denote path (directed, undirected, edge-coloured, oriented).
S denotes a matrix or a multiset.
X, X_i denote abstract sets or sets of vertices.
Y, Y_i denote sets of arcs.

Author Index

Subject Index