Some Parameterized Problems on Digraphs

Gregory Gutin[*] Anders Yeo[†]

Abstract

We survey known results on parameterized complexity of the feedback set and induced subdigraph problems for digraphs. We prove new results on some parameterizations of the paired comparison problems on digraphs. One of our theorems implies a new result for a parameterized version of the linear arrangement problem for undirected graphs. We state several open problems.

1 Introduction

It is well-known that directed graphs are at least as important to theoretical computer science and its applications as their undirected counterparts. Yet, while there is a large number of papers on fixed-parameter algorithmics for undirected graphs, only very few papers were published so far on parameterized problems for digraphs. There are several reasons for this seemingly strange situation including the following: many problems which can be formulated for both directed and undirected graphs are significantly more difficult for digraphs, we know much less about the structure of digraphs than about the structure of undirected graphs, and many more graph theory papers deal with undirected graphs than with digraphs. We believe that the situation should be changed, and we view this paper as a small step towards this goal.

In this paper, we deal only with a small number of parameterized problems on digraphs. We survey some results on the feedback vertex and arc set problems, and the induced subdigraph problem. We prove new results on the backward (and forward) paired comparison problems. We state several open parameterized problems for digraphs.

The paper is organized as follows. In the rest of this section, we provide some basic terminology and notation. In Section 2, we describe some motivation and results on the

[*] Corresponding author. Department of Computer Science, Royal Holloway University of London, Egham, Surrey TW20 OEX, UK, gutin@cs.rhul.ac.uk and Department of Computer Science, University of Haifa, Israel

[†] Department of Computer Science, Royal Holloway University of London, Egham, Surrey TW20 OEX, UK, anders@cs.rhul.ac.uk

classical feedback arc and vertex set problems. Section 3 is devoted to the parameterized feedback arc and vertex set problems in digraphs, whose complexity is still not known. We overview some partial results and approaches to the problems. In particular, we indicate that an approach sufficient to prove that the feedback vertex set problem is fixed-parameter tractable for undirected graphs appears too weak to settle the feedback vertex set problem for digraphs. In Section 4, we introduce the method of paired comparisons and formulate the forward and backward paired comparison problems (FPCP and BPCP). In Section 5, we prove that a natural parametrization of BPCP is fixed-parameter tractable and raise a question of establishing the complexity of a more 'relaxed' parametrization of BPCP. In Section 6, we consider an alternative parametrization of BPCP using so-called distance from triviality. We prove that the alternative parametrization is fixed-parameter tractable. This result is applied to the linear arrangement problem (on undirected graphs) in Section 7. In Section 8, we give a short overview of results on the induced subdigraph problem obtained in [40]. We conclude by stating some open parameterized problems for digraphs.

We provide only very basic terminology of parameterized complexity, for an in-depth treatment of the topic we refer the reader to the classic text [12] and the new monographs [15, 36]. A parameterized problem Π can be considered as a set of pairs (I, k) where I is the *problem instance* and k (usually an integer) is the *parameter*. Π is called *fixed-parameter tractable (FPT)* if membership of (I, k) in Π can be decided in time $O(f(k)|I|^c)$, where $|I|$ is the size of I, $f(k)$ is a computable function, and c is a constant independent from k and I. An algorithm A for an FPT problem is *FPT* if its time complexity is $O(f(k)|I|^c)$.

We consider directed and undirected graphs with no loops and parallel arcs and edges. A *cycle* is a simple (no self-crossings) directed cycle in digraphs and a simple undirected cycle in undirected graphs. A *p-cycle* is a cycle with p vertices. If xy is an arc in a digraph D, then y is an *out-neighbor* of x and x is an *in-neighbor* of y. The number of out-neighbors (in-neighbors) of a vertex z is its *out-degree* (*in-degree*); it is denoted by $d^+(z)$ $(d^-(z))$.

2 Feedback Arc and Vertex Set Problems

In a digraph D, a set S of vertices (arcs) is a *feedback vertex set* (a *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 [34]). An electronic circuit can be modelled by a directed graph by letting each (boolean) gate correspond to a vertex and the wires into each gate be modelled 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 [33]).

The classical decision problems Feedback Vertex Set (FVS) and Feedback Arc Set

(FAS) can be stated as follows.

FVS
Instance: A digraph D and a positive integer k.
Question: Is $\tau_0(D) \leq k$?

FAS
Instance: A digraph D and a positive integer k.
Question: Is $\tau_1(D) \leq k$?

The following result shows that FVS and FAS are of the same complexity up to a polynomial factor. The *line digraph* $L(D)$ of a digraph D is a digraph H with $V(H) = A(D)$ and in which there is an arc from $xy \in V(H)$ to $uv \in V(H) \setminus \{xy\}$ in H if $y = u$. A digraph H is a *line digraph* if there is a digraph D such that $H = L(D)$.

Proposition 2.1 *[4] For every digraph D there exist digraphs D' and D'' such that $\tau_0(D) = \tau_1(D')$ 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 the line digraph of D. To construct D' replace every vertex x of D with two new vertices x^- and x^+ and replace the arcs of D by the following: arc x^-x^+ for each vertex $x \in V(D)$ and x^+y^- for each arc $xy \in A(D)$. It is not difficult to verify that the equalities of this proposition indeed hold. $\diamond$

Karp [29] was the first to prove that FAS is $\mathcal{NP}$-complete. Proposition 2.1 and Karp's result imply immediately that FVS is also $\mathcal{NP}$-complete. For a digraph D, let $\Delta^0(D)$ be the minimum integer such that every vertex x of D has in-degree and out-degree at most $\Delta^0(D)$. Gavril [17] proved that FAS remains $\mathcal{NP}$-complete even for digraphs D with $\Delta^0(D) \leq 3$ and for line digraphs. Bang-Jensen and Thomassen [5] proved that FVS is $\mathcal{NP}$-complete even for tournaments. FVS remains $\mathcal{NP}$-complete for digraphs D with $\Delta^0(D) \leq 2$ and for planar digraphs D with $\Delta^0(D) \leq 3$ [16]. This problem, unlike FAS, is $\mathcal{NP}$-complete even for undirected graphs [16]. Bang-Jensen and Thomassen [5] conjectured that FAS is $\mathcal{NP}$-complete even for tournaments. This conjecture was proved independently by at least four groups of researchers [1, 2, 8, 9]. Interestingly, FAS is polynomial time solvable for planar digraphs [4, 35] and trivially polynomial time solvable for undirected graphs.

3 Parameterized FVS and FAS

The natural parameterizations of FVS and FAS are formulated as follows.

Parameterized FVS
Instance: A digraph D.
Parameter: A positive integer k.
Question: Is $\tau_0(D) \leq k$?

Parameterized FAS
Instance: A digraph D.
Parameter: A positive integer k.
Question: Is $\tau_1(D) \leq k$?

By Proposition 2.1, these problems are either both FPT, or are both not. So far, the parameterized complexity remains a (well-known) open problem [12]. Many researchers believe that both problems are FPT. Fellows et al. [13] introduce a parameterized version of the shortest common sequence problem such that if FVS is not FPT, then the problem in [13] is not FPT either.

However, FVS was proved to be FPT for some special classes of digraph as well as for undirected graphs. We describe the main idea behind the following FPT algorithm for undirected graphs from [38]. This idea, which we call the *shortest cycle approach*, is used by many FPT algorithms for FVS for both directed and undirected graphs. Let $G = (V, E)$ be an undirected graph. Clearly, we may delete a vertex of degree 1 from G without changing any solution of FVS. If G has a vertex x of degree 2, then either G has a 3-cycle through x (in which case we have a short cycle of G) or we can replace the edges yx, zx incident to x by yz without creating parallel edges and without changing $\tau_0(G)$. If the minimum degree of a vertex in G is at least 3, then, by a theorem of Erdős and Posa [24], G has a cycle of length at most $2\lg|V| + 1$.

This suggests the following algorithm first considered in [38]: transform an input graph $G = (V, E)$ such that either G has a cycle C of length 3 or the minimum degree of G is a least 3. In the last case, find a cycle C of length at most $2\lg|V| + 1$ using a shortest cycle algorithm (there is such an algorithm of complexity $O(n^{\omega})$ [25], the complexity of multiplying two boolean matrices of size $n \times n$). For each $v \in V(C)$, run the algorithm recursively with input $G - v$ and parameter $k - 1$ (rather than k). Observe that the running time of this algorithm is $O((2\lg|V| + 1)^k n^{\omega})$. Since $(\lg|V|)^k \leq |V| + (3k\lg k)^k$, FVS is FPT for undirected graphs. Note that there are faster FPT algorithms for FVS for undirected graphs [10, 19] that run in time $O(c^k n^{O(1)})$ for some constant c.

The shortest cycle approach can be used for *tournaments*, i.e., digraphs in which there is exactly one arc between every pair of distinct vertices. Indeed, a tournament with a cycle through a vertex x, has a cycle of length 3 through x [4]. Such a cycle can be found by first running a strong component algorithm for a tournament $T = (V, A)$ (with running time $O(|V|^2)$) to find a vertex x in a non-trivial strong component of T (if one exists) and then checking whether T has a 3-cycle with vertex set $\{x, y, z\}$ for each pair $y \neq z \in V - \{x\}$ (overall $O(|V|^2)$ time). This implies that FVS has an FPT algorithm of complexity $O(3^k|V|^2)$. A faster algorithm was designed in [39]. The current fastest FPT algorithm is given in [11] and its time complexity is $O(2^k|V|^2(\lg|V| + k))$.

Certainly, the shortest cycle approach can be used for the whole class of *multipartite tournaments*, i.e., directed graphs obtained from complete multipartite graphs by orienting all their edges. Let C be a shortest cycle through a vertex x of a strongly connected multipartite tournament. Observe that C is of length 3 or 4. Such a cycle can be found in time $O(|V|^3)$ using a procedure similar to the one described above, which gives an $O(4^k|V|^3)$-algorithm. Another approach brings the complexity down. In particular, for bipartite tournaments, Truß [44] noted that a result on hitting sets in hypergraphs by Fernau [14] implies an $O(3.12^k + |V|^{O(1)})$-algorithm.

We will give a very brief outline of the $O(3.12^k + |V|^{O(1)})$-algorithm for bipartite tournaments. Let T be any bipartite tournament and let H_T be the hypergraph with vertex set $V(H_T) = V(T)$ and edge set $E(H_T) = \{V(C) :\ C \text{ is a 4-cycle in } T\}$. Observe that if X is a hitting set in H_T (that is, X intersects every edge in H_T), then $T - X$ is acyclic. In fact it is not difficult to show that a minimum hitting set in H_T is exactly the minimum set X such that $T - X$ is acyclic. A hypergraph H is *rank-m* if the number of vertices in each edge of H does not exceed m. The result in [14] states that it can be decided whether an n-vertex rank-4 hypergraph has a hitting set of size at most k in $O(3.12^k + n^{O(1)})$ time. This implies the algorithm. Note that the exactly same algorithm also works for multipartite tournaments, if we consider edges to be the vertex sets of all 3-cycles and all 4-cycles.

Since the current best fixed-parameter algorithms for FVS in bipartite tournaments and multipartite tournaments use Fernau's hypergraph algorithm for rank-4 hypergraphs, it would be interesting to improve the running time of the algorithm. However, as the hypergraphs produced above have a special structure, it is plausible that there exist algorithms for FVS in bipartite tournaments and multipartite tournaments that are faster than those for the minimum hitting problem on rank-4 hypergraphs.

A digraph D is *locally semicomplete* if, for each vertex x of D, every pair of out-neighbors of x are linked by at least one arc and every pair in-neighbors of x are linked by at least one arc. Locally semicomplete digraphs constitute another class of digraphs for which it is easy to prove that FVS is FPT. An easy way to see this (suggested by Kim [30]) is to apply the following theorem by Bang-Jensen [3].

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, the out-neighbors of v_i are $v_{i+1}, v_{i+2} \ldots, v_{i+d^+(v_i)}$ and the in-neighbors of v_i are $v_{i-d^-(v_i)}, v_{i-d^-(v_i)+1}, \ldots, v_{i-1}$ (all subscripts are taken modulo n).

Theorem 3.1 *Let D be a connected locally semicomplete digraph with no 2-cycles. Then D is round if and only if, for each vertex x of D, there is no cycle formed by out-neighbors of x and there is no cycle formed by in-neighbors of x.*

Let D be a locally semicomplete digraph. If D has a 2-cycle or 3-cycle, we can use the shortest cycle approach. Otherwise, D is round by Bang-Jensen's theorem. If D is not strongly connected, it does not have cycles. If D is strongly connected, we can find a vertex x with minimum out-degree and a vertex y with minimum in-degree. The out-neighbors of x form a minimum feedback vertex set if $d^+(x) \le d^-(y)$ and the in-neighbors of y form a minimum minimum feedback vertex, otherwise.

Using the shortest cycle approach one can prove that FAS is FPT for multipartite tournaments, but we need a tool that allows us not to delete arcs (otherwise, we may move outside of the class of multipartite tournaments). This tool is the following simple proposition (its part (ii)).

An *ordering* of a digraph $D = (V, A)$ is a bijection $\alpha : V \rightarrow \{1, 2, \dots, |V|\}$. An arc $uv \in A$ is *forward* (*backward*) if $\alpha(u) < \alpha(v)$ $(\alpha(u) > \alpha(v))$.

Proposition 3.2 *Let D be a digraph. Then $\tau_1(D)$ equals*

(i) the minimum number of backward arcs in an ordering of D;

(ii) the minimum number of arcs in D whose reorientation makes D acyclic.

Let $h(n)$ be a function with $\lim_{n\to\infty} h(n) = 0$. Let $\mathcal{S}_h$ be the family of digraphs with n vertices and m arcs in which $m \ge \binom{n}{2} - n^{1+h(n)}$. Using the shortest cycle approach, we can prove the following:

Theorem 3.3 *FAS is FPT for digraphs in $\mathcal{S}_h$.*

We provide only a scheme of the proof of Theorem 3.3 in [39]. We will use the following:

Lemma 3.4 *[6] Let $l \ge 2$ be any integer and let H be a strongly connected digraph with n vertices and m arcs such that*

$$m \ge \frac{n^2 + (3-2l)n + (l^2 - l)}{2}.$$

Then the length of a shortest cycle in H is at most l.

Using this lemma it is not difficult to prove that the length of a shortest cycle C in a digraph D of $\mathcal{S}_h$ is at most $n^{g(n)}$ for some function $g(n) = o(1)$. By Part (ii) of Proposition 3.2, D has a FAS of size at most k if and only if $(D+yx)-xy$ has a FAS of size at most $k-1$ for at least one arc xy of C. This observation implies an $O((n^{g(n)})^k n^{O(1)})$-time algorithm. We will show that $(n^{g(n)})^k \le n + f(k)$ for some function $f(k)$. For every k, let $n(k) \ge 2$ be an integer such that $g(n) \le \frac{1}{k}$ for all $n \ge n(k)$. Then $(n^{g(n)})^k \le n$ for all $n \ge n(k)$. Now let $f(k) = \max\{(n^{g(n)})^k : \ n < n(k)\}$, and note that $f(k)$ is a well-defined function as the maximum is taken over a finite number of values. Furthermore, this implies that $(n^{g(n)})^k \le f(k)$ for all $n < n(k)$. Thus, $(n^{g(n)})^k \le n + f(k)$ and FVS is FPT for $\mathcal{S}_h$.

The shortest cycle approach appears too weak for digraphs. Indeed, there is no digraph analog the Erdős-Posa Theorem. In fact, the following simple example shows that no such theorem is valid for digraphs. The mth *power of a digraph* D is a digraph D^m with the same vertex set as D and in which xy is an arc if and only if there is an (x, y)-path in D of length at most m. Consider an n-cycle $\vec{C}_n$ as a digraph. Clearly, in the digraph $\vec{C}_n^m$, the in-degree and out-degree of every vertex is m yet the shortest cycle is of length at least n/m. Thus, if FVS and FAS are FPT for digraphs, new ideas that complement the shortest cycle approach or can be applied on their own are required. It is worth noting that recent improvements in FPT algorithms for FVS in undirected graphs were achieved using so-called iterative compression, see [10, 19].

In absence of a parameterized complexity result for FVS and FAS on all digraphs, parameterized complexity of FVS and FAS restricted to some special classes of digraphs should be investigated. Some examples of such special classes are planar digraphs, locally in-semicomplete digraphs (a digraph is *locally in-semicomplete* if, for each vertex x, every pair of in-neighbors of x are linked by at least one arc) and line digraphs (a *line digraph* is the line digraph of some digraph; the *line digraph* $L(D)$ of a digraph D is a digraph with vertex set equal the arc set of D, in which a vertex uv of $L(D)$ is an in-neighbor of a vertex xy of $L(D)$ if $v = x$).

We finish this section with the following graph-theoretical result potentially useful in establishing parameterized complexity of FVS and FAS. Younger [45] conjectured that for every k, there 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 2.1, the validity of the 'vertex' version of Younger's conjecture implies that the 'arc' version holds and vice versa. Moreover, Proposition 2.1 implies that, if the functions $t_0(k)$ and $t_1(k)$ exist, then they are equal. Younger's conjecture was completely settled by Reed, Robertson, Seymour and Thomas [41].

4 Methods of Paired Comparison

Proposition 3.2 shows that the minimal size of a feedback arc set in a digraph D equals the minimal number of backward arcs in an ordering of D. In Proposition 3.2, we view the 'length' of every arc as 1. In some cases, the length can be different for different arcs depending on the ordering. We consider such a case in the next three sections.

In the well-known method of paired comparisons, which has numerous applications [31, 42, 43], we first compare objects of a set S pairwise (the comparison process is often performed by experts) and consequently produce an optimal ordering of S. In general, the outcome of the process of paired comparisons can be modelled by the following weighted digraphs. 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) $0 < \epsilon(xy) \le 1$ 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 $yx \notin A$.

The weight $\epsilon(xy)$ indicates the relative preference of x over y.

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 *score* of a vertex $x \in V$ is

$$\sigma^{+}(x) = \sum_{xy \in A} \epsilon(xy).$$

We describe two methods (forward and backward) of ordering V introduced by Kano and Sakamoto [27, 28]. Notice that, for semicomplete digraphs (digraphs that can be obtained from complete graphs by replacing every edge xy with the arc xy or the arc yx or both xy and yx), these methods agree with the score method (the larger the score the earlier the vertex in an optimal ordering).

Recall that an ordering of a digraph $D = (V, A)$ is a bijection $\alpha : V \to \{1, 2, \ldots, |V|\}$. An arc $uv \in A$ is forward (backward) if $\alpha(u) < \alpha(v)$ ($\alpha(u) > \alpha(v)$).

The *length* $\lambda_\alpha(uv)$ of an arc $uv \in A$ is $\epsilon(uv)|\alpha(u) - \alpha(v)|$. The *forward (backward) length* $f_D(\alpha)$ ($b_D(\alpha)$) of α is the sum of the lengths of all forward (backward) arcs. An ordering α is *forward (backward) optimal* if the corresponding length $f_D(\alpha)$ ($b_D(\alpha)$) is maximum (minimum) over all orderings of D. The *forward and backward paired comparison problems (FPCP and BPCP)* are to find an optimal forward and backward ordering of a PCD D. Since treatment of FPCP is often similar to that of BPCP, we restrict ourself mostly to BPCP in the next two sections.

5 Natural Parametrization of BPCP

The weights $\epsilon(xy)$ in a PCD D reflects the relative preference of x over y. This preference is normally obtained after some averaging of individual preferences by several experts.

This and the impossibility to have irrational numbers in computer memory, allows us to assume, in the following problem, that each weight $\epsilon(xy)$ is a rational number proportional to $1/q$, where q is a positive integer.

BPCP(k,q)
Instance: A PCD D with weights proportional to $1/q$.
Parameters: Positive integer q and nonnegative integer k.
Question: Does D have an ordering of backward length at most k?

Theorem 5.1 *BPCP*(k,q) *is FPT.*

Proof: Let $D_1, D_2, \dots, D_r$ be an acyclic ordering of the strongly connected components of D, i.e., there is no arc from D_j to D_i for any pair i,j with $j > i$. Both the components and their acyclic ordering can be found in time $O(|A(D)|)$ [4]. Observe that if we find an optimal backward ordering α_i of each D_i, then we can get the following optimal backward ordering of D: $\alpha(x) = \alpha_i(x) + \sum_{k=1}^{i-1} |V(D_k)|$ for every $x \in V(D)$. If each component D_i consists of a single vertex, the answer to the questions in BPCP(k,q) is obviously 'Yes'. So, we may assume that D has a nontrivial component $H = D_i$.

We prove that, for each ordering β of H, we have $b_D(\beta) \geq \frac{p-1}{q}$, where $p = |V(H)|$. Construct a sequence of backward arcs of β as follows: let $i_1 = p$ and j_1 be the minimal integer j for which $\beta^{-1}(i_1)\beta^{-1}(j)$ is an arc in H (j_1 must exist as H is strongly connected). Let j_2 be the minimal integer s for which there is an arc in H from a vertex $\beta^{-1}(l)$ to $\beta^{-1}(s)$, where $j_1 \leq l \leq i_1 - 1$ (again the existence of such an arc is due to strong connectivity of D). Let i_2 be an arbitrary integer such that $j_1 \leq i_2 \leq i_1 - 1$ and $\beta^{-1}(i_2)\beta^{-1}(j_2) \in A(H)$. We continue finding arcs $\beta^{-1}(i_a)\beta^{-1}(j_a)$ till we get $j_a = 1$. Clearly, all arcs $\beta^{-1}(i_a)\beta^{-1}(j_a)$, $a = 1, 2, \dots, b$ are backward and the sum of their lengths is

$$\sum_{a=1}^{b} \epsilon(\beta^{-1}(i_a)\beta^{-1}(j_a))(i_a - j_a) \geq \frac{1}{q}\sum_{a=1}^{b}(i_a - j_a) \geq \frac{p-1}{q}.$$

Assume, without loss of generality, that $D_1, D_2, \dots, D_t$ are the only non-trivial components of D; let $p_i = |V(D_i)|$. It follows from the above arguments that, for each ordering γ of D, we can construct a sequence of backward arcs of total length at least $l = \sum_{i=1}^{t} \frac{p_i - 1}{q}$. If $l > k$, then the answer to the question in BPCP(k,q) is obviously 'No'. If $l \leq k$, then let $c = \sum_{i=1}^{t} p_i$ and note that $c \leq 2kq$. Now observe that we can find an optimal backward ordering of the subgraph of D induced by $V(D_1) \cup \ldots \cup V(D_t)$ in time $O(c!c^2) = O((2kq+2)!)$. Combining the optimal ordering with the trivial ordering of the vertices of the trivial components of D, we obtain an optimal backward ordering of D. $\diamond$

We were unable to determine parameterized complexity of the following 'relaxation' of BPCP(k,q):

BPCP(k)
Instance: A PCD D.
Parameter: A nonnegative integer k.
Question: Does D have an ordering of backward length at most k?

6 Alternative Parametrization of BPCP

The natural parametrization of the previous section is of interest only for PCDs close to acyclic. Indeed, k can be considered as an indicator of the 'distance' from a given PCD to the closest acyclic digraph. So, k can be viewed as a 'distance from triviality' in the terminology of [20]. We would rather call it 'distance from tractability.' In this section, we consider another distance from tractability, which is of interest for PCDs that may be far from acyclic digraphs.

Let $D = (V, A, \epsilon)$ be a PCD. A digraph D is *semicomplete multipartite* if D is obtained from a complete multipartite graph G by replacing every edge xy of G with the arc xy or the arc yx or both xy and yx. A semicomplete multipartite digraph with no 2-cycle is a multipartite tournament. Let $D = (V, A, \epsilon)$ be a semicomplete multipartite PCD and let α be an ordering of D. 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 set of D containing x. The following theorem is an important ingredient of our approach.

Theorem 6.1 *[26] Let D be a semicomplete multipartite PCD with n vertices. An ordering α with $\alpha(v_i) = i,\ i = 1, 2, \ldots, n$ 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 $\sigma^+(x) \geq \sigma^+(y)$.*

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 shown in the following theorem:

Theorem 6.2 *[28] Let D be a non-semicomplete multipartite PCD. Then*

$$\ell(D) = \min\{\ell(H) : \ H \in \mathcal{C}(D)\}.$$

The *completion number* $\mathrm{cn}(D)$ of a PCD D is the minimal number of arcs that we have to add to D in order to obtain a semicomplete multipartite PCD. A set S of $\mathrm{cn}(D)$ arcs is called a *completion set* of D if $D+S$ is a semicomplete multipartite PCD (the weight of each arc in S equals 1).

Theorem 6.3 *Let D be a PCD with n vertices. We can check whether $\mathrm{cn}(D) \leq k$ and, if $\mathrm{cn}(D) \leq k$, find a backward optimal ordering of D in time $O(n^3 + n^2 2^k)$.*

Proof: Let G be an undirected graph such that $V(G) = V(D)$ and two distinct vertices x, y are adjacent in G if and only if they are not adjacent in D. Observe that $\mathrm{cn}(D)$ equals $\mathrm{qd}(G)$, which is the minimum number of edges that have to be deleted from G to obtain a collection of disjoint cliques. Gram et al. [18] obtained an $O(n^3 + 1.77^k)$-time algorithm to verify whether $\mathrm{qd}(G) \leq k$. We can use this algorithm to verify whether $\mathrm{cn}(D) \leq k$.

Assume that $\mathrm{cn}(D) \leq k$. It is straightforward to use Theorems 6.1 and 6.2 in order to find an optimal backward ordering of D in time $O(n^2 2^k)$. $\diamond$

This result shows that BPCP with $\mathrm{cn}(D)$ as a parameter is FPT. We can obtain a similar result for FPCP with $\mathrm{cn}(D)$ using the polynomial time algorithm described in [23].

7 Application of BPCP to the Linear Arrangement Problem

For a graph $G = (V, E)$, let α be an ordering of G. The *length* of an edge $uv \in E$ relative to α is defined as

$$\lambda'_\alpha(uv) = |\alpha(u) - \alpha(v)|.$$

The *cost* $\mathrm{c}(\alpha, G)$ of a ordering α is the sum of lengths of all edges of G relative to α. Orderings of minimal cost are *optimal*; $\mathrm{ola}(G)$ denotes the cost of an optimal ordering of G. The well-known *Linear Arrangement Problem (LAP)* is the problem of finding $\mathrm{ola}(G)$. LAP can naturally be parameterized by asking whether $\mathrm{ola}(G) \leq k$. It is easy to see that this problem is FPT. It is significantly harder to prove that the problem $\mathrm{ola}(G) - |E| \leq k$ is FPT; the authors of [22] obtained an $O(|V| + |E| + 5.88^k)$-algorithm for this problem. Notice that the problem $\mathrm{ola}(G) - |E| \leq |E|^r$ is NP-complete for every fixed $r > 0$ [22]. It makes sense to consider $\mathrm{ola}(G) - |E|$ rather than $\mathrm{ola}(G)$ since $\lambda'_\alpha(uv) \geq 1$ for each edge uv.

In view of the last two results, we may observe that the natural parametrization of LAP and its modification ($\mathrm{ola}(G) - |E|$) are of interest only for small or sparse graphs. For dense graphs, we may consider, as a parameter, $\mathrm{qd}(\bar{G})$, which is the minimum number of edges that have to be deleted from the complement of G, $\bar{G}$, to obtain a collection of disjoint cliques. Consider G^*, the PCD obtained from G by replacing every edge xy with two arcs xy and yx, each of weight 0.5. Observe that we have $\lambda'_\alpha(xy) = 2\lambda_\alpha(yx)$ for each edge xy of G. Now the next theorem follows from Theorem 6.3.

Theorem 7.1 *Let G be a graph with n vertices. We can check whether* $\mathrm{qd}(\bar{G}) \leq k$ *and, if* $\mathrm{qd}(\bar{G}) \leq k$*, we can determine* $ola(G)$ *in time* $O(n^3 + n^2 2^k)$.

Note that $\mathrm{qd}(\bar{G})$ is a distance from tractability.

8 Induced subdigraph problem

In this section we provide a short overview of interesting results obtained recently by Raman and Sikdar [40].

A digraph $D = (V, A)$ is *complete* if $xy, yx \in A$ for each pair x, y of vertices in D. A digraph $D = (V, A)$ is *empty* if $A = \emptyset$. A set $\mathcal{D}$ of digraphs is called *hereditary* if every induced subdigraph of a digraph in $\mathcal{D}$ belongs to $\mathcal{D}$ as well. A set $\mathcal{D}$ is called non-trivial if there are infinitely many non-isomorphic digraphs in $\mathcal{D}$ and there are infinitely many non-isomorphic digraphs not in $\mathcal{D}$. Consider the following parameterized problem for a non-trivial hereditary set $\mathcal{D}$ of digraphs.

Induced Subdigraph from $\mathcal{D}$ (ISD($\mathcal{D}$))
Instance: A digraph $D = (V, A)$.
Parameter: A nonnegative integer $k \leq |V|$.
Question: Does D contain an induced subdigraph belonging to $\mathcal{D}$ with at least k vertices?

It follows from the famous Ramsey's theorem in combinatorics that for any three positive integers p, q, s there is a positive integer $R(p, q, r)$ such that every digraph with at least $R(p, q, r)$ vertices contains, as an induced subdigraph, either an empty digraph with p vertices, or a complete digraph with q vertices, or an acyclic tournament with s vertices. Thus, if a non-trivial hereditary set $\mathcal{D}$ of digraphs contains all empty digraphs, all complete digraphs and all acyclic tournaments, then ISD($\mathcal{D}$) is FPT. (The same result holds when, for some t, $\mathcal{D}$ does not contain an empty digraph with at least t vertices, complete digraph with at least t vertices and acyclic digraph with with at least t vertices.)

As a corollary of this result, the authors of [40] conclude that ISD($\mathcal{D}$) is FPT for the following sets $\mathcal{D}$: transitive digraphs, quasi-transitive digraphs, and kernel-perfect digraphs. The digraph $D = (V, A)$ is *transitive* (*quasi-transitive*) if $xy, yz \in A$ implies $xz \in A$ (either $xz \in A$, or $zx \in A$, or both). A *kernel* in D is an empty subdigraph I such that every vertex of D not in I has an out-neighbor in I. A digraph D is *kernel-perfect* if D and all its induced subdigraphs have a kernel.

Above we considered three graph types: empty digraphs, complete digraphs and acyclic tournaments. If $\mathcal{D}$ contains only two graph types or just one graph type from the three graph types, it is proved in [40] that ISD($\mathcal{D}$) is W[1]-complete and, thus, it is rather unlikely that ISD($\mathcal{D}$) is FPT.

The authors of [40] considered also all *oriented* graphs, i.e., digraphs without 2-cycles.

Clearly, a complete digraph is not oriented. Ramsey's theorem implies the following result analogous to the FPT result above: if a non-trivial hereditary set $\mathcal{D}$ of oriented graphs contains all empty digraphs and all acyclic tournaments, then ISD($\mathcal{D}$) is FPT. Similarly to the above, if $\mathcal{D}$ contains all empty digraphs, but not all acyclic digraphs or if $\mathcal{D}$ contains all acyclic digraphs, but not all empty digraphs, then ISD($\mathcal{D}$) is W[1]-complete. Thus, in particular, if $\mathcal{D}'$ is the set of all acyclic digraphs and $\mathcal{D}''$ is the set of all acyclic oriented graphs, then ISD($\mathcal{D}'$) is W[1]-complete, but ISD($\mathcal{D}''$) is FPT.

The authors of [40] formulate the following problem: given a graph G and a positive integer k (as a parameter), is there an edge induced subgraph with k edges from a fixed set $\mathcal{G}$ of graphs. The parameterized complexity of the problem is open for both undirected and directed graphs.

9 Other parameterized problems on digraphs

We have already observed that there are not so many papers on parameterized problems on digraphs. We would like to mention [21], where an FPT algorithm is obtained for the kernel problem on planar digraphs, and [14], where an FPT algorithm is derived for the Directed LAP Parameterized Above Guaranteed Value (DLAPAGV). A *kernel* in a digraph $D = (V, A)$ is a set S of independent vertices such that for each $v \in V \setminus S$ there is a vertex $s \in S$ such that $vs \in A$. In [21], it is proved that the problem of checking whether there is a kernel of size at most k in a planar digraph is FPT and an $O(2^{19.1\sqrt{k}}k^9 + |V|^2)$-time algorithm is obtained. DLAPAG is the problem of checking whether an acyclic digraph $D = (V, A)$ has on ordering with no backward arcs and with total length at most $|A| + k$. Fernau [14] describes an $O(2^k(|V| + |A|))$-time algorithm for the problem.

Certainly, it would be interesting to obtain faster algorithms for the above two problems. It is worth noting two other problem for which we do not even know whether they admit FPT algorithms or not.

Let T be a tree and let x be a vertex in T, which we call its *root*. A *rooted tree* is a digraph H obtained from T by orienting the edges of T in such a way that every path from x to another vertex y in T is a (directed) path from x to y in H. All vertices of H of out-degree zero are called *leaves*. The following problem was brought to our attention by Mike Fellows in 2005.

Directed Max-Leaf
Instance: A digraph D.
Parameter: A nonnegative integer k.
Question: Does D contain a rooted tree with at least k leaves?

A bipartite digraph $H = (V, A)$ with partite sets U and W is called a (k, ℓ)-*dominator* if $|U| = k, |W| = \ell$ and $A = \{uw : \ u \in U, w \in W\}$.

The following problem was posed in [32], a paper on data-mining the internet to identify online communities. The problem seems to be an obvious and perhaps easy candidate for $W[1]$-hardness, but in fact has resisted much effort.

(k, l)-Dominator
Instance: A digraph D.
Parameters: Positive integers k, ℓ.
Question: Does D contain a (k, ℓ)-dominator?

Acknowledgements We are thankful to the referee for several important suggestions and for proving us with [40].

References

[1] N. Ailon, M. Charikar, and A. Newman, Aggregating inconsistent information: ranking and clusstering. In *Proc. 37th STOC*, pp. 684-693, 2005.

[2] N. Alon, Ranking tournaments. To appear in *SIAM J. Discrete Math.*

[3] J. Bang-Jensen, Locally semicomplete digraphs: a generalization of tournaments. J. Graph Theory 14 (1990), 371-390.

[4] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer-Verlag, London, 2000.

[5] J. Bang-Jensen and C. Thomassen, A polynomial algorithm for the 2-path problem for semicomplete digraphs. SIAM J. Discrete Math. 5 (1992), 366-376.

[6] J.C. Bermond, A. Germa, M.C. Heydemann, and D. Sotteau, Girth in digraphs. *J. Graph Theory* 4 (1980), 337-341.

[7] L. Caccetta and R. Häggkvist, On minimal digraphs with given girth. *Congressus Numerantium* 21 (1978), 181–187.

[8] P. Charbit, S. Thomasse, and A. Yeo, The minimum feedback arc set problem is NP-hard for tournaments.To appear in *Combinatorics, Probability and Computing.*

[9] V. Conitzer, Computing Slater rankings using similarities among candidates. Tech. Report RC23748, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 2005.

[10] F.K.H.A. Dehne, M.R. Fellows, M.A. Langston, F.A. Rosamond, and K. Stevens, An $O(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem. In *Proc. 11th COCOON*, Lect. Notes in Computer Science 3595 (2005), 859–869.

[11] M. Dom, J. Guo, F. Hüffner, R. Niedermeier, and A. Truß, Fixed-Parameter tractability results for feedback set problems in tournaments. To appear in *Proc. 6th Conference on Algorithms and Complexity (CIAC 2006), Rome, Italy*, May, 2006.

[12] R.G. Downey and M.R. Fellows, *Parameterized Complexity*, Springer–Verlag, New York, 1999.

[13] M. Fellows, M. Hallett and U. Stege, Analogs and duals of the MAST problem for sequences and trees. *J. Algorithms* 49 (2003), 192–216.

[14] H. Fernau. *Parameterized Algorithmics: A Graph-Theoretic Approach.* Habilitationsschrift, Universität Tübingen, Germany, April 2005

[15] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer, 2006.

[16] M. R. Garey, and D. R. Johnson. *Computers and Intractability.* W. H. Freeman and Company, New York, 1979.

[17] F. Gavril, Some NP-complete problems on graphs. In *Proc. 11th Conf. on Information Sciences and Systems*, 91–95, John Hopkins University, Baltimore, MD, 1977.

[18] J. Gramm, J. Guo, F. Hüffner and R. Niedermeier, Graph-modeled data clustering: fixed-parameter algorithms for clique generation. In *Proc. 5th Conference on Algorithms and Complexity (CIAC 2003)*, Lect. Notes in Computer Science 2653 (2003), 108–119.

[19] J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, and S. Wernicke, Improved Fixed-Parameter Algorithms for Two Feedback Set Problems. In *Proc. WADS 2005, Waterloo, Canada, August, 2005* Lect. Notes in Computer Science 3608 (2005), 158 – 168.

[20] J. Guo, F. Hüffner and R. Niedermeier, A structural view on parameterizing problems: distance from triviality. In *Proc. 1st Intern. Workshop on Parameterized and Exact Computation (IWPEC 2004)*, Lect. Notes in Computer Science 3162 (2004), 162–173.

[21] G. Gutin, T. Kloks, C.M. Lee and A. Yeo, Kernels in planar digraphs. *Journal of Computer and System Sciences* 71 (2005), 174–184.

[22] G. Gutin, A. Rafiey, S. Szeider, and A. Yeo, The linear arrangement problem parameterized above guaranteed value. To appear in *Theory of Computing Systems.*

[23] G. Gutin and A. Yeo, Ranking the vertices of a complete multipartite paired comparison digraph. *Discrete Appl. Math.* 69 (1996), 75–82.

[24] P. Erdős and L. Posa, On the maximal number of disjoint circuits of a graph. *Publ. Math. Debrecen* 9 (1962), 3–12.

[25] A. Itai and M. Rodeh, Finding a minimum circuit in a graph. *SIAM J. Computing* 7 (1978), 413–423.

[26] M. Kano, Ranking the vertices of an r-partite paired comparison digraph. *Discrete Appl. Math.* 17 (1987), 245–253.

[27] M. Kano and A. Sakamoto, Ranking the vertices of a weighted digraph using the length of forward arcs. *Networks* 13 (1983), 143–151.

[28] M. Kano and A. Sakamoto, Ranking the vertices of a paired comparison digraph. *SIAM J. Algebraic Discrete Methods* 6 (1985), 79–92.

[29] R.M. Karp, Reducibility among combinatorial problems. Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972), 85–103, Plenum, New York, 1972.

[30] Eun Jung Kim, Private communications, July 2006.

[31] H. Kirvesoja, Experimental ergonomic evaluation with user trials: EEE product development procedures. PhD thesis, Univ. Oulu, Finland, 2001.

[32] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A.S. Tomkins and E. Upfal. The Web as a graph. In *Proc. 19th ACM SIGACT-SIGMOD-AIGART Symp. Principles of Database Systems, PODS* (2000), 1–10.

[33] A. Kunzmann and H.J. Wunderlich, An analytical approach to the partial scan problem. *J. Electronic Testing: Theory and Applications* 1 (1990), 163–174.

[34] C.E. Leiserson and J.B. Saxe, Retiming synchronous circuitry. *Algoritmica* 6 (1991), 5–35.

[35] C.L. Lucchesi, A minimumax equality for directed graphs. PhD Thesis, U. of Waterloo, Canada, 1976.

[36] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, 2006.

[37] T. Nishimura, Short cycles in digraphs. *Discrete Math.* 72 (1988), 295–298.

[38] V. Raman, Parameterized complexity. In *Proc. 7th National Seminar Theoret. Comput. Sci.* Chennai, India (1997), 1–18.

[39] V. Raman and S. Saurabh, Parameterized Complexity of feedback set problems and their duals in tournaments. *Theoretical Computer Science* 351 (2006), 446-458.

[40] V. Raman and S. Sikdar, Parameterized complexity of the induced subgraph problem in directed graphs. Submitted.

[41] B. Reed, N. Robertson, P.D. Seymour and R. Thomas, Packing directed circuits. Combinatorica 16 (1996), 535–554.

[42] R.S. Rosenberger, G.L. Peterson and J.B. Loomis, Applying a Method of Paired Comparisons to Measure Economic Values for Multiple Goods Sets. *Journal of Agricultural & Applied Economics* 34 (2002), 215–229.

[43] M.J.A. Strens and A.W. Moore, Policy Search using Paired Comparisons. *Journal of Machine Learning Research* 3 (2003), 921–950.

[44] A. Truß, Parameterized algorithms for feedback set problems in tournaments. [in German] Diplomarbeit, Institut für Informatik, Friedrich-Schiller-Univerität Jena, Dec., 2005.

[45] D.H. Younger, Graphs with interlinked directed circuits. Proc. Midwest symposium on circuit theory 2 (1973), XVI 2.1 – XVI 2.7.