Parameterized Minimum Leaf Out-Branching Problems

Gregory Gutin[1], Igor Razgon[2], and Eun Jung Kim[1]

[1] Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
gutin(eunjung)@cs.rhul.ac.uk

[2] Department of Computer Science
University College Cork, Ireland
i.razgon@cs.ucc.ie

Abstract. Given a digraph D, the Minimum Leaf Out-Branching problem (MinLOB) is the problem of finding in D an out-branching with the minimum possible number of leaves, i.e., vertices of out-degree 0. We describe three parameterizations of MinLOB and prove that two of them are NP-complete for every value of the parameter, but the third one is fixed-parameter tractable (FPT). The FPT parametrization is as follows: given a digraph D of order n and a positive integral parameter k, check whether D contains an out-branching with at most $n - k$ leaves (and find such an out-branching if it exists). We find a problem kernel of order $O(k \cdot 16^k)$ and construct an algorithm of running time $O(2^{O(k \log k)} + n^2 \log n)$, which is an 'additive' FPT algorithm.

1 Introduction

We say that a subgraph T of a digraph D is an *out-tree* if T is an oriented tree with only one vertex s of in-degree zero (called *the root*). The vertices of T of out-degree zero are called *leaves*. If T is a spanning out-tree, i.e. $V(T) = V(D)$, then T is called an *out-branching* of D. Given a digraph D, the *Minimum Leaf Out-Branching* problem (*MinLOB*) is the problem of finding in D an out-branching with the minimum possible number of leaves. Denote this minimum by $\ell_{\min}(D)$. When D has no out-branching, we write $\ell_{\min}(D) = 0$. The MinLOB problem has applications in the area of database systems, cf. the patent [8]. Notice that not every digraph D has an out-branching. It is not difficult to see that D has an out-branching (i.e., $\ell_{\min}(D) > 0$) if and only if D has just one strong initial connectivity component [3]. Since the last condition can be checked in linear time [3], we may often assume that $\ell_{\min}(D) > 0$.

Since MinLOB generalizes the hamiltonian directed path problem, MinLOB is NP-hard. In this paper, we consider three parameterizations of MinLOB and show that two of them are NP-complete for every value of the parameter, but the third one is fixed-parameter tractable.

We recall some basic notions of parameterized complexity here, for a more in-depth treatment of the topic we refer the reader to [7, 11, 19].

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. Let Π and Π' be parameterized problems with parameters k and k', respectively. An *fpt-reduction R from Π to Π'* is a many-to-one transformation from Π to Π', such that (i) $(I, k) \in \Pi$ if and only if $(I', k') \in \Pi'$ with $|I'| \leq g(k)$ for a fixed computable function g and (ii) R is of complexity $O(f(k)|I|^c)$. A *reduction to problem kernel* (or *kernelization*) is an fpt-reduction R from a parameterized problem Π to itself. In kernelization, an instance (I, k) is reduced to another instance (I', k'), which is called the *problem kernel*; $|I'|$ is the *size* of the kernel.

It is easy to see that a decidable parameterized problem is FPT if and only if it admits a kernelization (cf. [11, 19]); however, the problem kernels obtained by this general result have impractically large size. Therefore, one tries to develop kernelizations that yield problem kernels of smaller size. The survey of Guo and Niedermeier [12] on kernelization lists some problem for which polynomial size kernels and exponential size kernels were obtained. Notice that a kernelization allows one to obtain so-called *additive FPT* algorithms, i.e., algorithms of running time $O(n^{O(1)} + g(k))$, where $g(k)$ is independent of n, that are often significantly faster than their 'multiplicative' counterparts.

All digraphs in this paper are finite with no loops or parallel arcs. We use terminology and notation of [3]; in particular, for a digraph D, $V(D)$ and $A(D)$ denote its vertex and arc sets.

2 Parameterizations of MinLOB

The following is a natural way to parameterize MinLOB.

MinLOB Parameterized Naturally (MinLOB-PN)
Instance: A digraph D.
Parameter: A positive integer k.
Question: Is $\ell_{\min}(D) \leq k$?

Clearly, this problem is NP-complete already for $k = 1$ as for $k = 1$ MinLOB-PN is equivalent to the hamiltonian directed path problem. Let v be an arbitrary vertex of D. Transform D into a new digraph D_k by adding k vertices $v_1, v_2, \ldots, v_k$ together with the arcs $vv_1, vv_2, \ldots, vv_k$. Observe that D has a hamiltonian directed path terminating at v if and only if $\ell_{\min}(D) \leq k$. Since the problem is NP-complete of checking whether a digraph has a hamiltonian directed path terminating at a prescribed vertex, we conclude that MinLOB-PN is NP-complete for every fixed k.

Clearly, $\ell_{\min}(D) \leq n-1$ for every digraph D of order n. Consider a different parameterizations of MinLOB.

MinLOB Parameterized Below Guaranteed Value (MinLOB-PBGV)
Instance: A digraph D of order n with $\ell_{\min}(D) > 0$.
Parameter: A positive integer k.
Question: Is $\ell_{\min}(D) \leq n - k$?
Solution: An out-branching B of D with at most $n - k$ leaves or the answer NO to the above question.

Note that we consider MinLOB-PBGV as a search problem, not just as a decision problem. In the next section we will prove that MinLOB-PBGV is fixed-parameter tractable. We will find a problem kernel of order $O(k \cdot 16^k)$ and construct an additive FPT algorithm of running time $O(2^{O(k \log k)} + n^2 \log n)$. To obtain our results we use notions and properties of vertex cover and tree decomposition of underlying graphs.

The parametrization MinLOB-PBGV is of the type *below a guaranteed value*. Parameterizations above/below a guaranteed value were first considered by Mahajan and Raman [18] for the problems Max-SAT and Max-Cut; such parameterizations have lately gained much attention, cf. [9, 13–15, 19] (it worth noting that Heggernes, Paul, Telle, and Villanger [15] recently solved the longstanding minimum interval completion problem, which is a parametrization above guaranteed value). For directed graphs there have been only a couple of results on problems parameterized above/below a guaranteed value, see [4, 10]. In particular, Bang-Jensen and Yeo [4] proved that the following problem is FPT. Let $m_s(D)$ denote the minimum number of arcs in a strongly connected spanning subgraph of D.

Minimum Spanning Strong Subdigraph
Instance: A strongly connected digraph D of order n.
Parameter: A positive integer k.
Question: Is $m_s(D) \leq 2(n-1) - k$?

Observe that $m_s(D) \leq 2(n-1)$ for every strongly connected digraph D since a strongly connected spanning subgraph of D can be constructed as follows: find an out-branching B^+ and in-branching B^- rooted at the same vertex of D; clearly, $B^+ \cup B^-$ is strongly connected.

Let us denote by $\boldsymbol{K}_{1,p-1}$ the *star digraph* of order p, i.e., the digraph with vertices $1, 2, \ldots, p$ and arcs $12, 13, \ldots, 1p$. Our success with MinLOB-PBGV may lead us to considering the following stronger (than MinLOB-PBGV) parameterizations of MinLOB.

MinLOB Parameterized Strongly Below Guaranteed Value (MinLOB-PSBGV)
Instance: A digraph D of order n with $\ell_{\min}(D) > 0$.
Parameter: An integer $k \geq 2$.
Question: Is $\ell_{\min}(D) \leq n/k$?

Unfortunately, MinLOB-PSBGV is NP-complete for every fixed $k \geq 2$. To prove this consider a digraph D of order n and a digraph H obtained from D by adding to it the star digraph $\boldsymbol{K}_{1,p-1}$ on $p = \lfloor n/(k-1) \rfloor$ vertices ($V(D) \cap V(\boldsymbol{K}_{1,p-1}) = \emptyset$) and appending an arc from vertex 2 of $\boldsymbol{K}_{1,p-1}$ to an arbitrary vertex y of D. Observe that $\ell_{\min}(H) = p - 1 + \ell_{min}(D, y)$, where $\ell_{min}(D, y)$ is the minimum possible number of leaves in an out-branching rooted at y, and that $\frac{1}{k}|V(H)| = p + \epsilon$, where $0 \leq \epsilon < 1$. Thus, $\ell_{\min}(H) \leq \frac{1}{k}|V(H)|$ if and only if $\ell_{min}(D, y) = 1$. Hence, the hamiltonian directed path problem with fixed initial vertex (vertex y in D) can be reduced to MinLOB-PSBGV for every fixed $k \geq 2$ and, therefore, MinLOB-PSBGV is NP-complete for every $k \geq 2$.

3 Solving MinLOB-PBGV

As noted in the introduction, we may safely assume that the given digraph has at least one out-branching. Let D be a digraph and let B be an out-branching of D. Let P be the set of *parents of leaves* of B, i.e., P is the set of all such vertices u that at least one child of u is a leaf in B. We partition the vertices of P into the sets BP (*bad parents*) and GP (*good parents*) where BP consists of all the vertices of P whose out-degree in B is at least 2, and GP are those vertices whose out-degree in B is 1. (We'd like to emphasize that it is essential that we consider in this definition the out-degree of the vertices in B, *not* in the whole graph D.) We call a leaf u of B a *good leaf* if its parent belongs to GP, otherwise it is a *bad leaf.*

Definition 1. *We call B a* normalized out-branching *if D has no arc uv such that both u and v are leaves and v is a bad leaf.*

The following two lemmas reveal interesting properties of normalized out-branchings.

Lemma 1. *A normalized out-branching of D can be constructed in polynomial time.*

Proof. Pick an arbitrary out-branching B of D, this can be done in a polynomial time [3]. If B is normalized (this clearly can be checked in polynomial time), just stop and return it. Otherwise, let $uv \in A(D)$ such that both u and v are leaves and v is a bad leaf. Let w be the parent of v and consider the out-branching B' obtained from B by removal of wv and addition of uv.

Observe that B' has less leaves than B. Indeed, u is not a leaf in B' while w is not transformed to a leaf due to being a bad parent. Since any out-branching has at least one leaf, after less than n such transformations, we obtain a normalized out-branching. □

For the next lemma we need the following definitions. Let D be a digraph. The *underlying graph* $UG(D)$ of D is obtained from D by omitting all orientation of arcs and by deleting one edge from each resulting pair of parallel edges. A *vertex cover* of D is a vertex cover of $UG(D)$.

Lemma 2. *Let D be a digraph of order $n \geq 3$ and let B be a normalized out-branching of D. Then at least one of the following statements is true.*

1. *B has at most $n - k$ leaves*
2. *D has a vertex cover of size at most $2k - 3$.*

Proof. Assume that B has at least $n - k + 1$ leaves. Then the number of non-leaf vertices is at most $k - 1$. Note that the number of good leaves does not exceed the number of non-leaf vertices because each good leaf has a parent but no two good leaves share the same parent. Furthermore, the non-leaf vertices together with good leaves constitute a vertex cover of D because, by definition of a normalized branching there is no arc between two bad leaves. Therefore we may conclude that D has a vertex cover of size at most $2(k - 1)$. A slightly more precise evaluation allows us to reduce this number to $2k - 3$.

Notice that the root r of B does not belong to GP. Indeed, if $r \in GP$ then the only child of r is the respective good leaf, which, in turn, has no children. Thus, D has only two vertices which contradicts our assumption. Hence, the number of good leaves does not exceed the number of non-root non-leaf vertices of B, which is at most $k - 2$. Consequently, the size of the vertex cover is at most $2k - 3$. □

In the rest of this section we assume that the digraph D under consideration has at least 3 vertices.

It follows from the combination of Lemma 1 and Lemma 2 that there is a polynomial algorithm that given an instance (D, k) of the MinLOB-PBGV problem either returns a solution or specifies a vertex cover of D of size at most $2k - 3$. Using the following lemma, we will be able to show that in the latter case there is an elegant way of kernelization.

Lemma 3. *Let (D, k) be an instance of the MinLOB-PBGV problem and let U be a vertex cover of D. Then (D, k) can be reduced in polynomial time to an instance (D', k) of this problem where $|V(D')| = O(|U|4^{|U|})$.*

Proof. The strategy of this proof can be called 'folding-unfolding': We first partition the set of vertices of $V(D) \setminus U$ into classes (the folding stage) and show that the number of distinct classes cannot be too large. Then we show that the number of vertices within each class cannot be too large (the unfolding stage).

Let us partition the vertices of $V(D) \setminus U$ into *equivalence classes* so that the vertices within each equivalence class have the same sets of in- and out-neighbors. Let us evaluate the largest possible number of such equivalence classes. Let $u \in U$ and $v \in V(D) \setminus U$. Note that there may be 4 possibilities for the set $A(u,v)$ of arcs between u and v: $A(u,v) = \emptyset$, $|A(u,v)| = 2$, $A(u,v) = \{uv\}$, or $A(u,v) = \{vu\}$. Consequently, there may be at most $4^{|U|}$ possibilities of setting all adjacencies between U and v. Therefore the number of distinct equivalence classes does not exceed $4^{|U|}$.

In order to proceed it is more convenient to think about MinLOB-PBGV problem as a problem of constructing an out-branching with at least k non-leaf vertices rather than at most $n-k$ leaf vertices (clearly these two representations are equivalent).

Let D' be a digraph obtained from D as follows. For each equivalence class which contains $t > 2|U|$ vertices remove $t - 2|U|$ arbitrary vertices from this class. We claim that (D,k) is a 'YES' instance of the MinLOB-PBGV problem if and only if (D',k) is.

Assume that (D',k) is a 'YES' instance and let B' be an out-branching of D' having at least k non-leaf vertices. If no vertex has been removed then $D = D'$, i.e., B' is the desired out-branching of D and (D,k) is a 'YES' instance. Otherwise note that *the vertices within an equivalence class of size two or more have at least one in-neighbor in D*. Otherwise D has two or more vertices with in-degree 0 and hence no out-branching in contradiction to our assumption. Consequently the in-degree of each removed vertex is at least 1.

Therefore for each removed vertex v we can pick up an arbitrary in-neighbor u in D and add the arc uv to B'. Let B be the resulting digraph. It is not hard to observe that B is an out-branching of D with at least k non-leaf vertices. Therefore (D,k) is a 'YES' instance, as required.

Assume that (D,k) is a 'YES' instance and let B be an out-branching of D with at least k non-leaf vertices. Let S be a set of vertices of $V(D) \setminus U$ having the following properties.

- Each non-leaf vertex of $V(D) \setminus U$ belongs to S
- Let u be a non-leaf vertex of U such that all the children of u in B are leaves and belong to $V(D) \setminus U$. Then exactly one child of u belongs to S
- No other vertices are contained in S.

Let B^* be a digraph obtained by removal from B all the vertices of $(V(D) \setminus U) \setminus S$. The first property of S ensures that all the non-leaf vertices of B remain in B^*. Therefore B^* is an out-tree. The second property ensures that all the non-leaf vertices of B remain non-leaf vertices in B^*. Since no non-leaf vertices are introduced by transformation from B to B^* we may conclude that B and B^* have the same set of non-leaf vertices.

In order to proceed, we calculate the size of S. Each non-leaf vertex of S has a child in U and no two vertices share the same child. Hence the number of non-leaf vertices of S is at most $|U|$. Each leaf vertex of S has a parent in U and, by the second property, no two leaf vertices of S share a parent. That is the number of leaf vertices of S is at most $|U|$ and $|S| \leq 2|U|$.

Let $Z_1, \ldots, Z_l$ be the equivalence classes of $V(D) \setminus U$ mentioned above. Denote $S \cap Z_i$ by S_i. Then we have proven the following statement.

Statement 1 *It is possible to pick up $|S_i| \leq 2|U|$ vertices from each Z_i to form a set S which together with U constitute the vertex set for a subgraph B^* of D which is an out-tree with at least k non-leaf vertices and the set of non-leaf vertices of B^* is the set of non-leaf vertices of an out-branching of D.*

Taking into account that for each i all the vertices within Z_i have the same neighborhood, we can formulate a stronger statement.

Statement 2 *Pick up $|S_i| \leq 2|U|$ arbitrary vertices from each Z_i. The selected vertices form a set S which together with U constitute the vertex set for a subgraph B^* of D which is an out-tree with at least k non-leaf vertices and the set of non-leaf vertices of B^* is the set of non-leaf vertices of an out-branching of D.*

Note that for each i, $|S_i| \leq |V(D') \cap Z_i|$. This is clear if $V(D') \cap Z_i = V(D) \cap Z_i$. Otherwise, it follows from the fact that $|V(D') \cap Z_i| = 2|U|$, while $|S_i| \leq 2|U|$. Taking into account the construction of D' and Statement 2, we get the following:

Statement 3 *There is a subgraph B^* of D' which is an out-tree with at least k non-leaf vertices and the set of non-leaf vertices of B^* is the set of non-leaf vertices of an out-branching of D.*

If the in-degree of each vertex u of $V(D') \setminus V(B^*)$ in D' is at least 1, we can extend B^* to an out-branching of D' with the same non-leaf vertices as shown above (i.e. selecting an in-neighbor for each vertex of $V(D') \setminus V(B^*)$ and adding appropriate arcs to B^*). Therefore, in this case (D', k) is a 'YES' instance. Otherwise, there is a vertex $u \in V(D') \setminus V(B^*)$ whose in-degree is D' is zero. Note that the transformation from D to D' does not change the degrees (both in- and out-) of the vertices of $V(D) \setminus U$ and does not reduce the degrees of vertices of U from non-zero to zero. It follows that the in-degree of u in D is also zero. Consequently u is the root of *any* out-branching of D. In particular, it is a non-leaf vertex of *any* out-branching of D (because otherwise $|V(D)| = 1$ in contradiction to our assumption). It follows that u is a non-leaf vertex of B^* and cannot belong to $V(D') \setminus V(B^*)$. This contradiction shows that (D', k) is a 'YES' instance of the considered problem.

Thus we have shown that the the MinLOB-PBGV problem with parameter k on D can be reduced to the MinLOB-PBGV problem with parameter k on digraph D', $|V(D')| \leq 2|U|4^{|U|} + |U|$, which completes the proof of the present lemma. $\square$

Combining Lemmas 1, 2 and 3 we immediately obtain the following theorem.

Theorem 4. *The MinLOB-PBGV problem is FPT. In particular, there is a polynomial-time algorithm which given an instance (D, k) of the MinLOB-PBGV problem, either produces a solution or reduces the instance (D, k) to an instance (D', k) where $|V(D')| = O(k \cdot 16^k)$.*

Thus we have shown that the MinLOB-PBGV problem has a kernel of order proportional to $k \cdot 16^k$. Now we have to clarify how we explore this kernel in order to get the desired out-branching. A straightforward exploration of all possible out-branchings (using, e.g., the main algorithm of [16]) is not a good choice because the number of different out-branchings may be up to p^{p-1}, where $p = |V(D')| = (k \cdot 16^k)$. Indeed, by the famous Kelly's formula the number of spanning trees in the complete graph K_p on p vertices equals p^{p-2}. In the complete digraph on p vertices, one can get p out-branchings from each spanning tree of K_p by assigning a vertex to be the root.

In order to achieve a better running time we provide an alternative way of showing the fixed-parameter tractability of the MinLOB-PBGV problem based on the notion of *tree decomposition.*

A *tree decomposition* of an (undirected) graph G is a pair (X, U) where U is a tree whose vertices we will call *nodes* and $X = \{X_i : i \in V(U)\}$ is a collection of subsets of $V(G)$ (called *bags*) such that

1. $\bigcup_{i \in V(U)} X_i = V(G)$,
2. for each edge $\{v, w\} \in E(G)$, there is an $i \in V(U)$ such that $v, w \in X_i$, and
3. for each $v \in V(G)$ the set of nodes $\{i : v \in X_i\}$ form a subtree of U.

The *width* of a tree decomposition $(\{X_i : i \in V(U)\}, U)$ equals $\max_{i \in V(U)}\{|X_i| - 1\}$. The *treewidth* of a graph G is the minimum width over all tree decompositions of G. We use the notation $\mathrm{tw}(G)$ to denote the treewidth of a graph G.

By a *tree decomposition of a digraph D* we will mean a tree decomposition of the underlying graph $UG(D)$. Also, $\mathrm{tw}(D) = \mathrm{tw}(UG(D))$.

Theorem 5. *There is a polynomial algorithm that, given an instance* (D, k) *of the MinLOB-PBGV problem, either finds a solution or establishes a tree decomposition of* D *of width at most* $2k - 3$.

Proof. Combining Lemma 1 and Lemma 2, there is a polynomial algorithm which either finds a solution or specifies a vertex cover C of D of size at most $2k-3$. Let $I = \{v_1, \dots, v_s\} = V(D) \setminus C$. Consider a star U with nodes $x_0, x_1, \dots, x_s$ and edges $x_0x_1, x_0x_2, \dots, x_0x_s$. Let $X_0 = C$ and $X_i = X_0 \cup \{v_i\}$ for $i = 1, 2, \dots, s$ and let X_j be the bag corresponding to x_j for every $j = 0, 1, \dots, s$. Observe that $(\{X_0, X_1, \dots, X_s\}, U)$ is a tree decomposition of D and its width is at most $2k - 3$. □

Theorem 5 shows that an instance (D, k) of the MinLOB-PBGV problem can be reduced to another instance having treewidth $O(k)$. Using standard dynamic programming techniques we can solve this instance in time $2^{O(k \log k)} n^{O(1)}$. On the first glance it seems that this running time makes the above kernelization redundant. However, although the $O(k \cdot 16^k)$ kernel is not polynomial, yet it is much smaller than $2^{O(k \log k)}$. Therefore if we first find a kernel and then establish the tree decomposition, the resulting dynamic programming algorithm runs in time $2^{O(k \log k)} + n^{O(1)}$ *without* changing the constant at $k \log k$. More precisely, Theorem 4 and Theorem 5 imply the following corollary.

Corollary 1. *Let* D *a digraph. Assume that for this digraph a tree-decomposition of width* $2k-3$ *is specified. Assume also that given this tree-decomposition the MinLOB-PBGV problem can be solved in time* $2^{ck \log k} n^{O(1)}$. *Then for any instance* (D, k), *the MinLOB-PBGV problem can be solved in time* $O(2^{ck \log k + dk} + n^2 + nk \log n)$, *where* d *is a constant.*

Proof. The additional dk at the exponent follows from replacing $n^{O(1)}$ by $(k16^k)^{O(1)}$. It remains to clarify where $n^2 + nk \log n$ comes from. Observe that the polynomial part of the algorithm produces a normalized out-branching and partitions into equivalence classes the vertices which lie outside of the specified vertex cover. The normalization can be carried out in two stages. In the first stage all the arcs violating the normalized out-branching property are detected. This can be done in a time proportional to the number of arcs, i.e. $O(n^2)$. On the second stage all these arcs are eliminated. After that the branching is normalized because the elimination routine does not produce new eliminating arcs. [1] Thus the normalization can be done on $O(n^2)$. The partition into equivalence classes can be done by sorting the rows of the adjacency matrix according to any selected lexicographic order and traversing the rows

[1] Moreover, elimination of an arc can cause another arc to be non-violating.

of the resulting matrix. The sorting can be done in time $O(n \log n)$ multiplied by the comparison cost, i.e. $O(k)$. $\square$

The above results imply the following:

Theorem 6. *The MinLOB-PBGV problem can be solved by an additive FPT algorithm of running time* $O(2^{O(k \log k)} + n^2 \log n)$.

4 Further Research

We have proved that MinLOB-PBGV is FPT. It would be interesting check whether MinLOB-PBGV admits significantly more efficient FPT algorithms, i.e., algorithms of complexity $O(c^k n^{O(1)})$, where c is a constant. The same question is of interest for the following related problem, which is the natural parametrization of the Maximum Leaf Out-Branching problem.

MaxLOB Parameterized Naturally (MaxLOB-PN)
Instance: A digraph D.
Parameter: A positive integer k.
Question: Does D have an out-branching with at least k leaves ?

Alon et al. [1, 2] proved this problem is FPT for several special classes of digraphs such as strongly connected digraphs and acyclic digraphs and Bonsma and Dorn [5] proved that the problem is FPT. Note that in the three papers, MaxLOB-PN algorithms are of running time $O(2^{k(\log k)^{O(1)}} \cdot n^{O(1)})$.

Acknowledgements We are grateful to Bruno Courcelle for useful discussions related to his fundamental theorem. Research of Gutin and Kim was supported in part by an EPSRC grant. Part of the paper was written when Razgon was vising Department of Computer Science, Royal Holloway, University of London.

References

1. N. Alon, F. Fomin, G. Gutin, M. Krivelevich and S. Saurabh, Parameterized Algorithms for Directed Maximum Leaf Problems. *Proc. ICALP 2007*, LNCS **4596** (2007), 352-362.
2. N. Alon, F. Fomin, G. Gutin, M. Krivelevich and S. Saurabh, Better Algorithms and Bounds for Directed Maximum Leaf Problems. *Proc. Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2007, India.* To appear in LNCS.
3. J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications.* Springer, 2000; freely available online at `www.cs.rhul.ac.uk/books/dbook/`
4. J. Bang-Jensen and A. Yeo, The minimum spanning strong subdigraph problem is fixed parameter tractable. To appear in *Discrete Applied Math.*
5. P.S. Bonsma and F. Dorn, An FPT Algorithm for Directed Spanning k-Leaf. Preprint 046-2007, Combinatorial Optimization & Graph Algorithms Group, TU Berlin, Nov. 2007.

6. B. Courcelle, The expression of graph properties and graph transformations in monadic second-order logic. In *Handbook of graph grammars and computing by graph transformations, Vol. 1 : Foundations* (G. Rozenberg ed.), World Scientific, 1997, 313–400.
7. R.G. Downey and M.R. Fellows, *Parameterized Complexity*, Springer,1999.
8. A. Demers and A. Downing, Minimum leaf spanning tree. US Patent no. 6,105,018, August 2000.
9. H. Fernau, *Parameterized Algorithmics: A Graph-theoretic Approach.* Habilitation thesis, U. Tübingen, 2005.
10. H. Fernau, Parameterized Algorithmics for Linear Arrangement Problems. Manscript, July 2005.
11. J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer, 2006.
12. J. Guo and R. Niedermeier, Invitation to Data Reduction and Problem Kernelization. *ACM SIGACT News* 38 (2007), 31–45.
13. G. Gutin, A. Rafiey, S. Szeider and A. Yeo, The Linear Arrangement Problem Parameterized Above Guaranteed Value. *Theory of Computing Systems* 41 (2007), 521–538.
14. G. Gutin, S. Szeider and A. Yeo, Fixed-Parameter Complexity of Minimum Profile Problems. To appear in *Algorithmica.*
15. P. Heggernes, C. Paul, J.A. Telle, and Y. Villanger, Interval completion with few edges. Proc. STOC 2007 - 39th ACM Symposium on Theory of Computing, 374 – 381.
16. S. Kapoor and H. Ramesh, An Algorithm for Enumerating All Spanning Trees of a Directed Graph. *Algorithmica* **27** (2000), 120–130.
17. T. Kloks, *Treewidth-Computations and approximations*, Springer-Verlag LNCS 842, 1994.
18. M. Mahajan and V. Raman, Parameterizing above guaranteed values: MaxSat and Max-Cut. *J. Algorithms* 31 (1999), 335–354.
19. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, 2006.