REPORTS
IN
INFORMATICS

Parameterized Algorithms for Directed Maximum Leaf Problems

Noga Alon, Fedor V. Fomin, Gregory Gutin,
Michael Krivelevich, and Saket Saurabh

Department of Informatics

UNIVERSITY OF BERGEN
Bergen, Norway

Parameterized Algorithms for Directed Maximum Leaf Problems
(Extended Abstract)

Noga Alon[*]   Fedor V. Fomin[†]   Gregory Gutin[‡]
Michael Krivelevich[§]   Saket Saurabh[¶]

Abstract

We prove that finding a rooted subtree with at least k leaves in a digraph is a fixed parameter tractable problem. A similar result holds for finding rooted spanning trees with many leaves in digraphs from a wide family $\mathcal{L}$ that includes all strong and acyclic digraphs. This settles completely an open question of Fellows and solves another one for digraphs in $\mathcal{L}$. Our algorithms are based on the following combinatorial result which can be viewed as a generalization of many results for a 'spanning tree with many leaves' in the undirected case, and which is interesting on its own: If a digraph $D \in \mathcal{L}$ of order n with minimum in-degree at least 3 contains a rooted spanning tree, then D contains one with at least $(n/2)^{1/5} - 1$ leaves.

1 Introduction

The MAXIMUM LEAF SPANNING TREE problem (finding a spanning tree with the maximum number of leaves in a connected undirected graph) is an intensively studied problem from an algorithmic as well as a combinatorial point of view [5, 8, 11, 15, 20, 28]. It fits into the broader class of spanning tree problems on which hundreds of papers have been written; see e.g. the book [32]. It is known to be NP-hard [16], and APX-hard [14], but can be fairly well approximated efficiently with multiplicative factor 3 [24] and even 2 [28].

---

[*]Department of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: nogaa@post.tau.ac.il. Research supported in part by a USA-Israeli BSF grant and by a grant from the Israel Science Foundation.

[†]Department of Informatics, University of Bergen, POB 7803, 5020 Bergen, Norway. Email: fedor.fomin@ii.uib.no. Supported by the Norwegian Research Council.

[‡]Department of Computer Science, Royal Holloway, University of London, Egham, Surrey, TW20 0EX, UK. Email: gutin@cs.rhul.ac.uk. Research supported in part by an EPSRC grant.

[§]Department of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: krivelev@post.tau.ac.il. Research supported in part by a USA-Israel BSF grant and by a grant from the Israel Science Foundation.

[¶]Department of Informatics, University of Bergen, POB 7803, 5020 Bergen, Norway and the Institute of Mathematical Sciences, Chennai, 600 017, India. Email: saket@imsc.res.in

In this paper, we initiate the combinatorial and algorithmic study of two natural generalizations of the problem to digraphs. We say that a subdigraph 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 DIRECTED MAXIMUM LEAF OUT-BRANCHING problem is the problem of finding in D an out-branching with the maximum possible number of leaves. Denote this maximum by $\ell_s(D)$. When D has no out-branching, we write $\ell_s(D) = 0$. Similarly, the DIRECTED MAXIMUM LEAF OUT-TREE problem is the problem of finding in D an out-tree with the maximum possible number of leaves, which we denote by $\ell(D)$. Both these problems are equivalent for connected undirected graphs, as any maximum leaf tree can be extended to a maximum leaf spanning tree with the same number of leaves.

Notice that $\ell(D) \geq \ell_s(D)$ for each digraph D. Let $\mathcal{L}$ be the family of digraphs D for which either $\ell_s(D) = 0$ or $\ell_s(D) = \ell(D)$. It is easy to see that $\mathcal{L}$ contains all strong and acyclic digraphs.

We investigate the above two problems from the *parameterized complexity* point of view. Parameterized Complexity is a recent approach to deal with intractable computational problems having some parameters that can be relatively small with respect to the input size. This area has been developed extensively during the last decade. For decision problems with input size n, and a parameter k, the goal is to design an algorithm with runtime $f(k)n^{O(1)}$ where f is a function of k alone. Problems having such an algorithm are said to be fixed parameter tractable (FPT). The book by Downey and Fellows [9] provides a good introduction to the topic of parameterized complexity. For recent developments see the books by Flum and Grohe [13] and by Niedermeier [26].

The parameterized version of the DIRECTED MAXIMUM LEAF OUT-BRANCHING (the DIRECTED MAXIMUM LEAF OUT-TREE) problem is defined as follows: Given a digraph D and a positive integral parameter k, is $\ell(D) \geq k$ ($\ell_s(D) \geq k$)? We denote the parameterized versions of the DIRECTED MAXIMUM LEAF OUT-BRANCHING and the DIRECTED MAXIMUM LEAF OUT-TREE problems by k-DMLOB and k-DMLOT respectively.

While the parameterized complexity of almost all natural problems on undirected graphs is well understood, the world of digraphs is still wide open. The main reason for this anomaly is that most of the techniques developed for undirected graphs cannot be used or extended to digraphs. One of the most prominent examples is the FEEDBACK VERTEX SET problem, which is easily proved to be FPT for undirected graphs, while its parameterized complexity on digraphs is a long standing open problem in the area. In what follows we briefly explain why the standard techniques for the MAXIMUM LEAF SPANNING TREE problem on undirected graphs cannot be used for its generalizations to digraphs.

- The Graph Minors Theory of Robertson and Seymour [29] is a powerful (yet non-constructive) technique for establishing membership in FPT. For example, this machinery can be used to show that the MAXIMUM LEAF SPANNING

Tree problem is FPT for undirected graphs (see [10]). However, Graph Minors Theory for digraphs is still in a preliminary stage and at the moment cannot be used as a tool for tackling interesting directed graph problems.

- Bodlaender [3] used the following arguments to prove that the Maximum Leaf Spanning Tree problem is FPT: If an undirected graph G contains a star $K_{1,k}$ as a minor, then it is possible to construct a spanning tree with at least k leaves from this minor. Otherwise, there is no $K_{1,k}$ minor in G, and it is possible to prove that the treewidth of G is at most $f(k)$. Thus, dynamic programming can be used to decide whether there is a tree with k leaves. This approach does not work on directed graphs because containing a big out-tree as a minor does not imply the existence of an out-branching or out-tree with many leaves in the original graph. In short, the properties of having no out-branching with at least k leaves or having no out-tree with k leaves are not minor closed.

- The seemingly most efficient approach for designing FPT algorithms for undirected graphs is based on a combination of combinatorial bounds and pre-processing rules for handling vertices of small degrees. Kleitman and West [20] and Linial and Sturtevant [23] showed that every connected undirected graph G on n vertices with minimum degree at least 3 has a spanning tree with at least $n/4 + 2$ leaves. Bonsma et al. [5] combined this combinatorial result with clever preprocessing rules to obtain the fastest known algorithm for the k-Maximum Leaf Spanning Tree problem, running in time $O(n^3 + 9.4815^k k^3)$. It is not clear how to devise a similar approach for digraphs.

Our Contribution. We obtain a number of combinatorial and algorithmic results for the Directed Maximum Leaf Out-Branching and the Directed Maximum Leaf Out-tree problems. Our main combinatorial result (Theorem 3.3) is the proof that for every digraph $D \in \mathcal{L}$ of order n with minimum in-degree at least 3, $\ell_s(D) \geq (n/2)^{1/5} - 1$ provided $\ell_s(D) > 0$. This can be viewed as a generalization of many combinatorial results for undirected graphs related to the existence of spanning trees with many leaves [17, 20, 23].

Our main algorithmic contributions are fixed parameter tractable algorithms for the k-DMLOB and the k-DMLOT problems for digraphs in $\mathcal{L}$ and for all digraphs, respectively. The algorithms are based on a decomposition theorem which uses ideas from the proof of the main combinatorial result. More precisely, we show that either a digraph contains a structure that can be extended to an out-branching with many leaves, or the pathwidth of the underlying undirected graph is small. This settles completely an open question of Mike Fellows [12, 19] and solves another one for digraphs in $\mathcal{L}$.

3

2 Preliminaries

Let D be a digraph. By $V(D)$ and $A(D)$ we represent the vertex set and arc set of D, respectively. An *oriented graph* is a digraph with no directed 2-cycle. Given a subset $V' \subseteq V(D)$ of a digraph D, let $D[V']$ denote the digraph induced on V'. The *underlying undirected graph* $UN(D)$ of D is obtained from D by omitting all orientation of arcs and by deleting one edge from each resulting pair of parallel edges. The *connectivity components* of D are the subdigraphs of D induced by the vertices of components of $UN(D)$. A vertex y of D is an *in-neighbor* (*out-neighbor*) of a vertex x if $yx \in A$ ($xy \in A$, respectively). The *in-degree* $d^-(x)$ (*out-degree* $d^+(x)$) of a vertex x is the number of its in-neighbors (out-neighbors). A vertex s of a digraph D is a *source* if the in-degree of s is 0. A strong component S of a digraph D is a *source strong component* if no vertex of S has an in-neighbor in $V(D) \setminus V(S)$. The following simple result gives necessary and sufficient conditions for a digraph to have an out-branching.

Proposition 2.1 ([2]). *A digraph D has an out-branching if and only if D has a unique source strong component.*

This assertion allows us to check whether $\ell_s(D) > 0$ in time $O(|V(D)|+|A(D)|)$. Thus, we will often assume, in the rest of the paper, that the digraph D under consideration has an out-branching.

Let $P = u_1 u_2 \ldots u_q$ be a directed path in a digraph D. An arc $u_i u_j$ of D is a *forward* (*backward*) *arc for* P if $i \leq j - 2$ ($j < i$, respectively). Every backward arc of the type $v_{i+1} v_i$ is called *double*.

For a natural number n, $[n]$ denotes the set $\{1, 2, \ldots, n\}$.

The notions of treewidth and pathwidth were introduced by Robertson and Seymour in [30] and [31] (see [3] and [25] for surveys).

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 $X = (\{X_i \mid i \in V(U)\})$ is a collection of subsets of $V(G)$ 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 $\{i \mid v \in X_i\}$ forms a subtree of U.

The *width* of a tree decomposition $(\{X_i \mid i \in V(U)\}, U)$ 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.

If in the definitions of a tree decomposition and treewidth we restrict U to be a tree with all vertices of degree at most 2 (i.e., a path) then we have the definitions of path decomposition and pathwidth. We use the notation $tw(G)$ and $pw(G)$ to denote the treewidth and the pathwidth of a graph G.

We also need an equivalent definition of pathwidth in terms of vertex separators with respect to a linear ordering of the vertices. Let G be a graph and let $\sigma =$

$(v_1, v_2, \ldots, v_n)$ be an ordering of $V(G)$. For $j \in [n]$ put $V_j = \{v_i : i \in [j]\}$ and denote by ∂V_j all vertices of V_j that have neighbors in $V \setminus V_j$. Setting

$$vs(G, \sigma) = \max_{i \in [n]} |\partial V_i|,$$

we define the *vertex separation* of G as

$$vs(G) = \min\{vs(G, \sigma) \colon \sigma \text{ is an ordering of } V(G)\}.$$

The following assertion is well-known. It follows directly from the results of Kirousis and Papadimitriou [22] on interval width of a graph, see also [21].

Proposition 2.2 ([21, 22])**.** *For any graph G, $vs(G) = pw(G)$.*

3   Combinatorial Lower Bounds on $\ell(D)$ and $\ell_s(D)$

Let $\mathcal{D}$ be a family of digraphs. Notice that if we can show that $\ell_s(D) \geq g(n)$ for every digraph $D \in \mathcal{D}$ of order n, where $g(n)$ is tending to infinity as n tends to infinity, then k-DMLOB is FPT on $\mathcal{D}$. Indeed, $g(n) < k$ holds only for digraphs with less than some $G(k)$ vertices and we can generate all out-branchings in such a digraph in time bounded by a function of k.

Unfortunately, bounds of the type $\ell_s(D) \geq g(n)$ are not valid for all strongly connected digraphs. Nevertheless, such bounds hold for wide classes of digraphs as we show in the rest of this section.

The following assertion shows that $\mathcal{L}$ includes a large number digraphs including all strong and acyclic digraphs (and, also, well-studied classes of semicomplete multipartite digraphs and quasi-transitive digraphs, see [2] for the definitions).

Proposition 3.1. *Suppose that a digraph D satisfies the following property: for every pair R and Q of distinct strong components of D, if there is an arc from R to Q then each vertex of Q has an in-neighbor in R. Then $D \in \mathcal{L}$.*

Proof: Let T be a maximal out-tree of D with $\ell(D)$ leaves. We may assume that $\ell_s(D) > 0$ and $V(T) \neq V(D)$. Let H be the unique source strong component of D and let r be the root of T. Observe that $r \in V(H)$ as otherwise we could extend T by adding to it an arc ur, where u is some vertex outside the strong component containing r. Let C be a strong component containing a vertex from T. Observe that $V(C) \cap V(T) = V(C)$ as otherwise we could extend T by appending to it some arc uv, where $u \in V(C) \cap V(T)$ and $v \in V(C) \setminus V(T)$. Similarly, one can see that T must contain vertices from all strong components of D. Thus, $V(T) = V(D)$, a contradiction. $\qquad\qquad\square$

3.1   Digraphs with Restricted In-Degree

Lemma 3.2. *Let D be an oriented graph of order n with every vertex of in-degree 2 and let D have an out-branching. If D has no out-tree with k leaves, then $n \leq 2k^5$.*

Proof: Assume that D has no out-tree with k leaves. Consider an out-branching T of D with p leaves (clearly $p < k$). As long as the remaining part of T is not a path consider a directed path from the root of T to a leaf, and omit the part of it that starts right after the last vertex along the path whose degree in T is at least 3, and ends at the leaf. This process provides a collection $\mathcal{P}$ of p vertex-disjoint directed paths covering all vertices of D.

Let $P \in \mathcal{P}$ have $q \geq n/p$ vertices and let $P' \in \mathcal{P} \setminus \{P\}$. There are at most $k-1$ vertices on P with in-neighbors on P' since otherwise we could choose a set X of at least k vertices on P for which there were in-neighbors on P'. The vertices of X would be leaves of an out-tree formed by the vertices $V(P') \cup X$. Thus, there are $m \leq (k-1)(p-1) \leq (k-1)(k-2)$ vertices of P with in-neighbors outside P and at least $q - (k-2)(k-1)$ vertices of P have both in-neighbors on P.

Let $P = u_1 u_2 \dots u_q$. Suppose that there are $2(k-1)$ indices

$$i_1 < j_1 \leq i_2 < j_2 \leq \dots \leq i_{k-1} < j_{k-1}$$

such that each $u_{i_s} u_{j_s}$ is a forward arc for P. Then the arcs

$$\{u_{i_s} u_{j_s}, u_{j_s} u_{j_s+1}, \dots, u_{i_{s+1}-1} u_{i_{s+1}} : 1 \leq s \leq k-2\} \cup \{u_{i_{k-1}} u_{j_{k-1}}\} \cup \{u_{i_s} u_{i_s+1} : 1 \leq s \leq k-1\}$$

form an out-tree with k leaves, a contradiction.

Let f be the number of forward arcs for P. Consider the graph G whose vertices are all the forward arcs and a pair $u_i u_j, u_s u_r$ of forward arcs are adjacent in G if the intervals $[i, j-1]$ and $[s, r-1]$ of the real line intersect. Observe that G is an interval graph and, thus, a perfect graph. By the result of the previous paragraph, the independence number of G is less than $k-1$. Thus, the chromatic number of G and the order of its largest clique Q is at least $f/(k-2)$. Let $V(Q) = \{u_{i_s} u_{j_s} : 1 \leq s \leq g\}$ and let $h = \min\{j_s - 1 : 1 \leq s \leq g\}$. Observe that each interval $[i_s, j_s - 1]$ contains h. Therefore, we can form an out-tree with vertices

$$\{u_1, u_2, \dots, u_h\} \cup \{u_{j_s} : 1 \leq s \leq g\}$$

in which $\{u_{j_s} : 1 \leq s \leq g\}$ are leaves. Hence we have $\frac{f}{k-2} \leq k-1$ and, thus, $f \leq (k-2)(k-1)$.

Let uv be an arc of $A(D) \setminus A(P)$ such that $v \in V(P)$. There are three possibilities: (i) $u \notin V(P)$, (ii) $u \in V(P)$ and uv is forward for P, (iii) $u \in V(P)$ and uv is backward for P. By the inequalities above for m and f, we conclude that there are at most $2(k-2)(k-1)$ vertices on P which are not terminal vertices of backward arcs. Consider a path $R = v_0 v_1 \dots v_r$ formed by backward arcs. Observe that the arcs $\{v_i v_{i+1} : 0 \leq i \leq r-1\} \cup \{v_j v_j^+ : 1 \leq j \leq r\}$ form an out-tree with r leaves, where v_j^+ is the out-neighbor of v_j on P. Thus, there is no path of backward arcs of length more than $k-1$.

If the in-degree of u_1 in $D[V(P)]$ is 2, remove one of the backward arcs terminating at u_1. Observe that now the backward arcs for P form a vertex-disjoint collection of out-trees with roots at vertices that are not terminal vertices of backward arcs. Therefore, the number of the out-trees in the collection is at most $2(k-2)(k-1)$. Observe that each out-tree in the collection has at most $k-1$

6

leaves and thus its arcs can be decomposed into at most $k-1$ paths, each of length at most k. Hence, the original total number of backward arcs for P is at most $2k(k-2)(k-1)^2 + 1$. On the other hand, it is at least $(q-1) - 2(k-2)(k-1)$. Thus, $(q-1) - 2(k-2)(k-1) \leq 2k(k-2)(k-1)^2 + 1$. Combining this inequality with $q \geq n/(k-1)$, we conclude that $n \leq 2k^5$. $\square$

Theorem 3.3. *Let D be a digraph in $\mathcal{L}$ with $\ell_s(D) > 0$.*

(a) *If D is an oriented graph with minimum in-degree at least 2, then $\ell_s(D) \geq (n/2)^{1/5} - 1$.*

(b) *If D is a digraph with minimum in-degree at least 3, then $\ell_s(D) \geq (n/2)^{1/5} - 1$.*

Proof: (a) Let T be an out-branching of D. Delete some arcs from $A(D) \backslash A(T)$, if needed, such that the in-degree of each vertex of D becomes 2. Now the inequality $\ell_s(D) \geq (n/2)^{1/5} - 1$ follows from Lemma 3.2 and the definition of $\mathcal{L}$.

(b) Let T be an out-branching of D. Let P be the path formed in the proof of Lemma 3.2. (Note that $A(P) \subseteq A(T)$.) Delete every double arc of P, in case there are any, and delete some more arcs from $A(D) \setminus A(T)$, if needed, to ensure that the in-degree of each vertex of D becomes 2. It is not difficult to see that the proof of Lemma 3.2 remains valid for the new digraph D. Now the inequality $\ell_s(D) \geq (n/2)^{1/5} - 1$ follows from Lemma 3.2 and the definition of $\mathcal{L}$. $\square$

It is not difficult to give examples showing that the restrictions on the minimum in-degrees in Theorem 3.3 are optimal. Indeed, any directed cycle C is a strong oriented graph with all in-degrees 1 for which $\ell_s(C) = 1$ and any directed double cycle D is a strong digraph with in-degrees 2 for which $\ell_s(D) = 2$ (a *directed double cycle* is a digraph obtained from an undirected cycle by replacing every edge xy with two arcs xy and yx).

4 Parameterized Algorithms for k-DMLOB and k-DMLOT

In the previous section, we gave lower bounds on $\ell(D)$ and $\ell_s(D)$ for digraphs $D \in \mathcal{L}$ with minimum in-degree at least 3. These bounds trivially imply the fixed parameter tractability of the k-DMLOB and the k-DMLOT problems for these class of digraphs. Here we extend these FPT results to digraphs in $\mathcal{L}$ for k-DMLOB and to all digraphs for k-DMLOT. We prove a decomposition theorem which either outputs an out-tree with k leaves or provides a path decomposition of the underlying undirected graph of width $O(k^3)$ in polynomial time.

Theorem 4.1. *Let D be a digraph in $\mathcal{L}$ with $\ell_s(D) > 0$. Then either $\ell_s(D) \geq k$ or the underlying undirected graph of D is of pathwidth at most k^3.*

Proof: Let D be a digraph in $\mathcal{L}$ with $0 < \ell_s(D) < k$. Let us choose an out-branching T of D with p leaves. As in the proof of Lemma 3.2, we obtain a collection $\mathcal{P}$ of $p \, (< k)$ vertex-disjoint directed paths covering all vertices of D.

For a path $P \in \mathcal{P}$, let $W(P)$ be the set of vertices not on P which are out-neighbors of vertices on P. If $|W(P)| \geq k$, then the vertices P and $W(P)$ would form an out-tree with at least k leaves, which by the definition of $\mathcal{L}$, contradicts the assumption $\ell_s(D) < k$. Therefore, $|W(P)| < k$. We define

$$U_1 = \{v \in W(P) : P \in \mathcal{P}\}.$$

Note that

$$|U_1| \leq p(k-1) \leq (k-1)^2.$$

Let D_1 be the graph obtained from D after applying the following trimming procedure around all vertices of U_1: for every path $P \in \mathcal{P}$ and every vertex $v \in U_1 \cap V(P)$ we delete all arcs emanating out of v and directed into v except those of the path P itself. Thus for every two paths $P, Q \in \mathcal{P}$ there is no arc in D_1 that goes from P to Q.

For $P \in \mathcal{P}$ let $D_1[P]$ be the subdigraph of D_1 induced by the vertices of P. Observe that P is a Hamiltonian directed path in $D_1[P]$. We denote by $S[P]$ the set of vertices which are heads of forward arcs in $D_1[P]$.

We claim that $|S[P]| \leq (k-2)(k-1)$. Indeed, for each vertex $v \in S[P]$, delete all forward arcs terminating at v but one. Observe that the procedure has not changed the number of vertices which are heads of forward arcs. Also the number of forward arcs in the new digraph is $|S[P]|$. As in the proof of Lemma 3.2, we can show that the number of forward arcs in the new digraph is at most $(k-2)(k-1)$.

Define $U_2 = \{v \in S[P] : P \in \mathcal{P}\}$. Thus

$$|U_2| \leq p(k-2)(k-1) \leq (k-2)(k-1)^2.$$

Let D_2 be the graph obtained from D_1 after applying the trimming procedure as before around all vertices of U_2, that is, for every path $P \in \mathcal{P}$ and every vertex $v \in U_2 \cap V(P)$ we delete all arcs emanating out of v or directed into v except those of the path P.

Put $U = U_1 \cup U_2$. Let C be a connectivity component of D_2. Observe that C consists of a directed path $P = v_1 v_2 \ldots v_q \in \mathcal{P}$ passing through all its vertices, together with its backward arcs. For every $j \in [q]$ let $V_j = \{v_i : i \in [j]\}$. If for some j the set V_j contained at least k vertices, say $\{v'_1, v'_2, \cdots, v'_t\}$ with $t \geq k$, having in-neighbors in the set $\{v_{j+1}, v_{j+2}, \ldots, v_q\}$, then D would contain an out-tree with at least k leaves formed by the path $v_{j+1} v_{j+2} \ldots v_q$ together with a backward arc terminating at v'_i from a vertex on the path for each $1 \leq i \leq t$, a contradiction. Thus, for the underlying undirected graph $UN(C)$ of C, $vs(UN(C)) \leq k$. By Proposition 2.2, the pathwidth of $UN(C)$ is at most k. Since the pathwidth of a graph is equal to the maximum pathwidth of its connected components, we have that the pathwidth of $UN(D_2)$ is at most k.

Finally, let $(X_1, X_2, \ldots, X_p)$ be a path decomposition of $UN(D_2)$ of width at most k. Then $(X_1 \cup U, X_2 \cup U, \ldots, X_p \cup U)$ is a path decomposition of $UN(D)$ of width at most $k + |U| \leq k^3$. $\qquad\square$

Theorem 4.2. *k-DMLOB is FPT for digraphs in $\mathcal{L}$.*

Proof Sketch: Let D be a digraph in $\mathcal{L}$ with $\ell_s(D) > 0$. The proof of Theorem 4.1 can be easily turned into a polynomial time algorithm to either build an out-branching of D with at least k leaves or to show that $pw(UN(D)) \leq k^3$ and provide the corresponding path decomposition. Now the algorithm follows by a simple dynamic programming over the decomposition. Alternatively, the property of containing a directed out-branching with at least k leaves can be formulated as a monadic second order formula. Thus, by the fundamental theorem of Courcelle [6, 7], the k-DMLOB problem for all digraphs D with $pw(UN(D)) \leq k^3$ can be solved in $O(f(k) \cdot |V(D)|)$ time, where f is a function depending only on k. $\qquad\square$

Let D be a digraph and let R_v be the set of vertices reachable from a vertex $v \in V(D)$ in D. Observe that D has an out-tree with k leaves if and only if there exists a $v \in V(D)$ such that $D[R_v]$ has an out-tree with k leaves. Notice that each $D[R_v]$ has an out-branching rooted at v. Thus, we can prove the following theorem, using the arguments in the previous proofs.

Theorem 4.3. *For a digraph D and $v \in V(D)$, let R_v be the set of vertices reachable from a vertex $v \in V(D)$ in D. Then either we have $\ell(D[R_v]) \geq k$ or the underlying undirected graph of $D[R_v]$ is of pathwidth at most k^3. Moreover, one can find, in polynomial time, either an out-tree with at least k leaves in $D[R_v]$, or a path decomposition of it of width at most k^3.*

To solve k-DMLOT, we apply Theorem 4.3 to all the vertices of D and then either apply dynamic programming over the decomposition or apply Courcelle's Theorem as in the proof of Theorem 4.2. This gives the following:

Theorem 4.4. *k-DMLOT is FPT for digraphs.*

We can, in fact, show that the k-DMLOB problem for digraphs in $\mathcal{L}$ is linear time solvable for a fixed k. To do so, given a digraph $D \in \mathcal{L}$ with $\ell_s(D) > 0$ we first apply Bodlaender's linear time algorithm [4] to check whether the treewidth of $UN(D)$ is at most k^3. If $tw(UN(D)) > k^3$ then by Theorem 4.1 D has an out-branching with k leaves. Else $tw(UN(D)) \leq k^3$ and we can use Courcelle's Theorem to check in linear time whether D has an out-branching of size k. This gives the following:

Theorem 4.5. *The k-DMLOB problem for digraphs in $\mathcal{L}$ is linear time solvable for every fixed k.*

5 Concluding Remarks and Open Problems

We have seen that every digraph $D \in \mathcal{L}$ with $\ell_s(D) > 0$ of order n and with minimum in-degree at least 3 contains an out-branching with at least $(n/2)^{1/5} - 1$ leaves. Combining the ideas in the proof of this combinatorial result with the fact that the problem of deciding whether a given digraph in $\mathcal{L}$ has an out-branching with at least k leaves can be solved efficiently for digraphs of pathwidth at most k^3 we have shown that the k-DMLOB problem for digraphs in $\mathcal{L}$ as well as the k-DMLOT

problem for general digraphs are fixed parameter tractable. The parameterized complexity of the k-DMLOB problem for all digraphs remains open.

For some subfamilies of $\mathcal{L}$, one can obtain better bounds on $\ell_s(D)$. An example is the class of multipartite tournaments. A multipartite tournament is an orientation of a complete multipartite graph. It is proved in [18, 27] that every multipartite tournament D with at most one source has an out-branching T such that the distance from the root of T to any vertex is at most 4. This implies that $\ell_s(D) \geq \frac{n-1}{4}$. Also for a tournament D of order n, it is easy to prove that $\ell_s(D) \geq n - \log_2 n$. (This bound is essentially tight, i.e., we cannot replace the right hand side by $n - \log_2 n + \Omega(\log_2 \log_2 n)$ as shown by random tournaments; see [1], pages 3-4, for more details.)

It seems that the bound $\ell_s(D) \geq (n/2)^{1/5} - 1$ is far from tight. It would be interesting to obtain better bounds for digraphs $D \in \mathcal{L}$ (with $\ell_s(D) > 0$) of minimum in-degree at least 3.

References

[1] N. Alon and J. Spencer, *The Probabilistic Method.* Wiley, NY, 2nd Ed., 2000.

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

[3] H.L. Bodlaender, On linear time minor tests and depth-first search. Journal of Algorithms **14** (1993), 1-23.

[4] H. L. Bodlaender, A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth. SIAM Journal on Computing **25(6)** (1996), 1305-1317.

[5] P.S. Bonsma, T. Brueggermann and G.J. Woeginger, A faster FPT algorithm for finding spanning trees with many leaves. Proc. of MFCS 2003, LNCS. **2747** (2003), 259-268.

[6] B. Courcelle, The Monadic second-order logic of graphs I: recognizable sets of finite graphs. Information and Computation **85(1)** (1990), 12-75.

[7] B. Courcelle, The monadic second-order logic of graphs III: tree-decompositions, minor and complexity issues. Informatique Théorique et Applications (ITA) **26** (1992), 257-286.

[8] G. Ding, Th. Johnson, and P. Seymour. Spanning trees with many leaves. Journal of Graph Theory **37** (2001), 189-197.

[9] R.G. Downey and M.R. Fellows, *Parameterized Complexity*, Springer-Verlag, (1999).

[10] M.R. Fellows and M.A. Langston, On well-partial-order theory and its applications to combinatorial problems of VLSI design. SIAM Journal on Discrete Mathematics **5** (1992), 117-126.

[11] M.R. Fellows, C. McCartin, F.A. Rosamond, and U. Stege, Coordinated kernels and catalytic reductions: An improved FPT algorithm for max leaf spanning tree and other problems. Proc. of FSTTCS 2000, LNCS. **1974** (2000), 240-251.

[12] M. Fellows, Private communications, 2005-2006.

[13] J. Flum and M. Grohe, *Parameterized Complexity Theory*. Springer-Verlag, (2006).

[14] G. Galbiati, F. Maffioli, and A. Morzenti, A short note on the approximability of the maximum leaves spanning tree problem. Information Processing Letters **52** (1994), 45-49.

[15] G. Galbiati, A. Morzenti, and F. Maffioli, On the approximability of some maximum spanning tree problems. Theoretical Computer Science 181 (1997), 107-118.

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

[17] J.R. Griggs and M. Wu, Spanning trees in graphs of minimum degree four or five. Discrete Mathematics **104** (1992), 167-183.

[18] G. Gutin, The radii of n-partite tournaments. *Math. Notes* **40** (1986), 743-744.

[19] G. Gutin and A. Yeo, Some Parameterized Problems on Digraphs. To appear in *The Computer Journal*.

[20] D.J. Kleitman and D.B. West, Spanning trees with many leaves. SIAM Journal on Discrete Mathematics **4** (1991), 99-106.

[21] N. G. Kinnersley, The vertex separation number of a graph equals its pathwidth, Information Processing Letters 42 (1992), 345-350.

[22] L. M. Kirousis and C. H. Papadimitriou, Interval graphs and searching, Discrete Mathematics 55 (1985), 181-184.

[23] N. Linial and D. Sturtevant (1987). Unpublished result.

[24] H.-I. Lu and R. Ravi, Approximating maximum leaf spanning trees in almost linear time. Journal of Algorithms **29** (1998), 132-141.

[25] R. H. Möhring, Graph problems related to gate matrix layout and PLA folding. In Computational graph theory, vol. 7 of Comput. Suppl., Springer, Vienna, (1990), 17-51.

[26] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and Its Applications, Oxford University Press, (2006).

[27] V. Petrovic and C. Thomassen, Kings in k-partite tournaments. *Discrete Math.* **98** (1991), 237-238.

[28] R. Solis-Oba, 2-approximation algorithm for finding a spanning tree with the maximum number of leaves. Proc. ESA 1998, LNCS. **1461** (1998), 441-452.

[29] N. Robertson and P. D. Seymour, Graph minors-a survey. In I. Anderson, editor, *Surveys in Combinatorics*, Cambridge Univ. Press, (1985), 153-171.

[30] N. Robertson and P. D. Seymour, Graph minors I: Excluding a forest. Journal of Combinatorial Theory Series B, **35** (1983), 39-61.

[31] N. Robertson and P. D. Seymour, Graph minors II: Algorithmic aspects of tree-width. Journal of Algorithms, **7** (1986), 309-322.

[32] B. Y. Wu and K. Chao, *Spanning Trees and Optimization Problems*. CRC Press, 2003.