Better Algorithms and Bounds for Directed Maximum Leaf Problems

Noga Alon[1], Fedor V. Fomin[2], Gregory Gutin[3], Michael Krivelevich[1], and Saket Saurabh[2,4]

[1] Department of Mathematics, Tel Aviv University
Tel Aviv 69978, Israel
{nogaa,krivelev}@post.tau.ac.il
[2] Department of Informatics, University of Bergen
POB 7803, 5020 Bergen, Norway
{fedor.fomin,saket}@ii.uib.no
[3] Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
gutin@cs.rhul.ac.uk
[4] The Institute of Mathematical Sciences
Chennai, 600 017, India
saket@imsc.res.in

Abstract. The Directed Maximum Leaf Out-Branching problem is to find an out-branching (i.e. a rooted oriented spanning tree) in a given digraph with the maximum number of leaves. In this paper, we improve known parameterized algorithms and combinatorial bounds on the number of leaves in out-branchings. We show that

- every strongly connected digraph D of order n with minimum in-degree at least 3 has an out-branching with at least $(n/4)^{1/3} - 1$ leaves;
- if a strongly connected digraph D does not contain an out-branching with k leaves, then the pathwidth of its underlying graph is $O(k \log k)$;
- it can be decided in time $2^{O(k \log^2 k)} \cdot n^{O(1)}$ whether a strongly connected digraph on n vertices has an out-branching with at least k leaves.

All improvements use properties of extremal structures obtained after applying local search and of some out-branching decompositions.

1 Introduction

Given a digraph D, a subdigraph T of D is an *out-tree* if T is an oriented tree with only one vertex s of in-degree zero (called *the root*) and if T is a spanning out-tree, i.e. $V(T) = V(D)$, then T is called an *out-branching* of D. The vertices of T of out-degree zero are called *leaves*. The Directed Maximum Leaf Out-Branching problem is to find an out-branching in a given digraph with the maximum number of leaves. This problem is a natural generalization of the intensively studied Maximum Leaf Spanning Tree problem on connected undirected graphs [5, 7, 9–11, 13, 15, 20, 22]. Unlike its undirected counterpart which has attracted a lot of attention in all algorithmic paradigms like approximation algorithms [13, 20, 22], parameterized algorithms [5, 9, 11], exact exponential time algorithms [10] and also combinatorial studies [7, 15, 16, 19], the Directed Maximum Leaf Out-Branching problem has largely been neglected until recently.

In [2] we initiated algorithmic and combinatorial study of Directed Maximum Leaf Out-Branching and obtained, as the main result of the paper, the first fixed parameter tractable algorithms for the problem on strongly connected digraphs and acyclic digraphs based on various combinatorial lemmas. In this paper we continue our investigation of the Directed Maximum Leaf Out-Branching (DMLOB) and obtain several improved parameterized algorithms for the problem as well as combinatorial results regarding the number of leaves possible in an out-branching of a digraph based on completely new approaches and ideas which are interesting in its own and could be useful for solving other problems on digraphs.

In parameterized algorithms, for decision problems with input size n, and a parameter k, the goal of is to design an algorithm with runtime $f(k)n^{O(1)}$, where f is a function of k alone. (For DMLOB such a parameter is the number of leaves in the out-tree.) Problems having such an algorithm are said to be fixed parameter tractable (FPT). The book by Downey and Fellows [8] provides an introduction to the topic of parameterized complexity. For recent developments see the books by Flum and Grohe [12] and by Niedermeier [21].

The parameterized version of DMLOB is defined as follows: Given a digraph D and a positive integral parameter k, does there exist an out-branching with at least k leaves? We denote the parameterized versions of DMLOB by k-DMLOB. If in the above definition we do not insist for an out-branching and ask whether there exists an out-tree with at least k leaves, we get parameterized Directed Maximum Leaf Out-Tree problem (denoted k-DMLOT).

In this paper we obtain the following new algorithmic and combinatorial results on k-DMLOB for strongly connected digraphs and acyclic digraphs. Before we go any further we remark that the algorithmic results presented here also hold for *all digraphs* if we consider k-DMLOT rather than k-DMLOB. However, we mainly restrict ourselves to k-DMLOB for clarity and the harder challenges it poses, and we briefly consider k-DMLOT only in the last section.

Faster Algorithm. We design a new algorithm which decides in time $2^{O(k \log^2 k)} \cdot n^{O(1)}$ whether a strongly connected digraph on n vertices has an out-branching with at least k leaves (Corollary 2). On acyclic graphs we can solve the problem even faster, in time $2^{O(k \log k)} \cdot n^{O(1)}$ (Corollary 1). These are significant improvements over running time $2^{O(k^2 \log k)} \cdot n^{O(1)}$ for both classes of digraphs obtained in [2]. The improvements do not result from a careful tuning of the algorithm from [2] but from several novel ideas. In particular, we use local search and specific tree partition arguments. While local search is a widely used technique in heuristics and approximation algorithms (see, e.g., [1]) we are not aware of its applications in parameterized complexity. We find it to be of independent interest.

Combinatorial bounds. Kleitman and West [16] and Linial and Sturtevant [19] 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. In [2] we proved an analogue of this result for directed graphs: every strongly connected digraph D of order n with minimum in-degree at least 3 has an out-branching with at least $(n/2)^{1/5}-1$ leaves. In this paper (Theorem 4), we improve this bound to $(n/4)^{1/3}-1$. We do not know whether the last bound is tight, however we show that there are strongly connected digraphs with minimum in-degree 3 in which every out-branching has at most $O(\sqrt{n})$ leaves (Theorem 6). Another parallel between the worlds of directed and undirected graphs established in this paper (and used intensively in the algorithmic part) is the relation between the number of leaves in a maximum leaf out-branching in a digraph D and the pathwidth of its underlying graph. It is easy to check (see, e.g., [4]), that every connected undirected graph of pathwidth at least k, contains a spanning tree with at least k leaves. We show (Theorem 8) that if a strongly connected digraph D does not contain an out-branching with k leaves, then the pathwidth of its underlying graph is $O(k \log k)$.

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 orientations 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 digraph D is *strongly connected* if, for every pair x, y of vertices there are directed paths from x to y and from y to x. A maximal strongly connected subdigraph of D is called a *strong component*. A vertex u of D is an *in-neighbor* (*out-neighbor*) of a vertex v if $uv \in A(D)$ ($vu \in A(D)$, respectively). The *in-degree* $d^-(v)$ (*out-degree* $d^+(v)$) of a vertex v is the number of its in-neighbors (out-neighbors).

We denote by $\ell(D)$ the maximum number of leaves in an out-tree of a digraph D and by $\ell_s(D)$ we denote the maximum possible number of leaves in an out-branching of a digraph D. When D has no out-branching, we write $\ell_s(D)=0$. The following simple result gives necessary and sufficient conditions for a digraph to have an out-branching. This assertion allows us to check whether $\ell_s(D)>0$ in time $O(|V(D)|+|A(D)|)$.

Proposition 1 ([3]). *A digraph D has an out-branching if and only if D has a unique strong component with no incoming arcs.*

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

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

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 $\sigma = (v_1, v_2, \dots, v_n)$ 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 [18] on interval width of a graph, see also [17].

Proposition 2 ([17, 18]). *For any graph G, $vs(G) = pw(G)$.*

3 Locally Optimal Out-Trees

Our improved parameterized algorithms are based on finding locally optimal out-branchings. Given a digraph, D and an out-branching T, we call a vertex *leaf*, *link* and *branch* if its out-degree in T is 0, 1 and ≥ 2 respectively. Let $S^+_{\ge 2}(T)$ be the set of branch vertices, $S^+_1(T)$ be the set of link vertices and $L(T)$ be the set of leaves in the tree T. Let $\mathscr{P}_2(T)$ be the set of maximal paths consisting of link vertices. By $p(v)$ we denote the *parent* of a vertex v in T; $p(v)$ is the unique in-neighbor of v. We call a pair of vertices u and v *siblings* if they do not belong to the same path from the root r in T. We start with the following well known and easy to observe facts.

Fact 1 $|S^+_{\ge 2}(T)| \le |L(T)| - 1$.

Fact 2 $|\mathscr{P}_2(T)| \le 2|L(T)| - 1$.

Now we define the notion of local exchange which is intensively used in our proofs.

Definition 3 ℓ-Arc Exchange (ℓ-AE) optimal out-branching: *An out-branching T of a directed graph D with k leaves is ℓ-AE optimal if for all arc subsets $F \subseteq A(T)$ and $X \subseteq A(D) - A(T)$ of size ℓ, $(A(T) \setminus F) \cup X$ is either not an out-branching, or an out-branching with $\le k$ leaves. In other words, T is ℓ-AE optimal if it can't be turned into an out-branching with more leaves by exchanging ℓ arcs.*

Let us remark, that for every fixed ℓ, an ℓ-AE optimal out-branching can be obtained in polynomial time. In our proofs we use only 1-AE optimal out-branchings. We need the following simple properties of 1-AE optimal out-branchings.

Lemma 1. *Let T be an 1-AE optimal out-branching rooted at r in a digraph D. Then the following holds:*

(a) For every pair of siblings $u, v \in V(T) \setminus L$ with $d_T^+(p(v)) = 1$, there is no arc $e = (u, v) \in A(D) \setminus A(T)$;
(b) For every pair of vertices $u, v \notin L$, $d_T^+(p(v)) = 1$, which are on the same path from the root with $dist(r, u) < dist(r, v)$ there is no arc $e = (u, v) \in A(D) \setminus A(T)$ (here $dist(r, u)$ is the distance to u in T from the root r);
(c) There is no arc (v, r), $v \notin L$ such that the directed cycle formed by the (r, v)-path and the arc (v, r) contains a vertex x such that $d_T^+(p(x)) = 1$.

4 Combinatorial Bounds

We start with a lemma that allows us to obtain lower bounds on $\ell_s(D)$.

Lemma 2. *Let D be a oriented graph of order n in which every vertex is of in-degree 2 and let D have an out-branching. If D has no out-tree with k leaves, then $n \le 4k^3$.*

Proof. Let us assume that D has no out-tree with k leaves. Consider an out-branching T of D with $p < k$ leaves which is 1-AE optimal. Let r be the root of T.

We will bound the number n of vertices in T as follows. Every vertex of T is either a leaf, or a branch vertex, or a link vertex. By Facts 1 and 2 we already have bounds on the number of leaf and branch vertices as well as the number of maximal paths consisting of link vertices. So to get an upper bound on n in terms of k, it suffices to bound the length of each maximal path consisting of link vertices. Let us consider such a path P and let x, y be the first and last vertices of P, respectively.

The vertices of $V(T) \setminus V(P)$ can be partitioned into four classes as follows:

(a) ancestor vertices: the vertices which appear before x on the (r, x)-path of T;
(b) descendant vertices : the vertices appearing after the vertices of P on paths of T starting at r and passing through y;
(c) sink vertices: the vertices which are leaves but not descendant vertices;
(d) special vertices: none-of-the-above vertices.

Let $P' = P - x$, let z be the out-neighbor of y on T and let T_z be the subtree of T rooted at z. By Lemma 1, there are no arcs from special or ancestor vertices to the path P'. Let uv be an arc of $A(D) \setminus A(P')$ such that $v \in V(P')$. There are two possibilities for u: (i) $u \notin V(P')$, (ii) $u \in V(P')$ and uv is backward for P' (there are no forward arcs for P' since T is 1-AE optimal). Note that every vertex of type (i) is either a descendant vertex or a sink. Observe also that 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 for P'. These roots are terminal vertices of arcs in which first vertices are descendant vertices or sinks.

We denote by $\{u_1, u_2, \ldots, u_s\}$ and $\{v_1, v_2, \ldots, v_t\}$ the sets of vertices on P' which have out-neighbors that are descendant vertices and sinks, respectively. Let the out-tree formed by backward arcs for P' rooted at $w \in \{u_1, \ldots, u_s, v_1, \ldots, v_t\}$ be denoted by $T(w)$ and let $l(w)$ denote the number of leaves in $T(w)$. Observe that the following is an out-tree rooted at z:

$$T_z \cup \{(in(u_1), u_1), \ldots, (in(u_s), u_s)\} \cup \bigcup_{i=1}^{s} T(u_i),$$

where $\{in(u_1), \ldots, in(u_s)\}$ are the in-neighbors of $\{u_1, \ldots, u_s\}$ on T_z. This out-tree has at least $\sum_{i=1}^{s} l(u_i)$ leaves and, thus, $\sum_{i=1}^{s} l(u_i) \leq k-1$. Let us denote the subtree of T rooted at x by T_x and let $\{in(v_1), \ldots, in(v_t)\}$ be the in-neighbors of $\{v_1, \ldots, v_t\}$ on $T - V(T_x)$. Then we have following out-tree:

$$(T - V(T_x)) \cup \{(in(v_1), v_1), \ldots, (in(v_t), v_t)\} \cup \bigcup_{i=1}^{t} T(v_i)$$

with at least $\sum_{i=1}^{t} l(v_i)$ leaves. Thus, $\sum_{i=1}^{t} l(v_i) \leq k-1$.

Consider a path $R = v_0 v_1 \ldots v_r$ 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 $k-1$. Every out-tree $T(w)$, $w \in \{u_1, \ldots, u_s\}$ has $l(w)$ leaves and, thus, its arcs can be decomposed into $l(w)$ paths, each of length at most $k-1$. Now we can bound the number of arcs in all the trees $T(w)$, $w \in \{u_1, \ldots, u_s\}$, as follows: $\sum_{i=1}^{s} l(u_i)(k-1) \leq (k-1)^2$. We can similarly bound the number of arcs in all the trees $T(w)$, $w \in \{v_1, \ldots, v_s\}$ by $(k-1)^2$. Recall that the vertices of P' can be either terminal vertices of backward arcs for P' or vertices in $\{u_1, \ldots, u_s, v_1, \ldots, v_t\}$. Observe that $s + t \leq 2(k-1)$ since $\sum_{i=1}^{s} l(u_i) \leq k-1$ and $\sum_{i=1}^{t} l(v_i) \leq k-1$.

Thus, the number of vertices in P is bounded from above by $1 + 2(k-1) + 2(k-1)^2$. Therefore,

$$\begin{aligned} n &= |L(T)| + |S^+_{\geq 2}(T)| + |S^+_1(T)| \\ &= |L(T)| + |S^+_{\geq 2}(T)| + \sum_{P \in \mathscr{P}_2(T)} |V(P)| \\ &\leq (k-1) + (k-2) + (2k-3)(2k^2 - 2k + 1) \\ &< 4k^3. \end{aligned}$$

Thus, we conclude that $n \leq 4k^3$. □

Theorem 4. *Let D be a strongly connected digraph with n vertices.*

(a) If D is an oriented graph with minimum in-degree at least 2, then $\ell_s(D) \geq (n/4)^{1/3} - 1$.
(b) If D is a digraph with minimum in-degree at least 3, then $\ell_s(D) \geq (n/4)^{1/3} - 1$.

Proof. Since D is strongly connected, we have $\ell(D) = \ell_s(D) > 0$. Let T be an 1-AE optimal out-branching of D with maximum number of leaves. (a) Delete some arcs from $A(D) \setminus A(T)$, if needed, such that the in-degree of each vertex of D becomes 2. Now the inequality $\ell_s(D) \geq (n/4)^{1/3} - 1$ follows from Lemma 2 and the fact that $\ell(D) = \ell_s(D)$.

(b) Let P be the path formed in the proof of Lemma 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 2 remains valid for the new digraph D. Now the inequality $\ell_s(D) \geq (n/4)^{1/3} - 1$ follows from Lemma 2 and the fact that $\ell(D) = \ell_s(D)$. □

Remark 5 *It is easy to see that Theorem 4 holds also for acyclic digraphs D with $\ell_s(D) > 0$.*

While we do not know whether the bounds of Theorem 4 are tight, we can show that no linear bounds are possible. The following result is formulated for Part (b) of Theorem 4, but a similar result holds for Part (a) as well.

Theorem 6. *For each $t \geq 6$ there is a strong digraph H_t of order $n = t^2 + 1$ with minimum in-degree 3 such that $0 < \ell_s(H_t) = O(t)$.*

Proof. Let $V(H_t) = \{r\} \cup \{u^i_1, u^i_2, \ldots, u^i_t \mid i \in [t]\}$ and

$$\begin{aligned} A(H_t) = &\left\{u^i_j u^i_{j+1}, u^i_{j+1} u^i_j \mid \ i \in [t], j \in \{0, 1, \ldots, t-3\}\right\} \\ &\bigcup \left\{u^i_j u^i_{j-2} \mid i \in [t], j \in \{3, 4, \ldots, t-2\}\right\} \\ &\bigcup \left\{u^i_j u^i_q \mid \ i \in [t], t-3 \leq j \neq q \leq t\right\}, \end{aligned}$$

where $u^i_0 = r$ for every $i \in [t]$. It is easy to check that $0 < \ell_s(H_t) = O(t)$. □

5 Decomposition Algorithms

Theorem 7. *Let D be an acyclic digraph with a single vertex of in-degree zero. Then either $\ell_s(D) \geq k$ or the underlying undirected graph of D is of pathwidth at most $4k$ and we can obtain this path decomposition in polynomial time.*

Proof. Assume that $\ell_s(D) \leq k-1$. Consider a 1-AE optimal out-branching T of D. Notice that $|L(T)| \leq k-1$. Now remove all the leaves and branch vertices from the tree T. The remaining vertices form maximal directed paths consisting of link vertices. Delete the first vertices of all paths. As a result we obtain a collection $\mathcal{Q}$ of directed paths. Let $H = \cup_{P\in\mathcal{Q}} P$. We will show that every arc uv with $u, v \in V(H)$ is in H.

Let $P' \in \mathcal{Q}$. As in the proof of Lemma 2, we see that there are no forward arcs for P'. Since D is acyclic, there are no backward arcs for P'. Suppose uv is an arc of D such that $u \in R'$ and $v \in P'$, where R' and P' are distinct paths from $\mathcal{Q}$. As in the proof of Lemma 2, we see that u is either a sink or a descendent vertex for P' in T. Since R' contains no sinks of T, u is a descendent vertex, which is impossible as D is acyclic. Thus, we have proved that $pw(UN(H)) = 1$.

Consider a path decomposition of H of width 1. We can obtain a path decomposition of $UN(D)$ by adding all the vertices of $L(T) \cup S^+_{\geq 2}(T) \cup F(T)$, where $F(T)$ is the set of first vertices of maximal directed paths consisting of link vertices of T, to each of the bags of a path decomposition of H of width 1. Observe that the pathwidth of this decomposition is bounded from above by

$$|L(T)| + |S^+_{\geq 2}(T)| + |F(T)| + 1 \leq (k-1) + (k-2) + (2k-4) + 1 \leq 4k-6.$$

The bounds on the various sets in the inequality above follows from Facts 1 and 2. This proves the theorem. □

Corollary 1. *For acyclic digraphs, the problem k-DMLOB can solved in time $2^{O(k\log k)} \cdot n^{O(1)}$.*

Proof. The proof of Theorem 7 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 $pw(UN(D)) \leq 4k$ and provide the corresponding path decomposition. A simple dynamic programming over the path decomposition gives us an algorithm of running time $2^{O(k\log k)} \cdot n^{O(1)}$. □

The following lemma is well known, see, e.g., [6].

Lemma 3. *Let $T = (V, E)$ be an undirected tree and let $w : V \to \mathbb{R}^+ \cup \{0\}$ be a weight function on its vertices. There exists a vertex $v \in T$ such that the weight of every subtree T' of $T - v$ is at most $w(T)/2$, where $w(T) = \sum_{v\in V} w(v)$.*

Let D be a digraph with $\ell_s(D) = \lambda$ and let T be an out-branching of D with λ leaves. Consider the following decomposition of T (called a *β-decomposition*) which is useful in the proof of Theorem 8.

Assign weight 1 to all leaves of T and weight 0 to all non-leaves of T. By Lemma 3, T has a vertex v such that each component of $T - v$ has at most $\lambda/2 + 1$ leaves (if v is not the root and its in-neighbor v^- in T is a link vertex, then v^- becomes a new leaf). Let $T_1, T_2, \ldots, T_s$ be the components of $T - v$ and let $l_1, l_2, \ldots, l_s$ be the numbers of leaves in the components. Notice that $\lambda \le \sum_{i=1}^{s} l_i \le \lambda + 1$ (we may get a new leaf). We may assume that $l_s \le l_{s-1} \le \cdots \le l_1 \le \lambda/2 + 1$. Let j be the first index such that $\sum_{i=1}^{j} l_i \ge \frac{\lambda}{2} + 1$. Consider two cases: (a) $l_j \le (\lambda + 2)/4$ and (b) $l_j > (\lambda + 2)/4$. In Case (a), we have

$$\frac{\lambda+2}{2} \le \sum_{i=1}^{j} l_i \le \frac{3(\lambda+2)}{4} \text{ and } \frac{\lambda-6}{4} \le \sum_{i=j+1}^{s} l_i \le \frac{\lambda}{2}.$$

In Case (b), we have $j = 2$ and

$$\frac{\lambda+2}{4} \le l_1 \le \frac{\lambda+2}{2} \text{ and } \frac{\lambda-2}{2} \le \sum_{i=2}^{s} l_i \le \frac{3\lambda+2}{4}.$$

Let $p = j$ in Case (a) and $p = 1$ in Case (b). Add to D and T a *copy* v' of v (with the same in- and out-neighbors). Then the number of leaves in each of the out-trees

$$T' = T[\{v\} \cup (\cup_{i=1}^{p} V(T_i))] \text{ and } T'' = T[\{v'\} \cup (\cup_{i=p+1}^{s} V(T_i))]$$

is between $\lambda(1+o(1))/4$ and $3\lambda(1+o(1))/4$. Observe that the vertices of T' have at most $\lambda + 1$ out-neighbors in T'' and the vertices of T'' have at most $\lambda + 1$ out-neighbors in T' (we add 1 to λ due to the fact that v 'belongs' to both T' and T'').

Similarly to deriving T' and T'' from T, we can obtain two out-trees from T' and two out-trees from T'' in which the numbers of leaves are approximately between a quarter and three quarters of the number of leaves in T' and T'', respectively. Observe that after $O(\log \lambda)$ 'dividing' steps, we will end up with $O(\lambda)$ out-trees with just one leaf, i.e., directed paths. These paths contain $O(\lambda)$ copies of vertices of D (such as v' above). After deleting the copies, we obtain a collection of $O(\lambda)$ disjoint directed paths covering $V(D)$.

Theorem 8. *Let D be a strongly connected digraph. Then either $\ell_s(D) \ge k$ or the underlying undirected graph of D is of pathwidth $O(k \log k)$.*

Proof. We may assume that $\ell_s(D) < k$. Let T be be a 1-AE optimal out-branching. Consider a β-decomposition of T. The decomposition process can be viewed as a tree $\mathcal{T}$ rooted in a node (associated with) T. The sons of T in

$\mathcal{T}$ are nodes (associated with) T' and T''; the leaves of $\mathcal{T}$ are the directed paths of the decomposition. The *first layer* of $\mathcal{T}$ is the note T, the *second layer* are T' and T'', the *third layer* are sons of T' and T'', etc. Assume that $\mathcal{T}$ has t layers. Notice that the last layer consists of (some) leaves of $\mathcal{T}$ and that $t = O(\log k)$, which was proved above ($k \leq \lambda - 1$).

Let Q be a node of $\mathcal{T}$ at layer j. We will prove that

$$pw(UN(D[V(Q)])) < 2(t - j + 2.5)k \tag{1}$$

Since $t = O(\log k)$, (1) for $j = 1$ implies that the underlying undirected graph of D is of pathwidth $O(k \log k)$.

We first prove (1) for $j = t$ when Q is a path from the decomposition. Let $W = (L(T) \cup S^+_{\geq 2}(T) \cup F(T)) \cap V(Q)$, where $F(T)$ is the set of first vertices of maximal paths of T consisting of link vertices. As in the proof of Theorem 7, it follows from Facts 1 and 2 that $|W| < 4k$. Obtain a digraph R by deleting from $D[V(Q)]$ all arcs in which at least one end-vertex is in W and which are not arcs of Q. As in the proof of Theorem 7, it follows from Lemma 1 and 1-AE optimality of T that there are no forward arcs for Q in R. Let $Q = v_1 v_2 \dots v_q$. For every $j \in [q]$, let $V_j = \{v_i :\ i \in [j]\}$. If for some j the set V_j contained k vertices, say $\{v'_1, v'_2, \cdots, v'_k\}$, having in-neighbors in the set $\{v_{j+1}, v_{j+2}, \dots, v_q\}$, then D would contain an out-tree with k leaves formed by the path $v_{j+1}v_{j+2}\dots v_q$ together with a backward arc terminating at v'_i from a vertex on the path for each $1 \leq i \leq k$, a contradiction. Thus $vs(UN(D_2[P])) \leq k$. By Proposition 2, the pathwidth of $UN(R)$ is at most k. Let $(X_1, X_2, \dots, X_s)$ be a path decomposition of $UN(R)$ of width at most k. Then $(X_1 \cup W, X_2 \cup W, \dots, X_s \cup W)$ is a path decomposition of $UN(D[V(Q)])$ of width less than $k + 4k$. Thus,

$$pw(UN(D[V(Q)])) < 5k \tag{2}$$

Now assume that we have proved (1) for $j = i$ and show it for $j = i - 1$. Let Q be a node of layer $i - 1$. If Q is a leaf of $\mathcal{T}$, we are done by (2). So, we may assume that Q has sons Q' and Q'' which are nodes of layer i. In the β-decomposition of T given before this theorem, we saw that the vertices of T' have at most $\lambda + 1$ out-neighbors in T'' and the vertices of T'' have at most $\lambda + 1$ out-neighbors in T'. Similarly, we can see that (in the β-decomposition of this proof) the vertices of Q' have at most k out-neighbors in Q'' and the vertices of Q'' have at most k out-neighbors in Q' (since $k \leq \lambda - 1$). Let Y denote the set of the above-mentioned out-neighbors on Q' and Q''; $|Y| \leq 2k$. Delete from $D[V(Q') \cup V(Q'')]$ all arcs in which at least one end-vertex is in Y and which do not belong to $Q' \cup Q''$

Let G denote the obtained digraph. Observe that G is disconnected and $G[V(Q')]$ and $G[V(Q'')]$ are components of G. Thus, $pw(UN(G)) \leq b$, where

$$b = \max\{pw(UN(G[V(Q')])), pw(UN(G[V(Q'')]))\} < 2(t - i + 4.5)k \tag{3}$$

Let $(Z_1, Z_2, \ldots, Z_r)$ be a path decomposition of G of width at most b. Then $(Z_1 \cup Y, Z_2 \cup Y, \ldots, Z_r \cup Y)$ is a path decomposition of $UN(D[V(Q') \cup V(Q'')])$ of width at most $b + 2k < 2(t - i + 2.5)k$. □

Similar to the proof of Corollary 1, we obtain the following:

Corollary 2. *For a strongly connected digraph D, the problem k-DMLOB can be solved in time $2^{O(k \log^2 k)} \cdot n^{O(1)}$.*

6 Discussion and Open Problems

In this paper, we continued algorithmic and combinatorial investigation of the Directed Maximum Leaf Out-Branching problem. In particular, we showed that for every strongly connected digraph D of order n and with minimum in-degree at least 3, $\ell_s(D) = \Omega(n^{1/3})$. The most interesting open combinatorial question here is whether this bound is tight. It would be even more interesting to find the maximum number r such that $\ell_s(D) = \Omega(n^r)$ for every strongly connected digraph D of order n and with minimum in-degree at least 3. It follows from our results that $\frac{1}{3} \le r \le \frac{1}{2}$.

We also provided an algorithm of time complexity $2^{O(k \log^2 k)} \cdot n^{O(1)}$ which solves k-DMLOB for a strongly connected digraph D. The algorithm is based on a combinatorial bound on the pathwidth of the underlying undirected graph of D. Unfortunately, this technique does not work on all digraphs. It remains an algorithmic challenge to establish the parameterized complexity of k-DMLOB on all digraphs.

Notice that $\ell(D) \ge \ell_s(D)$ for each digraph D. Let $\mathcal{L}$ be the family of digraphs D for which either $\ell_s(D) = 0$ or $\ell_s(D) = \ell(D)$. The following assertion shows that $\mathcal{L}$ includes a large number digraphs including all strongly connected digraphs and acyclic digraphs (and, also, well-studied classes of semicomplete multipartite digraphs and quasi-transitive digraphs, see [3] for the definitions).

Proposition 3 ([2]). *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 $D \in \mathcal{L}$.*

Let $\mathcal{B}$ be the family of digraphs that contain out-branchings. The results of this paper proved for strongly connected digraphs can be extended to the class $\mathcal{L} \cap \mathcal{B}$ of digraphs since in the proofs we use only the following property of strongly connected digraphs D: $\ell_s(D) = \ell(D) > 0$.

For a digraph D and a vertex v, let D_v denote the subdigraph of D induced by all vertices reachable from v. Using the $2^{O(k \log^2 k)} \cdot n^{O(1)}$ algorithm for k-DMLOB on digraphs in $\mathcal{L} \cap \mathcal{B}$ and the facts that (i) $D_v \in \mathcal{L} \cap \mathcal{B}$ for each digraph D and vertex v and (ii) $\ell(D) = \max\{\ell_s(D_v) | v \in V(D)\}$ (for details, see [2]), we can obtain an $2^{O(k \log^2 k)} \cdot n^{O(1)}$ algorithm for k-DMLOT on *all* digraphs. For acyclic digraphs, the running time can be reduced to $2^{O(k \log k)} \cdot n^{O(1)}$.

References

1. E. Aarts and J. K. Lenstra, editors. *Local search in combinatorial optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Ltd., Chichester, 1997. A Wiley-Interscience Publication.
2. N. Alon, F. V. Fomin, G. Gutin, M. Krivelevich and S. Saurabh, Parameterized Algorithms for Directed Maximum Leaf Problems. *Proc. ICALP 2007*, LNCS **4596** (2007), 352?-362.
3. J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*. Springer-Verlag, 2000.
4. D. Bienstock, N. Robertson, P. D. Seymour, and R. Thomas. Quickly excluding a forest. *J. Comb. Theory Series B*, 52:274–283, 1991.
5. P.S. Bonsma, T. Brueggermann and G.J. Woeginger, A faster FPT algorithm for finding spanning trees with many leaves. Lect. Notes Computer Sci. 2747 (2003), 259–268.
6. F.R.K. Chung, Separator theorems and their applications, In *Paths, flows, and VLSI-layout (Bonn, 1988)*, Series *Algorithms Combin.*, vol. 9 (1990), 17–34, Springer, Berlin.
7. G. Ding, Th. Johnson, and P. Seymour. Spanning trees with many leaves. Journal of Graph Theory 37 (2001), 189–197.
8. R.G. Downey and M.R. Fellows, *Parameterized Complexity*, Springer-Verlag, 1999.
9. V. Estivill-Castro, M.R. Fellows, M.A. Langston, and F.A. Rosamond, FPT is P-Time Extremal Structure I. Proc. ACiD (2005), 1–41.
10. F. V. Fomin, F. Grandoni, D. Kratsch, Solving Connected Dominating Set Faster Than 2^n. Lect. Notes Comput. Sci. 4337 (2006), 152–163.
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. Lect. Notes Comput. Sci. 1974 (2000), 240–251.
12. J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer-Verlag, 2006.
13. G. Galbiati, A. Morzenti, and F. Maffioli, On the approximability of some maximum spanning tree problems. Theoretical Computer Science 181 (1997), 107–118.
14. M.R. Garey and D.S. Johnson, *Computers and Intractability*, W.H. Freeman and Co., New York, 1979.
15. J.R. Griggs and M. Wu, Spanning trees in graphs of minimum degree four or five. Discrete Mathematics 104 (1992), 167–183.
16. D.J. Kleitman and D.B. West, Spanning trees with many leaves. SIAM Journal on Discrete Mathematics 4 (1991), 99–106.
17. N. G. Kinnersley, The vertex separation number of a graph equals its path-width, Information Processing Letters 42 (1992), 345–350.
18. L. M. Kirousis and C. H. Papadimitriou, Interval graphs and searching, Discrete Mathematics 55 (1985), 181–184.
19. N. Linial and D. Sturtevant (1987). Unpublished result.
20. H.-I. Lu and R. Ravi, Approximating maximum leaf spanning trees in almost linear time. Journal of Algorithms 29 (1998), 132–141.
21. R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, 2006.
22. R. Solis-Oba, 2-approximation algorithm for finding a spanning tree with the maximum number of leaves. Lect. Notes Comput. Sci. 1461 (1998), 441–452.