Linear Kernels for Outbranching Problems in Sparse Digraphs*

Marthe Bonamy[1], Łukasz Kowalik[2], Michał Pilipczuk[2], and Arkadiusz Socała[2]

1 LIRMM, France
marthe.bonamy@lirmm.fr
2 University of Warsaw, Poland
kowalik@mimuw.edu.pl, michal.pilipczuk@mimuw.edu.pl, arkadiusz.socala@students.mimuw.edu.pl

Abstract

In the k-Leaf Out-Branching and k-Internal Out-Branching problems we are given a directed graph D with a designated root r and a nonnegative integer k. The question is to determine the existence of an outbranching rooted at r that has at least k leaves, or at least k internal vertices, respectively. Both these problems were intensively studied from the points of view of parameterized complexity and kernelization, and in particular for both of them kernels with $O(k^2)$ vertices are known on general graphs. In this work we show that k-Leaf Out-Branching admits a kernel with $O(k)$ vertices on H-minor-free graphs, for any fixed H, whereas k-Internal Out-Branching admits a kernel with $O(k)$ vertices on any graph class of bounded expansion.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases FPT algorithm, kernelization, outbranching, sparse graphs

Digital Object Identifier 10.4230/LIPIcs.IPEC.2015.199

1 Introduction

In this work we are interested in kernelization algorithms for two problems investigated by Dorn et al. [4], namely k-Leaf Out-Branching (LOB) and k-Internal Out-Branching (IOB). In both problems, we are given a directed graph D with a specified root r and a nonnegative integer k. By an *outbranching rooted at* r we mean a spanning tree of D with all the edges oriented away from r. A vertex of D is a *leaf* in an outbranching T if it has outdegree 0 in T, and is *internal* otherwise. In LOB the question is to verify the existence of an outbranching rooted at r that has at least k leaves, whereas in IOB we instead ask for an outbranching rooted at r with at least k internal vertices. Both problems enjoy the existence of kernels with $O(k^2)$ vertices on general graphs [2, 9], however up to this work no better kernels were known even in the case of planar graphs. Although many problems for planar graphs admit kernels of linear size by a general framework of bidimensionality (see [7]), the directed nature of both problems studied here prevents them from satisfying even the most basic properties needed for the bidimensionality tools to be applicable.

* Work partially supported by the ANR Grant EGOS (2012-2015) 12 JS02 002 01 (MB), by the National Science Centre of Poland, grants number 2013/09/B/ST6/03136 (ŁK, AS). This work was done while Michał Pilipczuk has been holding a post-doc position at Warsaw Centre of Mathematics and Computer Science, and has been supported by the Foundation for Polish Science via the START stipend programme.

10th International Symposium on Parameterized and Exact Computation (IPEC 2015).
Editors: Thore Husfeldt and Iyad Kanj; pp. 199–211
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Dorn et al. [4] designed subexponential parameterized algorithms with running time $2^{\tilde{O}(\sqrt{k})} \cdot n^{O(1)}$ for both problems on H-minor-free graphs[1]. They did it, however, by circumventing in both cases the need of obtaining a linear kernel. In the case of LOB they show how to apply preprocessing rules to obtain an instance that can be still large in terms of k, but has treewidth $O(\sqrt{k})$ so that the dynamic programming on a tree decomposition can be applied. In the case of IOB they apply a variant of Baker's layering technique.

Our results and techniques. In this work we fill the gap left by Dorn et al. [4] and prove that both LOB and IOB admit linear kernels on H-minor-free graphs. In fact, for IOB our approach works even in the more general setting of graph classes of bounded expansion (see Section 2 for a definition). By slightly abusing notation, in what follows we say that a directed graph D belongs to some class of undirected graphs (e.g. is H-minor free) if the underlying undirected graph of D has this property.

▶ **Theorem 1.** *Let H be a fixed graph. There is an algorithm that, given an instance (D, k) of* LOB *where D is H-minor-free, in polynomial time either resolves the instance (D, k), or outputs an equivalent instance (D', k') of* LOB *where $|V(D')| = O(k)$, $k' \leq k$, and D' is H-minor free. The algorithm does not need to know H.*

Note that Theorem 1 implies also a kernel of linear size for any minor-closed family of graphs $\mathcal{G}$. Indeed, by the Roberson and Seymour's graph minor theorem there exists a fixed finite family $\mathcal{H}$ such that $\mathcal{G}$ contains exactly graphs that are H-minor free for every $H \in \mathcal{H}$. By Theorem 1, for any input graph $D \in \mathcal{G}$, the output graph D' is H-minor free for every $H \in \mathcal{H}$. Hence, D' is in $\mathcal{G}$. In particular, it follows that Theorem 1 implies linear kernels for planar graphs and other graphs embeddable on a surface of bounded genus.

▶ **Theorem 2.** *Let $\mathcal{G}$ be a hereditary graph class of bounded expansion. There is an algorithm that, given an instance (D, k) of* IOB *where $D \in \mathcal{G}$, in polynomial time either resolves the instance (D, k), or outputs an equivalent instance (D', k) of* IOB *where $|V(D')| = O(k)$ and D' is an induced subgraph of D.*

By applying these kernelization algorithms and then running dynamic programming on a tree decomposition of the obtained graph, we easily obtain the following corollary.

▶ **Theorem 3.** *Let H be a fixed graph. Then both* LOB *and* IOB *can be solved in time $2^{O(\sqrt{k})} + n^{O(1)}$ when the input is an n-vertex H-minor-free graph.*

Algorithms with a similar running time – but with additional $\log k$ factor in the exponent – were obtained by Dorn et al. [4]. If one follows their approach, then for LOB it is possible to shave off this factor in the exponent just by replacing the dynamic programming on a tree decomposition with a more modern one. However, for IOB the logarithmic factor is caused also by an application of the layering technique, and hence such a replacement and manipulation of parameters in layering would only improve $\log k$ to $\sqrt{\log k}$. By constructing a truly linear kernel we are able to shave this factor completely off. We remark that the running time given by Theorem 3 is optimal under the Exponential Time Hypothesis even on planar graphs; see appendix for further details.

[1] We remark that Dorn et al. state the result for IOB only for apex-minor-free graphs, but a combination of their approach with the contraction decomposition technique of Demaine et al. [3] immediately generalizes the result to H-minor-free graphs.

(a) A weak bipath

(b) A fat bipath (the black vertices may have outneighbors other than those depicted and some of the vertical edges may be contracted)

Figure 1 Different types of bipaths.

To prove Theorems 1 and 2, we revisit the quadratic kernels on general graphs given by Daligault and Thomassé [2] (for LOB) and by Gutin et al. [9] (for IOB). For LOB we need to modify the approach substantially, as the core reduction rule used by Daligault and Thomassé is the following: whenever there is a *cutvertex* in the graph – a vertex whose removal makes some other vertex not reachable from r – then it is safe to *shortcut* it: remove it and add an arc from every its inneighbor to every its outneighbor. Observe that an application of this rule does not preserve H-minor-freeness, so the kernel of Daligault and Thomassé [2] may start with an H-minor free graph and go outside of this class.

To circumvent this problem, we exploit the structural approach proposed by Dorn et al. [4]. While not achieving a linear kernel in the precise sense, Dorn et al. are able to simplify the structure of the instance so that it fits their purposes. The main idea is to contract cutedges instead of shortcutting cutvertices, which is a weaker operation that, however, preserves H-minor-freeness. Dorn et al. are able to expose a set of so-called *special* vertices S of size linear in k such that $G \setminus S$ has constant pathwidth; this is already enough to employ the bidimensionality technique. To obtain a linear kernel, we need to perform a much more refined analysis of the instance. More precisely, we construct a set S with $|S| = O(k)$ such that $G \setminus S$ is consists of *fat bipaths*: chains as depicted in Figure 1, possibly with some vertical (cut)edges contracted, and with outgoing edges with heads in S. After contracting the vertical edges, such a fat bipath becomes a weak bipath: a bidirectional path possibly with outgoing edges with heads in S. Weak bipaths are crucial in the structural approach of Daligault and Thomassé [2], and our fat bipaths can be thought of as more fuzzy variants of weak bipaths that cannot be reduced due to the inability to shortcut cutvertices.

To obtain a linear kernel, we need to reduce the total length of the fat bipaths. For this, we use concepts borrowed from the analysis of graph classes of bounded expansion, of which H-minor-free classes are special cases. Very recently Drange et al. [5] announced a linear kernel for Dominating Set on graph classes of bounded expansion, and the main tool used there is the analysis of the number of different neighborhoods that can arise in a graph G from a bounded expansion graph class $\mathcal{G}$. Essentially, there is a constant c such that for every $X \subseteq V(G)$ there are only $O(|X|)$ vertices in $V(G) \setminus X$ that neighbor more than c vertices in X, while the vertices of $V(G) \setminus X$ that neighbor at most c vertices in X can be grouped into $O(|X|)$ classes with exactly the same neighborhoods. We apply this idea to the instance at hand with the interior of every fat bipath contracted to one vertex. Thus, we infer that there are only $O(k)$ fat bipaths that neighbor more than c special vertices, and their total length can be bounded by $O(k)$ using reduction rules. On the other hand, fat bipaths with neighborhoods of size at most c are reduced within their neighborhood classes, whose number is also $O(k)$.

The same neighborhood diversity argument plays the key role also in our kernel for IOB (Theorem 2). The idea of Gutin et al. [9] is that if a solution to the instance cannot be found immediately by a simple local search, then one can expose a vertex cover U of size at most $2k$ in the graph. The vertices of $V(D) \setminus U$ are reduced using an argument involving crown decompositions in an auxiliary graph where vertices of $V(D) \setminus U$ are matched to pairs of adjacent vertices of U; this gives a quadratic dependence on k of the size of the kernel. We observe that in case D belongs to a class of bounded expansion, then there is only $O(|U|) = O(k)$ vertices of $V(D) \setminus U$ that have super-constant neighborhood size in U, while the others are grouped into $O(|U|) = O(k)$ neighborhood classes, each of which can be reduced to constant size using the same approach via crown decompositions.

For IOB we did not need any edge contractions in the reduction rules, so the kernelization procedure works on any graph class of bounded expansion. However, for LOB it seems necessary to apply contractions of subgraphs of unbounded diameter, e.g. to reduce long paths that contribute with at most one leaf to the solution. While the last phase relies mostly on the bounded expansion properties of the graph class, we need to allow contractions in the reduction rules and hence we do not achieve the same level of generality as for IOB.

We see the additional advantage of our approach in its simplicity. Instead of relying on complicated decomposition theorems for H-minor free graphs, which is a standard technique in such a setting, we use the methodology proposed by Drange et al. [5]: To exploit purely combinatorial, abstract notions of sparsity, like the concept of bounded expansion, and in this manner obtain a much cleaner treatment of the considered graph classes. Of particular interest is the usefulness of the approach of grouping vertices according to their neighborhoods in some fixed modulator X, which is the key idea in [5].

Organization of the paper. In Section 2 we give preliminaries on tools borrowed from the analysis of graph classes of bounded expansion. Sections 3 and 4 are devoted to the proofs of Theorems 1 and 2, respectively. Due to space limitations, proofs of claims marked by ★ are skipped in this extended abstract; the reader can find their proofs in our technical report at arxiv [1]. By the same reason, the proof of Theorem 3 and a discussion on the optimality of the obtained algorithms is also skipped.

Notation. In this paper we deal with digraphs. Let $D = (V, E)$ be a digraph. Consider an edge $(u, v) \in E$. We say that v is an *out-neighbor* of u and u is an *in-neighbor* of v. We also say that v is a *head* and u is a *tail* of (u, v). Also, v and u are *neighbors* of each other. For any vertex v we denote the sets of all its neighbors, out-neighbors and in-neighbors by $N_D(v)$, $N_D^+(v)$ and $N_D^-(v)$, respectively. Moreover, the *degree*, *out-degree*, and *in-degree* of v are defined as $\deg_D(v) = |N(v)|$, $\deg_D^+(v) = |N^+(v)|$, and $\deg_D^-(v) = |N^-(v)|$. We omit the subscripts and write simply $N(v)$ or $\deg(v)$ whenever it does not lead to ambiguity. For any set $S \subseteq V$ we denote $N_D^-(S) = \bigcup_{v \in S} N_D^-(v) \setminus S$ and $N_D^+(S) = \bigcup_{v \in S} N_D^+(v) \setminus S$.

2 Preliminaries on Sparse Graphs

In this section we recall some definitions and basic properties of sparse graphs, in particular d-degenerate graphs, bounded expansion graphs and H-minor-free graphs. Although in this section we refer to undirected graphs, all the notions and claims apply also to digraphs, by looking at the underlying undirected graph.

We say that graph G is k-degenerate when every subgraph of G has a vertex of degree at most k. This implies (and in fact is equivalent to) that we can remove all the edges of G by repeatedly removing vertices of degree at most k. It follows that G has at most $k|V(G)|$ edges. The *degeneracy* of a graph is the smallest value of k for which it is k-degenerate. Degeneracy is closely linked to *arboricity*, i.e., minimum number $\operatorname{arb}(G)$ of forests that cover the edges of G: it is well known that degeneracy is between $\operatorname{arb}(G)$ and $2\operatorname{arb}(G)$.

Recall that a graph H is a *minor* of graph G if there exists a *minor model* $(I_u)_{u\in V(H)}$ of H in G that satisfies the following properties:

- sets I_u for $u \in V(H)$ are pairwise disjoint subsets of $V(G)$ that moreover induce connected subgraphs;
- for each $uv \in E(H)$, there exist $x_u \in I_u$ and $x_v \in I_v$ such that $x_u x_v \in E(G)$.

For any fixed graph H, the class of *H-minor-free graphs* comprises all the graphs G that do not have H as a minor. Note that H-minor free graphs are closed under minor operations: vertex and edge deletions, and edge contractions. For example, graphs embeddable into a constant genus surface are H-minor-free for some fixed H; in particular, by Kuratowski's theorem, planar graphs are K_5-minor free and $K_{3,3}$-minor free. The following lemma provides a connection between H-minor-free graphs and degeneracy.

▶ **Lemma 4** (see Lemma 4.1 in [10]). *Any H-minor free graph is d_H-degenerate for $d_H = O(|H|\sqrt{\log |H|})$.*

Let r be a nonnegative integer. If a minor model $(I_u)_{u\in V(H)}$ satisfies in addition that $G[I_u]$ has radius at most r for each $u \in V(H)$, then $(I_u)_{u\in V(H)}$ is an *r-shallow minor model of H*, and we say that H is an *r-shallow minor of G*. If $\mathcal{G}$ is a class of graphs, then by $\mathcal{G} \triangledown r$ we denote the class of all r-shallow minors of graphs from $\mathcal{G}$; note that $\mathcal{G} \triangledown 0$ are all subgraphs of graphs of $\mathcal{G}$. We now define the *greatest reduced average degree* (*grad*) of a class $\mathcal{G}$ at depth r as

$$\nabla_r(\mathcal{G}) = \sup_{H\in\mathcal{G}\,\triangledown\, r} \frac{|E(H)|}{|V(H)|}.$$

That is, we take the greatest edge density among the r-shallow minors of $\mathcal{G}$. Class $\mathcal{G}$ is said to be of *bounded expansion* if $\nabla_r(\mathcal{G})$ is a finite constant for every r. Observe that then the graphs from $\mathcal{G}$ are in particular d-degenerate for $d = \lfloor 2\nabla_0(\mathcal{G})\rfloor$. For a single graph G, we denote $\nabla_r(G) = \nabla_r(\{G\})$.

Consider the class $\mathcal{G}_H$ of H-minor-free graphs. By Lemma 4, every graph $G \in \mathcal{G}_H$ has at most $d_H \cdot |V(G)|$ edges. Since $\mathcal{G}_H$ is closed under taking minors, it follows that $\mathcal{G}_H \triangledown r = \mathcal{G}_H$ for every nonnegative r, so also $\nabla_r(\mathcal{G}_H) \le d_H$. Thus, H-minor-free graphs form a class of bounded expansion with all the grads bounded independently of r.

In this paper we do not use the original definition of bounded expansion graphs, but we rather rely on the point of view of diversity of neighborhoods, which was found to be very useful in [5]. More precisely, we now use the following result from [8, Lemma 6.6]; the statement with adjusted notation is taken verbatim from [5].

▶ **Proposition 5** (Proposition 2.5 of [5]). *Let G be a graph, $X \subseteq V(G)$ be a vertex subset, and $R = V(G) \setminus X$. Then for every integer $p \ge \nabla_1(G)$ it holds that*

1. $|\{v \in R\colon |N(v) \cap X| \ge 2p\}| \le 2p \cdot |X|$, *and*
2. $|\{A \subseteq X\colon |A| < 2p$ *and* $\exists_{v\in R}\ A = N(v) \cap X\}| \le (4^p + 2p)|X|$.

Consequently, the following bound holds:

$$|\{A \subseteq X\colon \exists_{v\in R}\ A = N(v) \cap X\}| \le \left(4^{\nabla_1(G)} + 4\nabla_1(G)\right) \cdot |X|.$$

We need a strengthening of the first claim of Proposition 5.

▶ **Lemma 6.** (★) *Let $G = (X, Y, E)$ be a bipartite graph of degeneracy at most d. Then,*

$$\sum_{\substack{y\in Y\\ \deg_G(y) > 2d}} \deg_G(y) \le 2d|X|.$$

Note that Proposition 5 has the following corollary when applied to H-minor-free graphs.

▶ **Corollary 7.** *Let H be a graph. There exists $c_H = 2^{O(|H|\sqrt{\log |H|})}$ such that in any H-minor-free bipartite graph $G = (X, Y, E)$, there are at most $c_H \cdot |X|$ vertices in Y with pairwise distinct neighborhoods in X.*

3 k-Leaf Out-Branching in H-minor-free graphs

In this section we deal with rooted digraphs, i.e., digraphs with a vertex r, called *root*, of in-degree 0. In such digraphs we redefine some standard connectivity notions as follows. Let (D, r) be a rooted digraph. We say that D is *connected* when every vertex of D is reachable from r. A *cut-vertex* is any vertex $v \in V(D) \setminus \{r\}$ such that $D - r$ is not connected. The set of all cut-vertices of D is denoted by $\mathrm{cv}(D)$. We say that D is *2-connected* if D has no cut-vertex (equivalently, for every vertex $v \in V(D) \setminus \{r\}$ there are at least two paths from r to v that do not share internal vertices). Similarly, a *cut-edge* is any edge $(u, v) \in E(D)$ such that $D - (u, v)$ is not connected. We say that D is *2-edge-connected* if D has no cut-edge (equivalently, for every vertex $v \in V(D) \setminus \{r\}$ there are at least two edge-disjoint paths from r to v). Note that if (u, v) is a cut-edge then u is a cut-vertex or $u = r$.

By a *contraction* of edge (a, b) in D we mean the following operation: identify a and b into a newly introduced vertex $v_{(a,b)}$, replace a and b with $v_{(a,b)}$ in every edge of D, and remove all the loops and parallel edges created in this manner. Note that if D is H-minor-free, then it remains H-minor-free after contractions as well.

Following [2], we say that a vertex v of D is *special* if v is of in-degree at least 3 or there is an incoming simple edge, i.e., an edge (u, v) such that $(v, u) \notin E(D)$. The set of all special vertices of D is denoted by $\mathrm{sp}(D)$.

A *weak bipath* P is a sequence of vertices $u_1, \ldots, u_p$ for some $p \geq 3$, such that for each $i = 2, \ldots, p-1$, we have $N^-(u_i) = \{u_{i-1}, u_{i+1}\} \subseteq N^+(u_i)$. The *length* of P is $p - 1$. If additionally $N^+(u_i) = N^-(u_i) = \{u_{i-1}, u_{i+1}\}$ for every $i = 2, \ldots, p-1$, we say that P is *proper bipath* (or shortly a *bipath*). u_1 and u_p are called the *extremities* of P.

We say that a cut-edge (u, v) is *lonely* when there is no other cut-edge with the tail in u. We call a cut-edge *branching* is there is another cut-edge with the same tail. The graph obtained from D by contracting all lonely cut-edges is denoted by D_c and called the *contracted graph*. Consider a vertex v of D_c. Then either v was created by contracting some set of cut-edges Z in D or $v \in D$. In the prior case we define the *bag* B of v as the set of vertices incident to edges in Z. Also, for any edge $(x, y) \in Z$ the vertex x is called a *tail* of B and y is a *head* of B. In the latter case, i.e., when $v \in D$, we define the bag as $B = \{v\}$ and v is both head and tail of B. When B is a bag of v we denote $v_B = v$ and $B_v = B$. If there is exactly one head and exactly one tail of B, then they are denoted by h_B and t_B, respectively. We say that bags A and B are *linked* if there are edges both from A to B and from B to A.

Our kernelization algorithm. Let us describe our algorithm which outputs a kernel for k-Leaf Out-Branching. The algorithm exhaustively applies reduction rules. Each reduction rule is a subroutine which finds in polynomial time a certain structure in the graph and replaces it by another structure, so that the resulting instance is equivalent to the original one. More precisely, we say that a reduction rule for parameterized graph problem P is *correct* when for every instance (D, k) of P it returns an instance (D', k') such that a) (D', k') is an instance of P, b) (D, k) is a yes-instance of P iff (D', k') is a yes-instance of P, and c)

$k' \leq k$. Below we state the rules we use. The rules are applied in the given order, i.e., in each rule we assume that the earlier rules do not apply.

Rule 1. If there exists a vertex not reachable from r in D, then reduce to a trivial no-instance.

Rule 2. If there exists a cut-vertex v with exactly one incoming edge e then contract e. Similarly, if there exists a cut-vertex v with exactly one outgoing edge e then contract e.

Rule 3. Let P be a proper bipath of length 4 in D. Contract any edge of P.

Rule 4. Let x be a vertex of D. If there exists $y \in N^-(x)$ such that the removal of $N^-(x) \setminus \{y\}$ disconnects y from r, then delete the edge (y, x).

The correctness of the above reduction rules was proven in [2]. (In [2], Rule 2 is formulated in a more general way, but we restrict it so that if the input digraph was H-minor-free, then so is the resulting reduced graph.) Let us remark that Rule 4 remains true if $r \in N^-(x) \setminus \{y\}$, and in this case it triggers removal of all the incoming edges apart from the one coming from the root. Below we introduce two simple rules which will make our argument a bit easier.

Rule 5. If there are two cut-edges (x_1, y_1) and (x_2, y_2) such that $(x_1, x_2), (x_2, x_1) \in E(D)$, then contract (x_1, x_2).

Rule 6. If there is a cut-edge (u, v) such that $(v, u) \in E(D)$, then remove (v, u).

▶ **Lemma 8.** (★) *Rules 5 and 6 are correct.*

To complete the algorithm we need a final accepting rule which is applied when the resulting graph is too big. In the remainder of this section we sketch the proof that Rule 7 is correct for H-minor-free graphs for some constant $c = 2^{O(|H|\sqrt{\log |H|})}$.

Rule 7. If the graph has more than $c \cdot k$ vertices, return a trivial yes-instance (conclude that there is a rooted outbranching with at least k leaves in D).

We conclude with the following lemma.

▶ **Lemma 9.** *Let H be a graph. If the input is an H-minor-free graph, then the output of each of the rules 1– 7 is a minor of D, and hence an H-minor-free graph. Moreover, each rule can be recognized and applied in polynomial time, and the degree of the polynomial does not depend on H.*

Proof. The first claim follows from the fact that the rules modify the graph by means of deletions and contractions only. The second claim is straightforward to check. ◀

A few simple properties of the reduced graph. Let us we state simple auxiliary lemmas, which will be used in the remainder of the paper.

▶ **Lemma 10.** (★) *If reduction rules do not apply to D then every bag is of size at most two and contains at most one edge.*

▶ **Lemma 11.** (★) *If reduction rules 1-4 do not apply to D, then for arbitrary pair of bags A and B every edge from A to B has head in t_B.*

▶ **Lemma 12.** (★) *Assume no reduction rule applies to D. Let $S \subseteq V(D_c)$ be any set of vertices that contains the root r and every special vertex of D_c. Then one can find weak bipaths $P_1, P_2, \ldots, P_q$, such that:*

(i) *The sets of internal vertices of $P_1, P_2, \ldots, P_q$ form a partition of $V(D_c) \setminus S$.*

(ii) *The extremities of each P_i belong to S and are distinct.*

(iii) *The out-neighbors of the internal vertices of each P_i belong to S.*

Weak bipaths $P_1, \dots, P_q$ given by Lemma 12 are called *maximal bipaths*. Note that for every such maximal bipath $P = v_1, v_2, \dots, v_p$ and every $j = 2, \dots, p-1$, bag B_{v_j} is linked to $B_{v_{j-1}}$ and $B_{v_{j+1}}$, and to no other bag.

New lower bounds on the number of leaves. Now our goal is to establish a number of lower bounds on the number of leaves. Each of the lower bounds is a linear function of a number of some type of vertices or structures in D. These bounds will help us prove that Rule 7 is correct. Indeed, to this end it suffices to focus on a no-instance and prove that it has at most ck vertices. Hence, if we know that $\mathrm{maxleaf}(D)$ is large when there are many vertices of some kind A, then we know that in our no-instance there are few vertices of kind A. In other words vertices of type A are "easy". In the final part of this section we will show that because of sparsity arguments the number of the remaining vertices (not corresponding to an "easy type") is linear in the number of "easy" vertices. In fact, instead of looking for "easy" vertices in D, we focus on D_c. This is justified by the fact that by Lemma 10 we have $|V(D)| \leq 2|V(D_c)|$, so if we prove that $|V(D_c)| = O(k)$ then also $|V(D)| = O(k)$.

Daligault and Thomassé [2] show the following lower bound.

▶ **Theorem 13** ([2]). *Let D be a 2-connected rooted digraph. Then* $\mathrm{maxleaf}(D) \geq \frac{|\mathrm{sp}(D)|}{30}$.

Unfortunately, D_c is not necessarily 2-connected so we cannot use the above bound. However, we can generalize Theorem 13 as follows.

▶ **Theorem 14.** (★) *Let D be a connected rooted digraph such that every cut-edge is branching. Then* $\mathrm{maxleaf}(D) \geq \frac{|\mathrm{sp}(D)|}{30} - \mathrm{cv}(D)$ *and* $\mathrm{maxleaf}(D) \geq \frac{|\mathrm{sp}(D)|}{60}$.

We are able to show that in D_c every cut-edge is branching and $\mathrm{maxleaf}(D) \geq \mathrm{maxleaf}(D_c)$ (proofs skipped in this extended abstract). This implies the following.

▶ **Lemma 15.** (★) *Assume that rules 1-6 do not apply to D. Then,* $\mathrm{maxleaf}(D) \geq \frac{|\mathrm{sp}(D_c)|}{60}$.

We say that a bag B is *special* when v_B is special in D_c. We say that a bag B is *isolated* when B is a non-special bag of size 2 and there is no edge from t_B to a special bag. Vertex $v \in V(D_c)$ is *isolated* if $v = v_B$ for some isolated bag B. The set of all isolated vertices in D_c is denoted by $\mathrm{iso}(D_c)$.

▶ **Lemma 16.** (★) *If reduction rules do not apply to D then* $\mathrm{maxleaf}(D) \geq \frac{|\mathrm{iso}(D_c)|}{180}$.

We will say that a vertex v of D_c is *easy* when $v = r$, or v is special, or v is isolated in D_c. A vertex that is not easy is called *hard*. We now invoke Lemma 12 for S being the set of all the easy vertices. Every maximal bipath obtained in this decomposition will be called a *maximal hard bipath*. In other words, a weak bipath in D_c is *hard* if all its internal vertices are hard. The sets of all easy and hard vertices in D_c are denoted by $\mathrm{ea}(D_c)$ and $\mathrm{hd}(D_c)$, respectively. For any maximal hard weak bipath P' in D_c define $O(P') = N^+_{D_c}(V(P') \setminus \{u, v\})$, where u and v are the extremities of P'.

For every pair of easy vertices $u, v \in \mathrm{ea}(D_c)$ and a subset $S \subseteq V(D_c)$ with $\{u, v\} \subseteq S$, if there is a hard bipath P' between u and v such that $O(P') = S$, we choose arbitrarily two such paths (or one, if only one exists) and we call them *masters*, while all the remaining hard bipaths P'' between u and v with $O(P'') = S$ are called *slaves* of respective masters, or just *slaves*. The number of all slaves in D_c is denoted by $\mathrm{sl}(D_c)$.

▶ **Lemma 17.** (★) $\mathrm{maxleaf}(D) \geq \mathrm{sl}(D_c)$.

The size bound. The following theorem implies the correctness of Rule 7.

▶ **Theorem 18.** *Let H be a graph. Let D be an H-minor-free digraph such that rules 1–6 do not apply. If* $\mathrm{maxleaf}(D) < k$, *then* $|V(D)| = 2^{O(|H|\sqrt{\log|H|})}k$.

In what follows we prove Theorem 18. We assume that rules 1–6 do not apply to D. Since $\mathrm{maxleaf}(D) < k$, our lower bounds on $\mathrm{maxleaf}(D)$ imply an upper bound of $O(k)$ on the number of easy vertices. Our plan now is to show a linear bound on the number of hard vertices in terms of $|\mathrm{ea}(D_c)| + \mathrm{sl}(D_c)$ and next get a bound on $|V(D)|$ as a corollary. To this end, we state a few useful properties of hard weak bipaths in D_c.

▶ **Lemma 19.** *Let $\ell \geq 9$ and let $P' = v_1, \ldots, v_\ell$ be a hard bipath in D_c such that v_1 and v_ℓ are easy. For every $i = 3, \ldots, \ell - 6$ there is at least one edge in D from $t_{B_{v_j}}$, for some $j = i, \ldots, i+4$, to a vertex outside $\cup_{j'=2}^{\ell-1} B_{v_{j'}}$.*

Proof. Fix $i \in \{3, \ldots, \ell-6\}$ and consider the length 4 bipath $v_i, \ldots, v_{i+4}$. Denote $B_j = B_{v_j}$. If for some $j = i+1, i+2, i+3$ there is an edge from B_j with head $h \notin B_{j-1} \cup B_{j+1}$, then by Lemma 12(*iii*), $h \notin \cup_{j'=2}^{\ell-1} B_{v_{j'}}$ and we are done. Hence the edges leaving B_{i+1}, B_{i+2}, and B_{i+3} go only to the neighboring bags. Since Rule 3 does not apply, for some $j = i, \ldots, i+4$ the bag B_j is of size 2. Since v_j is hard, B_j is not isolated. Hence, there is an edge e in D from t_{B_j} to a special bag B. Since $v_2, \ldots, v_{\ell-1}$ are hard, B is none of $B_2, \ldots, B_{\ell-1}$. ◀

▶ **Lemma 20.** *For any maximal hard weak bipath P' in D_c,* $|\mathrm{hd}(D_c) \cap V(P')| \leq 10|O(P')| + 6$.

Proof. Let $P' = v_1, \ldots, v_\ell$. We can assume that $\ell \geq 9$, for otherwise $|\mathrm{hd}(D_c) \cap V(P')| \leq 6$ and the claim holds trivially. For convenience denote $B_i = B_{v_i}$. By Lemma 19 there are at least $\lfloor \frac{\ell-4}{5} \rfloor$ edges from tails of bags $B_3, \ldots, B_{\ell-2}$ to vertices outside $\cup_{i=2}^{\ell-1} B_{v_j}$. Let Z denote the set of these edges. We claim that for every vertex $u \in V(D)$ there are at most two edges from Z with heads in u. Indeed, assume that u has got three in-neighbors $t_{B_a}, t_{B_b}, t_{B_c}$ in D, with $a < b < c$. Then $N^-(u) \setminus \{t_{B_b}\}$ cuts t_{B_b} (and all vertices of $B_{a+1}, \ldots, B_{c-1}$) from r, a contradiction to the fact that D is reduced with respect to Rule 4. Hence the edges in Z have at least $\lfloor \frac{\ell-4}{5} \rfloor \cdot \frac{1}{2} \geq \frac{\ell-8}{5} \cdot \frac{1}{2}$ different heads. By Lemma 11 these heads are tails of bags, and by Lemma 10 each of them corresponds to a different vertex in D_c. It follows that the vertices $v_3, \ldots, v_{\ell-2}$ have in D_c at least $\frac{\ell-8}{10}$ neighbors in $O(P')$, so $|O(P')| \geq \frac{\ell-8}{10}$. Since $|\mathrm{hd}(D_c) \cap V(P)| = \ell - 2$ it follows that $|\mathrm{hd}(D_c) \cap V(P)| \leq 10|O(P')| + 6$. ◀

In what follows we are going to bound the size of D_c using its sparsity properties. To this end we use an auxiliary bipartite graph G, called the *bipath minor* of D_c, constructed as follows. We put $V(G) = A \cup B$, where $A = \mathrm{ea}(D_c)$, and B is the set of all maximal hard bipaths in D_c. For every maximal hard bipath P' in D_c with extremities $u, v \in \mathrm{ea}(D_c)$, the neighborhood of the corresponding vertex in B is exactly $O(P)$.

▶ **Lemma 21.** *If D is H-minor-free, then* $|\mathrm{hd}(D_c)| = 2^{O(|H|\sqrt{\log|H|})}(|\mathrm{ea}(D_c)| + \mathrm{sl}(D_c))$.

Proof. Consider an arbitrary hard vertex v of D_c. Consider the maximal hard weak bipath P' in D_c that contains v. Then P' corresponds to a vertex in B and by Lemma 20, it has at most $10|O(P')| + 6$ internal vertices. It follows that

$$|\mathrm{hd}(D_c)| \leq \sum_{v \in B} (10 \deg_G(v) + 6) \leq \sum_{v \in B} 16 \deg_G(v). \tag{1}$$

Note that G is a minor of (the undirected version of) D since it can be obtained from D_c by edge contractions and deletions, and D_c in turn is obtained from D by contractions.

Hence, G is H-minor-free. Moreover, G is simple. By Lemma 4, we know that G is d_H-degenerate, for $d_H = O(|H|\sqrt{\log |H|})$. Let B_m and B_s denote the vertices in B for which the corresponding maximal hard bipath is master and slave, respectively. By (1) we get

$$|\mathrm{hd}(D_c)| \le 16 \sum_{v \in B} \deg_G(v) \le 16 \sum_{\substack{v \in B \\ \deg_G(v) > 2d_H}} \deg_G(v) + 16 \sum_{\substack{v \in B_s \\ \deg_G(v) \le 2d_H}} \deg_G(v) + 16 \sum_{\substack{v \in B_m \\ \deg_G(v) \le 2d_H}} \deg_G(v)$$

Let us bound each of the terms separately. By Lemma 6, we have

$$\sum_{\substack{v \in B_s \\ \deg_G(v) > 2d_H}} d(v) \le 2d_H|A| = O(|H|\sqrt{\log |H|} \cdot |\mathrm{ea}(D_c)|).$$

Obviously,

$$\sum_{\substack{v \in B_s \\ \deg_G(v) \le 2d_H}} \deg_G(v) \le 2d_H \mathrm{sl}(D_c) = O(|H|\sqrt{\log |H|}\mathrm{sl}(D_c)).$$

Finally,

$$\sum_{\substack{v \in B_m \\ \deg_G(v) \le 2d_H}} \deg_G(v) = \sum_{\substack{S \subseteq A \\ |S| \le 2d_H}} |S| \cdot |\{v \in B_m \colon N_G(v) = S\}| \le 2d_H \sum_{\substack{S \subseteq A \\ |S| \le 2d_H}} |\{v \in B_m \colon N_G(v) = S\}|.$$

By Corollary 7, there is a constant $c_H = 2^{O(|H|\sqrt{\log |H|})}$ such that there are at most $c_H|A|$ distinct neighborhoods of vertices in B. For each such neighborhood $S \subseteq A$ and for every pair of vertices $u, v \in S$ there are at most two master bipaths P' with endpoints u and v and such that $O(P') = S$. Therefore for a fixed neighborhood S of size at most $2d_H$ we have $|\{v \in B_m | N_G(v) = S\}| \le 2\binom{|S|}{2} \le 2\binom{2d_H}{2} = O(d_H^2)$. Hence

$$\sum_{S \subseteq A, |S| \le 2d_H} |\{v \in B_m | N_G(v) = S\}| = O(c_H \cdot d_H^2 \cdot |\mathrm{ea}(D_c)|) = 2^{O(|H|\sqrt{\log |H|})}|\mathrm{ea}(D_c)|.$$

The claim follows. ◀

Now we can finish the proof of Theorem 18. Assume $\mathrm{maxleaf}(D) < k$. By Lemmas 15 and 16, $\mathrm{ea}(D_c) < 60k + 180k$. Moreover, by Lemma 17, $\mathrm{sl}(D_c) < k$. This, with Lemma 21 gives the claim of Theorem 18.

4 k-internal Out-Branching in graphs of bounded expansion

In this section we give a linear kernel for IOB on any graph class $\mathcal{G}$ of bounded expansion. To this end, we modify the approach of Gutin, Razgon and Kim [9]. Before we proceed to the argumentation, let us remark that Gutin et al. work with a slightly more general problem, where the root of the outbranching is not prescribed; of course, the outbranching is still required to span the whole vertex set. Note that the variant with a prescribed root r can be reduced to this variant simply by removing all in-arcs of r, which forces r to be the root of any outbranching of the given digraph. Since our kernel will be an induced subgraph of D and r will not be removed by any reduction, it will be still true that r is the only candidate for the root of an outbranching. Hence, the resulting instance will be equivalent in both variants. Therefore, from now on we work with variant without prescribed root in order to be able to use the observations of Gutin et al. as black-boxes.

First, Gutin et al. observe that in an instance that cannot be easily resolved, one can find a small vertex cover (of the underlying undirected graph).

▶ **Lemma 22** ([9]). *Given a digraph D, we can either build an out-branching with at least k internal vertices or obtain a vertex cover of size at most $2k-2$ in $O(n^2m)$ time.*

For a given directed graph D and a vertex cover U in D we build an undirected bipartite graph $B_{D,U}$ as follows. Let $W = V(D) \setminus U$. Then,

$$\begin{aligned} V(B) &= U' \cup W, \text{ where } U' = N^-(W) \cup (U \times U); \\ E(B) &= \{\{xy, w\} \ : \ xy \in U \times U, \ w \in W, (x,w) \in E(D), \ (w,y) \in E(D)\} \cup \\ &\quad\ \{\{x, w\} \ : \ x \in U, \ w \in W, \ (x,w) \in E(D)\}. \end{aligned}$$

A *crown decomposition* of an undirected graph G is a partitioning of $V(G)$ into three parts C, H and R, such that

- C is an independent set.
- There are no edges between vertices of C and R. That is, H separates C and R.
- C can be partitioned into $C_m \cup C_u$ with $|C_m| = |H|$, such that $G[C_m \cup H]$ contains a perfect matching that matches each vertex of C_m with a vertex of H.

Crown decompositions are used in multiple kernelization algorithms. In particular, the following lemma, which Gutin et al. attribute to Fellows et al. [6], shows that in certain situations a crown decomposition can be found efficiently.

▶ **Lemma 23** (see [9]). *Suppose G is an undirected graph on n vertices, and suppose I is an independent set in G such that $|I| \geq \frac{2n}{3}$. Then G admits a crown decomposition $(C = C_u \uplus C_m, H, R)$ with $C \subseteq I$, $H \subseteq V(G) \setminus I$ and $C_u \neq \emptyset$. Moreover, given I, the decomposition $(C = C_u \uplus C_m, H, R)$ can be found in $O(nm)$ time.*

The main idea of Gutin et al. is to search for crowns in $B_{D,U}$ with $C \subseteq W$ and $C_u \neq \emptyset$. Such crowns can be conveniently reduced using the following reduction rule, whose correctness is proved in Lemma 4.4 of [9].

Rule 1. Let U be a vertex cover in D and let $W = V(D) \setminus U$. Assume there is a crown decomposition $(C = C_m \cup C_u, H, R)$ in $B_{D,U}$ with $C \subseteq W$ and $C_u \neq \emptyset$. Then remove C_u from D.

Our idea is to combine Rule 1 with the knowledge that D belongs to a graph class of bounded expansion $\mathcal{G}$, and hence Proposition 5 can be used to reason about the sparseness of the adjacency structure between U and W. Let us introduce some notation. Consider a vertex cover U and an independent set $W = V(D) \setminus U$ in D. Let $W_s = \{w \in W \ : \ \deg_D(w) < 2\nabla_0(\mathcal{G})\}$, and let $W_b = W \setminus W_s$. Moreover, for $N \subseteq U$ with $|N| < 2\nabla_0(\mathcal{G})$, let $W_N = \{w \in W_s \ : \ N(w) = N\}$. Let $\mathcal{N}(U) = \{N \subseteq U \ : \ |N| < 2\nabla_0(\mathcal{G}), W_N \neq \emptyset\}$. Note that $|\mathcal{N}(U)| \leq |W_s|$. Our kernelization algorithm is as follows.

1. If the algorithm from Lemma 22 returns an outbranching, answer YES and terminate; otherwise it returns a vertex cover U of size at most $2k-2$. Let $W = V(D) \setminus U$.
2. Construct the graph $B := B_{D,U}$ and compute W_s, $\mathcal{N}(U)$, and nonempty sets W_N.
3. If there is a set $N \in \mathcal{N}(U)$ such that $|W_N| > 2|N_B(W_N)|$, then apply Lemma 23 to graph $B[N_B[W_N]]$ with $I = W_N$. This gives us a crown decomposition $(C = C_u \uplus C_m, H, R)$ of $B[N_B[W_N]]$ with $C \subseteq W_N$, $H \subseteq N_B(W_N)$, and $C_u \neq \emptyset$. Observe that $(C = C_u \uplus C_m, H, R \cup (V(B) \setminus N_B[W_N]))$ is a crown decomposition of B. Apply Rule 1 to this crown decomposition in order to remove C_u from D, and restart the algorithm in the reduced graph.
4. Otherwise, return D.

In case we have a prescribed root r of the outbranching that we would like to preserve in the kernelization process, we can add it to the constructed vertex cover U, thus increasing its size up to at most $2k-1$. The reduction rules never remove any vertex of U.

Given this algorithm, we can restate and prove our main result for IOB.

▶ **Theorem 2.** *Let $\mathcal{G}$ be a hereditary graph class of bounded expansion. There is an algorithm that, given an instance (D,k) of* IOB *where $D \in \mathcal{G}$, in polynomial time either resolves the instance (D,k), or outputs an equivalent instance (D',k) of* IOB *where $|V(D')| = O(k)$ and D' is an induced subgraph of D.*

Proof. The correctness of our kernelization algorithm and a polynomial bound on its running time follows from Lemmas 22 and 23. Note that the kernelization algorithm never decrements the budget k, so it suffices to show that it outputs an instance (D,k) such that $|V(D)| = O(k)$.

We can assume that the algorithm constructed a vertex cover U of D of size at most $2k-2$ ($2k-1$ if we want to preserve a prescribed root), because otherwise the algorithm would terminate and provide a positive answer. Let $W = V(D) \setminus U$. Then $V(D) = U \cup W_s \cup W_b$. By the first claim of Proposition 5 we get $|W_b| \leq 2\nabla_0(G)|U| \leq 4\nabla_0(\mathcal{G})k$. Hence it suffices to bound the size of W_s. Note that $W_s = \bigcup_{N\in\mathcal{N}(U)} W_N$. By the second claim of Proposition 5 we get $|\mathcal{N}(U)| \leq (4^{\nabla_1(G)} + 2\nabla_1(G))|U| = O(4^{\nabla_1(\mathcal{G})}k)$. However, since Step 2 of the kernelization algorithm cannot be applied, for every $N \in \mathcal{N}(U)$ we have $|W_N| \leq 2|N_{B_{D,U}}(W_N)|$. However, by the construction of $B_{D,U}$ it is clear that $|N_{B_{D,U}}(W_N)| \leq |N|^2 + |N| < 4\nabla_0(\mathcal{G})^2 + 2\nabla_0(\mathcal{G})$, and hence $|W_N| < 8\nabla_0(\mathcal{G})^2 + 4\nabla_0(\mathcal{G})$. It follows that $|W_s| = \sum_{N\in\mathcal{N}(U)} |W_N| = O(4^{\nabla_1(\mathcal{G})}\nabla_0(\mathcal{G})^2k)$, and hence $|V(D)| = |U|+|W_s|+|W_b| = O(4^{\nabla_1(\mathcal{G})}\nabla_0(\mathcal{G})^2k)$. This finishes the proof. ◀

Acknowledgments. The authors are very grateful to Marcin Pilipczuk for reading the manuscript carefully and providing useful comments.

References

1 Marthe Bonamy, Łukasz Kowalik, Michał Pilipczuk, and Arkadiusz Socała. Linear kernels for outbranching problems in sparse digraphs. *CoRR*, abs/1509.01675, 2015.

2 Jean Daligault and Stéphan Thomassé. On finding directed trees with many leaves. In *Parameterized and Exact Computation*, pages 86–97. Springer, 2009.

3 Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Contraction decomposition in H-minor-free graphs and algorithmic applications. In *Proc. STOC'11*, pages 441–450. ACM, 2011.

4 Frederic Dorn, Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Beyond bidimensionality: Parameterized subexponential algorithms on directed graphs. *Inf. Comput.*, 233:60–70, 2013.

5 Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Stephan Kreutzer, Daniel Lokshtanov, Marcin Pilipczuk, Michał Pilipczuk, Felix Reidl, Saket Saurabh, Fernando Sánchez Villaamil, and Somnath Sikdar. Kernelization and sparseness: the case of dominating set. *CoRR*, abs/1411.4575, 2014.

6 Mike Fellows, Pinar Heggernes, Frances A. Rosamond, Christian Sloper, and Jan Arne Telle. Finding k disjoint triangles in an arbitrary graph. In *WG'04*, pages 235–244, 2004.

7 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. Bidimensionality and kernels. In *Proc. SODA'10*, pages 503–510, 2010.

8 Jakub Gajarský, Petr Hlinený, Jan Obdrzálek, Sebastian Ordyniak, Felix Reidl, Peter Rossmanith, Fernando Sanchez Villaamil, and Somnath Sikdar. Kernelization using structural parameters on sparse graph classes. In *ESA 2013*, pages 529–540. Springer, 2013.

9 Gregory Gutin, Igor Razgon, and Eun Jung Kim. Minimum leaf out-branching and related problems. *Theor. Comput. Sci.*, 410(45):4571–4579, 2009.

10 Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity: Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and Combinatorics*. Springer, 2012.