HAL open science

Adapting The Directed Grid Theorem into an FPT Algorithm

Victor Campos, Raul Lopes, Ana Karolinna Maia de Oliviera, Ignasi Sau

To cite this version:

Victor Campos, Raul Lopes, Ana Karolinna Maia de Oliviera, Ignasi Sau. Adapting The Directed Grid Theorem into an FPT Algorithm. Electronic Notes in Theoretical Computer Science, 2019, 346, pp.229-240. 10.1016/j.entcs.2019.08.021 . lirmm-02410620

HAL Id: lirmm-02410620

https://hal-lirmm.ccsd.cnrs.fr/lirmm-02410620v1

Submitted on 13 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 346 (2019) 229–240

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Adapting The Directed Grid Theorem into an FPT Algorithm[1,2]

Victor Campos, Raul Lopes, Ana Karolinna Maia

ParGO group, Universidade Federal do Cear
Fortaleza, Brazil

Ignasi Sau

CNRS, LIRMM, Universit de Montpellier
Montpellier, France

Abstract

Originally proved in 1986 by Robertson and Seymour, the Grid Theorem is one of the most important tools in the field of structural graph theory, finding numerous applications in the design of algorithms for undirected graphs. An analogous version of the Grid Theorem isn directed graphs was conjectured by Johnson et al. in 2001, and proved recently by Kawarabayashi and Kreutzer in 2015. Namely, they showed that there is a function $f(k)$ such that every directed graph of directed tree-width at least $f(k)$ contains a cylindrical grid of size k as a butterfly minor. Moreover, they claim that their proof can be turned into an XP algorithm, with parameter k, that either constructs a decomposition of the appropriate width, or finds the claimed large cylindrical grid as a butterfly minor. In this article, we adapt some of the steps of the proof of Kawarabayashi and Kreutzer and we improve the XP algorithm into an FPT algorithm.
The first step of the proof is an XP algorithm by Johnson et al. in 2001 that decides whether a directed graph D has directed tree-width at most $3k-2$ or admits a haven of order k. It is worth mentioning that a skecth of an FPT algorithm for this problem appears in Chapter 9 of the book "Classes of Directed Graphs", from 2018, with an approximation factor of $5k+2$. Our first contribution is to adapt the proof from Johnson et al. to find either an arboreal decomposition of width at most $3k-2$ or a haven of order k in a directed graph D in FPT time, by making use of important separators. We then follow the roadmap of the proof by Kawarabayashi and Kreutzer by locally improving the complexity at some steps, in particular concerning the problem of finding hitting sets for brambles of large order.

Keywords: Directed graphs, directed tree-width, grid theorem, FPT algorithms.

1 Introduction

Width parameters can be seen as an estimation of how close a given graph is to a typical structure. For example, the *tree-width* of a graph, a parameter of particular

[1] Emails: campos@lia.ufc.br, raul.lopes@lia.ufc.br, karolmaia@lia.ufc.br, ignasi.sau@lirmm.fr

[2] Work supported by projects CAPES/STIC-AmSud 88881.197438/2018-01, CNPq - Universal project 425297/2016-0, FUNCAP - PRONEM PNE-011200061.01.00/16, DEMOGRAPH (ANR-16-CE40-0028), and ESIGMA (ANR-17-CE23-0010).

https://doi.org/10.1016/j.entcs.2019.08.021
1571-0661/© 2019 Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

interest in the literature, measures how tightly the graph can be approximated by a tree. A *tree decomposition* of a graph with bounded tree-width shows how one can place the vertices of the original graph into "bags" of bounded size which, in turn, can be arranged as the vertices of some tree in a way that the intersections pairs of adjacent bags in the tree are separators in the original graph. Thus, a tree decomposition exposes a form of global connectivity measure for graphs: as only a bounded number of vertices can be placed in each bag, many small separators of the graph can be identified through the decomposition.

A number of hard problems can be efficiently solved in graphs of bounded tree-width, either by making use of classical algorithmic techniques like dynamic programming, or by making use of Courcelle's Theorem [10]. Applications of algorithms based on tree decompositions range from frequency allocation problems to the traveling salesman problem [9, 20]. The tree-width of graphs was first introduced in [2], then again in [16] and finally reintroduced in [26]. For a survey on the subject, we refer the reader to [3].

Given the enormous success achieved by applications based on width parameters on undirected graphs, it is no surprise that there is interest into finding analogous definitions for directed graphs. An analogous measure for tree-width in the directed case is given in [17]. The *directed tree-width* of a directed graph measures its distance to being a DAG, and an *arboreal decomposition* of a directed graph exposes a (strong) connectivity measure of the original graph. An intuition for the similarities between the undirected and directed cases can be found in [25].

It is natural to ask what can be said of a graph with large tree-width. One of the most relevant results in structural graph theory states that graphs with large tree-width contain large grid minors. More precisely, the Grid Theorem [26] states that there is a function $f : \mathbb{N} \to \mathbb{N}$ such that every graph of tree-width at least $f(k)$ contains a $k \times k$-grid as a minor. A polynomial function has been recently given in [7].

It is worth mentioning that, sometimes, large tree-width (and therefore, the existence of a large grid minor) implies that we are actually working with a positive instance of a particular problem. One such case is the LONGEST PATH problem. In [12], it is given a framework that generates fixed-parameter algorithms for many such problems, known as *bidimensional* problems. This list includes VERTEX COVER, FEEDBACK VERTEX SET, MINIMUM MAXIMAL MATCHING, DOMINATING SET, EDGE DOMINATING SET, and many others. This seminal work is known as *Bidimensionality*.

As another example of application of the Grid Theorem, consider the k-DISJOINT PATHS problem. It is solved in polynomial time on general graphs through the following approach [27]. If the input graph has bounded tree-width, the problem is solved directly through standard dynamic programming techniques. If not, then by the Grid Theorem, the graph contains a large grid. Now, one can show that if a solution to the instance uses a vertex which is "very deep" into the grid, this solution can be re-routed to avoid such vertex. Until the graph satisfies a set of properties (which, in this case, includes having small tree-width), the existence of

an *irrelevant vertex*, that is, a vertex that when removed does not change the answer to the problem, can be verified. This leads to an iterative algorithm, reducing the problem to a smaller instance, until it satisfies conditions for the tractability of the problem. This irrelevant vertex technique was widely used in a number of different problems (cf. [15, 19], for example).

For the directed case, an analogous result to the Grid Theorem was conjectured by Johnson et al. [17] and recently proved by Kawarabayashi and Kreutzer [18]. Namely, it is shown in [18] that there is a function $f : \mathbb{N} \to \mathbb{N}$ such that every directed graph of directed tree-width at least $f(k)$ contains a *cylindrical grid* of order k as a *butterfly minor* [3]. All the definitions of the objects mentioned in the introduction can be found in Section 2.

The proof of the Directed Grid Theorem given in [18] is constructive. They start from a result of [17] that, in XP time, given a directed graph D and an integer parameter k, outputs either an arboreal decomposition of D of width at most $3k-2$ or a haven $\mathcal{H}$ of order k. From here, in [18], they obtain a bramble $\mathcal{B}$ of order $\lfloor k/2 \rfloor$from $\mathcal{H}$. Using $\mathcal{B}$, they give a constructive proof that there is a path P containing a well-linked set A of size roughly $\sqrt{k/2}$. In particular, the construction of P cannot easily be adapted into an XP algorithm as an exhaustive description of the bramble is used and there is no guarantee that a bramble of order k has size bounded by a function of k. From P and A, the remainder of the proof of the Directed Grid Theorem [18] runs in FPT time, with parameter k.

The main contribution of this paper is an FPT algorithm that, given D and k, either produces an arboreal decomposition of width at most $3k-2$ or outputs a path P containing a well-linked set A of size precisely $\sqrt{k}$. Our algorithm strongly follows the ideas of [17, 18]. In Theorem 2.9, we give an FPT algorithm that, given a directed graph D and parameter k, outputs either an arboreal decomposition of D of width at most $3k-2$ or a haven $\mathcal{H}$ of order k. We acknowledge that a similar result is mentioned in [1, Theorem 9.4.4], but with an approximation factor of $5k+10$ and containing only an incomplete sketch of the proof. In Section 4, we show how to obtain a compact definition of a bramble $\mathcal{B}$ of order k from $\mathcal{H}$. This compact definition is used to avoid an enumeration of the bramble elements. In Lemma 4.9 we use $\mathcal{B}$ to find a path P containing a well-linked set A of order $\sqrt{k}$ in FPT time. We make use of the FPT algorithm given in [14], that solves a variation of the Multicut problem for directed graphs. A roadmap of the aforementioned algorithm is given in Figure 1. We mark by a dashed arc the steps of [18] which are already FPT and do not need to be adapted.

All others arcs represent steps that we adapt in this paper.

2 Formal definitions and preliminaries

In this section we give the relevant definitions for this paper and mention some known results and relations about the objects defined below. The contents of this section are mostly based on [17].

[3] The full version of [18] is available at `https://arxiv.org/abs/1411.5681`.

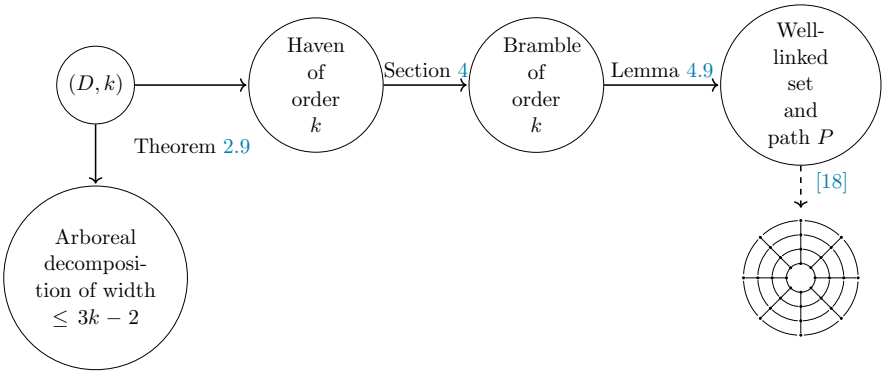

Fig. 1. Sketch of the algorithm used in the proof of the Directed Grid Theorem [18].

2.1 Parameterized complexity

We refer the reader to [11,13] for basic background on parameterized complexity, and we recall here only some basic definitions. A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$. For an instance $I = (x, k) \in \Sigma^* \times \mathbb{N}$, k is called the *parameter*.

A parameterized problem is *fixed-parameter tractable* (FPT) if there exists an algorithm $\mathcal{A}$, a computable function f, and a constant c such that given an instance $I = (x, k)$, $\mathcal{A}$ (called an FPT *algorithm*) correctly decides whether $I \in L$ in time bounded by $f(k) \cdot |I|^c$. For instance, the Vertex Cover problem parameterized by the size of the solution is FPT.

A parameterized problem is in XP if there exists an algorithm $\mathcal{A}$ and two computable functions f and g such that given an instance $I = (x, k)$, $\mathcal{A}$ (called an XP *algorithm*) correctly decides whether $I \in L$ in time bounded by $f(k) \cdot |I|^{g(k)}$. For instance, the Clique problem parameterized by the size of the solution is in XP.

Within parameterized problems, the class W[1] may be seen as the parameterized equivalent to the class NP of classical decision problems. Without entering into details (see [11,13] for the formal definitions), a parameterized problem being W[1]-*hard* can be seen as a strong evidence that this problem is *not* FPT. The canonical example of W[1]-hard problem is Clique parameterized by the size of the solution.

2.2 Arboreal decompositions and obstructions

We refer the reader to [4] for basic background on graph theory, and recall here only some basic definitions. All paths mentioned below, unless stated otherwise, are considered to be directed. For a graph $G = (V, E)$, directed or not, and a set $X \subseteq V(G)$, we write $G \setminus X$ for the graph resulting from the deletion of X from G.

By an *arborescence* R, we mean an orientation of a tree with root r in such a way that all edges are pointing away from r. If a vertex v of R has out-degree zero, we say that v is a *leaf* of R. We now define guarded sets and arboreal decompositions of directed graphs. From here on, we refer to oriented edges only, unless stated otherwise. D will always stand for a directed graph, and G for an undirected graph.

Definition 2.1 [Z-guarded sets] Let Z be a subset of $V(D)$ and $S \subseteq V(D) \setminus Z$.

We say that S is *Z-guarded* if there is no directed walk in $D \setminus Z$ with first and last vertices in S that uses a vertex of $D \setminus (Z \cup S)$.

If a set S is Z-guarded, we may also say that Z is a *guard* for S. We remark that in [17], the authors use the terminology *Z-normal* sets instead of Z-guarded sets.

Let R be an arborescence, $r \in V(R)$, $e \in E(R)$ and r' be the head of e. We say that $r > e$ if $r = r'$ or $r > r'$. We also say that $e \sim r$ if r is the head or the tail of e. The tree-width of directed graphs is defined as follows.

Definition 2.2 [Arboreal decomposition and directed tree-width] An *arboreal decomposition* of a directed graph D is a triple $(R, \mathcal{X}, \mathcal{W})$ where R is an arborescence, $\mathcal{X} = \{X_e : e \in E(R)\}$, $\mathcal{W} = \{W_r : r \in V(R)\}$, and $\mathcal{X}, \mathcal{W}$ are collections of sets of vertices of D (called *bags*) such that

(D1) $\mathcal{W}$ is a partition of $V(D)$ into nonempty sets; and

(D2) if $e \in E(R)$, then $\bigcup\{W_r : r \in V(R), r > e\}$ is X_e-guarded.

For a vertex $r \in V(R)$, we refer to the *width* of r as $|W_r \cup (\bigcup_{e \sim r} X_e)|$. The *width* of $(R, \mathcal{X}, \mathcal{W})$ is the least integer k such that, for all $r \in V(R)$, $\mathrm{width}(r) \leq k + 1$. The *directed tree-width* of D, denoted by $\mathsf{dtw}(D)$, is the least integer k such that D has an arboreal decomposition of width k.

Definition 2.3 [Nice arboreal decompositions.] We say that an arboreal decomposition $(R, \mathcal{X}, \mathcal{W})$ of a directed graph D is *nice* if

(i) for every $e \in E(R)$ with head r, W_r induces a strong component of $D \setminus X_e$; and

(ii) if $r \in V(R)$ and $r_1, \ldots, r_\ell$ are the out-neighbors of r in R, then

$$\left(\bigcup_{1 \leq i \leq \ell} W_{r_i} \right) \cap \left(\bigcup_{e \sim r} X_e \right) = \emptyset.$$

We now define some blocking structures for large directed tree-width.

Definition 2.4 [Well-linked sets] Let D be a directed graph and $A \subseteq V(D)$. We say that A is *well-linked* if, for all $X, Y \subseteq A$ with $|X| = |Y| = k$, there are k vertex-disjoint paths from X to Y in D.

Notice that one can always assume that $X \cup Y = A$. We can always include in X and in Y all vertices of A there were not originally in X or in Y, as each vertex in $X \cap Y$ is a directed path from X to Y.

Definition 2.5 [Havens in directed graphs] Let D be a directed graph. A *haven of order k* in D is a function β assigning to every set $Z \subseteq V(D)$, with $|Z| \leq k - 1$, the vertex set of a strong component of $D \setminus Z$ in such way that if $Z' \subseteq Z \subseteq V(D)$ then $\beta(Z) \subseteq \beta(Z')$. The *haven number* of D is the maximum k such that D admits a haven of order k.

There is a direct correlation between the haven number and the tree-width of undirected graphs. A haven in an undirected graph is defined similarly. The function β retains all its properties, but mapping sets of at most $k-1$ vertices to components of the graph resulting from the deletion of those vertices.

Theorem 2.6 [28] *Let G be a graph and $k \geq 0$ an integer. Then G has a haven of order k if and only if its tree-width is at least $k-1$.*

For directed graphs, only one implication of the previous result is known to be true.

Theorem 2.7 [17] *Let D be a directed graph and k a non-negative integer. If D has a haven of order k, then $\mathsf{dtw}(D) \geq k-1$.*

For the reverse direction of Theorem 2.7, only an approximate version is known.

Theorem 2.8 [17] *Let D be a directed graph and k a positive integer. Then either $\mathsf{dtw}(D) \leq 3k-2$ or D admits a haven of order k.*

The proof of Theorem 2.8 given in [17] yields an XP algorithm that decides whether a directed graph D admits a haven of order k or produces an arboreal decomposition of D of width at most $3k-2$. Furthermore, although not explicitly mentioned in the article, this algorithm actually produces a nice (as in Definition 2.3) arboreal decomposition for D and can be used as a procedure that, given a directed graph D' such that $\mathsf{dtw}(D') \leq k-2$, generates a nice arboreal decomposition for D' of width at most $3k-2$. In Section 3, by making use of a variation of the Multicut [5,8] problem, we provide an FPT algorithm with the same output.

Theorem 2.9 (First main contribution) *Let D be a directed graph and k a non-negative integer. There is an FPT algorithm, with parameter k, that decides whether D admits a haven of order k or produces an arboreal decomposition of D of width at most $3k-2$.*

The proof of the Directed Grid Theorem [18] starts by deciding whether a directed graph has bounded directed tree-width. Theorem 2.9 is the first step for the proof of Corollary 2.13.

Next, we define a structure which will be useful to find highly connected subgraphs in directed graphs of large directed tree-width.

Definition 2.10 [Brambles in directed graphs] A *bramble* $\mathcal{B} = \{B_1, \ldots, B_\ell\}$ in a directed graph D is a family of strongly connected subgraphs of D such that if $\{B, B'\} \subseteq \mathcal{B}$ then $B \cap B' \neq \emptyset$ or there are edges in D from B to B' and from B' to B. A *hitting set* of a bramble $\mathcal{B}$ is a set $C \subseteq V(D)$ such that $C \cap B \neq \emptyset$ for all $B \in \mathcal{B}$. The *order* $\mathsf{ord}(\mathcal{B})$ of a bramble $\mathcal{B}$ is the minimum size of set that hits $\mathcal{B}$. The *bramble number* of a directed graph D, denoted by $\mathrm{bn}(D)$, is the the maximum k such that D admits a bramble of order k.

Brambles of high order are also key to the original proof of the Directed Grid Theorem. In Section 4, we show how to obtain, from a path P intersecting all

elements of a bramble of order $(k+1)(\lfloor k/2+1)$, a well-linked set A of size k such that $A \subseteq V(P)$. From this point forward, the proof of Corollary 2.13 follows the proof contained in [18].

We now formally define cylindrical grids and formally enunciate our second main contribution. We refer the reader to [18] for the formal definitions of butterfly contractions and butterfly minors.

Definition 2.11 [Cylindrical grid] A *cylindrical grid of order* k is a directed graph D_k consisting of k pairwise disjoint cycles $C_1, \ldots, C_k$ and set of $2k$ pairwise disjoint paths $P_1, \ldots, P_{2k}$ such that

(i) $V(P_i) \cap V(C_j)| = 1$, for $i \in \{1, \ldots, 2k\}$ and $j \in \{1, \ldots, k\}$, and both extremities of P_i are in $V(C_1) \cup V(C_k)$;

(ii) the paths $P_1, \ldots, P_{2k}$ appear on each C_j in this order; and

(iii) for odd i the cycles $C_1, \ldots, C_k$ occur on P_i in this order, and for even i they occur in reverse order $C_k, \ldots, C_1$, for $i \in \{1, \ldots, 2k\}$.

Theorem 2.12 (Second main contribution) *Let $g(k) = (k+1)(\lfloor k/2 \rfloor + 1) - 1$ and $\mathcal{B}$ be a bramble of order $g(k)$ in a directed graph D. There is an* FPT *algorithm, with parameter k, that finds a path P that is a hitting set of $\mathcal{B}$ and contains a well-linked set A of order k.*

As a Corollary we get the following.

Corollary 2.13 *Let k be a non-negative integer and D be a directed graph. There is an* FPT *algorithm, with parameter k, that either*

(i) *produces an arboreal decomposition of width at most $f(k)$ for D; or*

(ii) *finds a cylindrical grid of order k as a butterfly minor of D.*

3 FPT algorithm for arboreal decompositions

The algorithm for arboreal decompositions given in [17] starts with a trivial decomposition $(\{r\}, \emptyset, \{W_r\})$ whose underlying arborescence contains only one vertex r. Thus, $W_r = V(G)$. Each iteration splits the vertices contained in an excessively large leaf of the current decomposition, if one exists, into a set of new leaves, while guaranteeing that the width of the non-leaf vertices remains bounded from above by a function of k. At any given iteration, the algorithm has to efficiently answer the following question:

Problem $\mathcal{P}$	
Input:	A directed graph D and a set $T \subseteq V(D)$ with $\lvert T\rvert \leq 2k-1$.
Parameters:	k and r.
Question:	Is there a set $Z \subseteq V(D)$ with at most r vertices such that every strong component C of $D \setminus Z$ satisfies $\lvert V(C) \cap T\rvert < k$?

Problem $\mathcal{P}$ can be naively solved by checking all $\binom{n}{r}$ sets Z size r in $V(D)$ and enumerating the strong components of $D \setminus Z$. Thus, it is in XP. In fact, this is the only step of the algorithm given in [17] which is done in XP time. In this section we show that it is also FPT , thus proving Theorem 2.9.

Theorem 3.1 *Problem $\mathcal{P}$ is* FPT .

Theorem 3.1, together with [17, Theorem 3.3], implies the existence of an FPT algorithm that returns a haven of order k on a directed graph D with $\mathsf{dtw}(D) > 3k-2$ as required in the first step of the proof of Corollary 2.13.

We make use of the following claim, also taken from [17]. We include a short proof for completeness.

Claim 3.2 *Let D be a graph and $T \subseteq V(D)$ with $|T| \leq 2k-1$. If $(D, T, k, k-1)$ is a negative instance for $\mathcal{P}$, then D admits a haven of order k.*

We refer the reader to [8, 21, 22, 24] for the formal definitions of multicuts and multiway cuts in the undirected and directed cases, and related results. We formally define only the variation of directed multicuts known as *linear cuts* [14]. In the Linear Edge Cut problem, we are given a graph D, a collection $\{T_1, \ldots, T_\ell\}$ of subsets of $V(D)$ and we want to find a minimum set of edges Z such that there is no path from T_i to T_j in $D' = (V, E - Z)$ whenever $j > i$. This problem is FPT when parameterized by the size of the solution.

Theorem 3.3 [14] *The* Linear Edge Cut *problem is* FPT *when parameterized by the size r of the solution, and can be solved in time $O(4^r \cdot r \cdot n^4)$.*

The vertex version is defined as follows.

Linear Vertex Cut	
Input:	A directed graph D and a collection of terminal sets $\mathcal{T}$, with $\mathcal{T} = \{T_1, T_2, \ldots T_\ell\}$, where $T_i \subseteq V(D)$ for $i \in \{1, \ldots, \ell\}$.
Parameter:	r.
Question:	Is there a set of vertices $Z \subseteq V(D)$ with $\lvert Z\rvert \leq r$ such that there are no paths in $D \setminus Z$ from T_i to T_j, for $1 \leq i < j \leq \ell$?

In this section, we show how to reduce the Linear Vertex Cut problem to the Linear Edge Cut problem. Thus by Theorem 3.3 we get an FPT algorithm for the Linear Vertex Cut problem.

Corollary 3.4 *There is an* FPT *algorithm for the* Linear Vertex Cut *problem parameterized by the size r of the solution and running in time $O(4^r \cdot r \cdot n^4)$.*

Next, we show that any solution to an instance of $\mathcal{P}$ is also a solution for a related instance of the Linear Vertex Cut problem.

Lemma 3.5 *Let (D, T, k, r) be an instance of $\mathcal{P}$. Then Z is a solution for (D, T, k, r) if and only if there is a partition $\mathcal{T}$ of $T - Z$ into sets $T_1, T_2, \ldots, T_\ell$*

such that $|T_i| \leq k-1$*, for* $i \in \{1, \ldots, \ell\}$*, and* Z *is a solution for the instance* $(D, \mathcal{T}, r)$ *of the* LINEAR VERTEX CUT *problem.*

The FPT algorithm for problem $\mathcal{P}$ follows from Lemma 3.5 and Corollary 3.4. The running time is heavily tied to the number of partitions $\mathcal{T}$ that can be generated from a given set T of an instance (D, T, k, r) of $\mathcal{P}$. This value is bounded by the k-th *ordered Bell number* [6]. The k-th ordered Bell number counts the number of ordered partitions of a set of size k and is of the form $2^{O(k \log k)}$.

Lemma 3.6 *Problem* $\mathcal{P}$ *is in* FPT *and can be solved in time* $2^{O(k \log k)} \cdot n^{O(1)}$.

We are now ready to state the main result of this section. We remark that the proof follows [17, Theorem 3.3] except that we replace the XP procedure of the proof by our FPT algorithm.

Theorem 3.7 *Let* D *be a directed graph and* k *a non-negative integer. There is an* FPT *algorithm, with parameter* k*, that either concludes that* D *admits a haven of order* k *or produces a nice arboreal decomposition of* D *of width at most* $3k-2$*. Furthermore, if the algorithm concludes there is a haven of order* k*, then it outputs a set* $T \subseteq V(D)$ *with* $|T| = 2k-1$ *such that* $(D, T, k, k-1)$ *is a negative instance of* $\mathcal{P}$*. Furthermore, the algorithm runs in time* $2^{O(k \log k)} \cdot n^{O(1)}$.

In the next section we provide the second, and last, step that needs to be changed in the proof contained in [18].

4 Finding a well-linked system of paths

It is known that, from any haven of order $k+1$ in a directed graph D, one can find a bramble of order $k/2$ in D [23, Lemma 6.4.20]. This procedure generates a bramble $\mathcal{B}'$ by enumerating all sets $\beta(Z)$ for all sets Z of size at most $k/2$, and we can verify whether a given set X is a hitting set of $\mathcal{B}'$ by going through all its elements. In Section 3 we provided and FPT algorithm, with parameter k, that either generates an arboreal decomposition of width bounded by a function of k or outputs a set T of size at most $2k-1$ such that $(D, T, k, k-1)$ is a negative instance of problem $\mathcal{P}$. In this section, we use T to identify a bramble $\mathcal{B}$ of high order that is better than $\mathcal{B}'$ in the following sense: with only a compact description of $\mathcal{B}$, we can use the methods described in Section 3 to compute a hitting set of $\mathcal{B}$ in FPT time with parameter k. In what follows, we define a special type of bramble that can be used to achieve this compact description and prove some properties of its hitting sets.

Definition 4.1 Let D be a directed graph and T be a subset of $V(D)$ such that $|T| \leq 2k-1$. A *T-bramble* of D is a bramble $\mathcal{B}$ such that $\mathcal{B} = \{B \subseteq D \mid B \text{ is induced, strongly connected, and } |V(B) \cap T| \geq k\}$.

For the remaining of this section, and unless stated otherwise, let $(D, T, k, k-1)$ be a negative instance of $\mathcal{P}$ and $\mathcal{B}_T$ be a T-bramble of D. First we show that $\mathsf{ord}(\mathcal{B}_T) = k$.

Claim 4.2 *$\mathcal{B}_T$ is a bramble of order k in D.*

Notice that, as $|T| \leq 2k-1$, any two elements of $\mathcal{B}_T$ intersect, and that any $\mathcal{B}' \subseteq \mathcal{B}_T$ is a bramble of order at most k. The next lemma is also needed to show the main result of this section.

Lemma 4.3 [18] *Let $\mathcal{B}$ be a bramble of a directed graph D. In polynomial time, one can always find a path P in D such that $V(P)$ is a hitting set of $\mathcal{B}$.*

We want to adapt the next lemma so that its statement can be verified in FPT time. This is precisely the second step needed to achieve an FPT algorithm for the Directed Grid Theorem.

Lemma 4.4 [18] *Let D be a digraph, $\mathcal{B}$ be a bramble of order $k(k+2)$, and $P = P(\mathcal{B})$ be a path intersecting every $B \in \mathcal{B}$. Then there is a set $A \subseteq V(P)$ of size k which is well-linked.*

To do that, we need to show how to split $\mathcal{B}$ into smaller brambles of order at least $\lfloor k/2 \rfloor$ which are intersected by subpaths of $P(\mathcal{B})$. We can grow a subpath of P' of P iteratively while checking, on each iteration, if the set of elements of $\mathcal{B}$ intersecting $V(P')$ is a bramble of adequate order. This can be done by enumerating all elements of the given bramble, which is not a procedure we can afford when aiming at an FPT algorithm. To do this more efficiently, we make use of some properties given by our very particular choice of the bramble $\mathcal{B}_T$.

Definition 4.5 Let $X \subseteq V(D)$ and $\mathcal{B}$ be a bramble of D. We define $\mathcal{B}(X)$ as the elements of $\mathcal{B}$ intersecting X. Similarly, we define $\bar{\mathcal{B}}(X)$ as the elements of $\mathcal{B}$ disjoint from X. Formally, $\mathcal{B}(X) = \{B \in \mathcal{B} \mid V(B) \cap X \neq \emptyset\}$ and $\bar{\mathcal{B}}(X) = \{B \in \mathcal{B} \mid V(B) \cap X = \emptyset\}$.

Notice that $\mathcal{B}(X)$ and $\bar{\mathcal{B}}(X)$ are brambles, as both are subsets of a bramble $\mathcal{B}$. Additionally, $\mathcal{B}(X)$ is disjoint from $\bar{\mathcal{B}}(X)$ and the union of a hitting set for the former with a hitting set for the latter is a hitting set for $\mathcal{B}$. From this remark, we have that

$$\mathsf{ord}(\mathcal{B}(X)) + \mathsf{ord}(\bar{\mathcal{B}}(X)) \geq \mathsf{ord}(\mathcal{B})$$

and, although it may be hard to compute the order of $\mathcal{B}(X)$, we can estimate its order by knowing the order of its "complement bramble" $\bar{\mathcal{B}}(X)$.

Consider now the brambles $\mathcal{B}_T$, $\mathcal{B}_T(X)$ and $\bar{\mathcal{B}}_T(X)$. The following claims show that hitting sets for $\bar{\mathcal{B}}_T(X)$ are exactly solutions for the instance $(D \setminus X, T - X, k, k)$ of $\mathcal{P}$ and, by Lemma 3.6, can be found in FPT time.

Claim 4.6 *Let $X, Z \subseteq V(D)$. There is a $B \in \bar{\mathcal{B}}_T(X)$ such that $V(B) \cap Z = \emptyset$ if and only if $D \setminus (Z \cup X)$ contains a strongly connected component C such that $|V(C) \cap (T - X)| \geq k$.*

The contrapositive of Claim 4.6 characterizes hitting sets of $\bar{\mathcal{B}}_T(X)$.

Claim 4.7 *Let $X, Z \subseteq V(D)$. Z is a hitting set of $\bar{\mathcal{B}}_T(X)$ if and only if every strong component of $D \setminus (X \cup Z)$ contains at most $k-1$ vertices of $T - X$.*

Therefore we can decide if $\mathsf{ord}(\bar{\mathcal{B}}_T(X)) \leq r$ by solving the instance $(D \backslash X, T - X, k, r)$ of $\mathcal{P}$. The following result is a direct consequence of Lemma 3.6 and Claim 4.7.

Corollary 4.8 *For any $X \subseteq V(D)$, there is an* FPT *algorithm, running in time $2^{O(k \log k)} \cdot n^{O(1)}$, that decides whether $\mathsf{ord}(\bar{\mathcal{B}}_T(X)) \leq r$ or if $\mathsf{ord}(\bar{\mathcal{B}}_T(X)) \geq r + 1$.*

Finally, we follow the original proof from [18, Lemma 4.4], but choosing the subpaths of P in FPT time.

Lemma 4.9 *Let D be a directed graph, $g(k) = (k+1)(\lfloor k/2 \rfloor + 1) - 1$, $T \subseteq V(D)$ with $|T| = 2g(k) - 1$ and assume that $(D, T, g(k), g(k) - 1)$ is a negative instance of $\mathcal{P}$. Let $\mathcal{B}_T$ be a T-bramble of order $g(k)$ of D and P be a path intersecting every element of $\mathcal{B}_T$. Then there is a set $A \subseteq V(P)$ of order k which is well-linked. Furthermore A can be found in* FPT *time with parameter k.*

The last lemma shows how to find a large well-linked set in a directed graph D of large directed tree-width. By following the remaining of the proof of the Directed Grid Theorem [18], which yields FPT algorithms for all the remaining steps, we can validate Corollary 2.13.

References

[1] Bang-Jensen, J. and G. Gregory, "Classes of Directe Graphs," Springer Monographs in Mathematics, 2008.

[2] Bertele, U. and F. Brioschi, "Nonserial Dynamic Programming," Academic Press, Inc., Orlando, FL, USA, 1972.

[3] Bodlaender, H. L., *Treewidth: characterizations, applications, and computations*, in: *Proc. of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, LNCS **4271**, 2006, pp. 1–14.

[4] Bondy, A. and M. R. Murty, "Graph Theory," Springer-Verlag London, 2008.

[5] Bousquet, N., J. Daligault and S. Thomass, *Multicut is fpt*, SIAM Journal on Computing **47** (2018), pp. 166–207.
URL https://doi.org/10.1137/140961808

[6] Cayley, A., "On the analytical forms called trees. Second part," Cambridge Library Collection - Mathematics **4**, Cambridge University Press, 2009 p. 112115.

[7] Chekuri, C. and J. Chuzhoy, *Polynomial bounds for the grid-minor theorem*, Journal of the ACM **63** (2016), pp. 40:1–40:65.
URL http://doi.acm.org/10.1145/2820609

[8] Chitnis, R., M. Hajiaghayi and D. Marx, *Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset*, in: *Proc. of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2012, pp. 1713–1725.

[9] Cook, W. and P. D. Seymour, *Tour merging via branch-decompositions*, INFORMS Journal on Computing **15** (2003), pp. 233–248.

[10] Courcelle, B., *The monadic second-order logic of graphs. I. Recognizable sets of finite graphs*, Information and Computation **85** (1990), pp. 12–75.
URL http://www.sciencedirect.com/science/article/pii/089054019090043H

[11] Cygan, M., F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk and S. Saurabh, "Parameterized Algorithms," Springer, 2015.

[12] Demaine, D., V. Fomin, M. Hajiaghayi and D. M. Thilikos, *Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs*, Journal of the ACM **52** (2005), pp. 866–893.

[13] Downey, R. G. and M. R. Fellows, "Fundamentals of Parameterized Complexity," Texts in Computer Science, Springer, 2013.

[14] Erbacher, R. F., T. Jaeger, N. Talele and J. Teutsch, *Directed multicut with linearly ordered terminals*, CoRR **abs/1407.7498** (2014).

[15] Grohe, M., K.-i. Kawarabayashi, D. Marx and P. Wollan, *Finding topological subgraphs is fixed-parameter tractable*, in: *Proc. of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, 2011, pp. 479–488.

[16] Halin, R., *S-functions for graphs*, Journal of Geometry **8** (1976), pp. 171–186.
URL https://doi.org/10.1007/BF01917434

[17] Johnson, T., N. Robertson, P. D. Seymour and R. Thomas, *Directed tree-width*, Journal of Combinatorial Theory, Series B **82** (2001), pp. 138–154.

[18] Kawarabayashi, K.-i. and S. Kreutzer, *The Directed Grid Theorem*, in: *Proc. of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, 2015, pp. 655–664.
URL http://doi.acm.org/10.1145/2746539.2746586

[19] Kleinberg, J. M., *Decision algorithms for unsplittable flow and the half-disjoint paths problem*, in: *Proc. of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, 1998, pp. 530–539.

[20] Koster, A., S. van Hoesel and A. Kolen, *Solving frequency assignment problems via tree-decomposition*, Electronic Notes in Discrete Mathematics **3** (1999), pp. 102–105.

[21] Kratsch, S., M. Pilipczuk, M. Pilipczuk and M. Wahlstrm, *Fixed-parameter tractability of multicut in directed acyclic graphs*, SIAM Journal on Discrete Mathematics **29** (2015), pp. 122–144.
URL https://doi.org/10.1137/120904202

[22] Marx, D. and I. Razgon, *Fixed-parameter tractability of multicut parameterized by the size of the cutset*, SIAM Journal on Computing **43** (2014), pp. 355–388.
URL https://doi.org/10.1137/110855247

[23] Matthias, D. and F. Emmert-Streib, "Quantitative Graph Theory," Discrete Mathematics and its Applications, Chapman and Hall/CRC, 2014.

[24] Pilipczuk, M. and M. Wahlström, *Directed multicut is w[1]-hard, even for four terminal pairs*, ACM Transactions on Computation Theory **10** (2018), pp. 13:1–13:18.
URL http://doi.acm.org/10.1145/3201775

[25] Reed, B., *Introducing directed tree-width*, Electronic Notes in Discrete Mathematics **3** (1999), pp. 222–229.

[26] Robertson, N. and P. D. Seymour, *Graph minors. V. Excluding a planar graph*, Journal of Combinatorial Theory, Series B **41** (1986), pp. 92–114.

[27] Robertson, N. and P. D. Seymour, *Graph minors. XIII. The disjoint paths problem*, Journal of Combinatorial Theory, Series B **63** (1995), pp. 65–110.
URL http://www.sciencedirect.com/science/article/pii/S0095895685710064

[28] Seymour, P. D. and R. Thomas, *Graph searching and a min-max theorem for tree-width*, Journal of Combinatorial Theory, Series B **58** (1993), pp. 22–33.
URL http://www.sciencedirect.com/science/article/pii/S0095895683710270