



HAL
open science

k-apices of Minor-closed Graph Classes. II. Parameterized Algorithms

Ignasi Sau, Giannos Stamoulis, Dimitrios M. Thilikos

► **To cite this version:**

Ignasi Sau, Giannos Stamoulis, Dimitrios M. Thilikos. k-apices of Minor-closed Graph Classes. II. Parameterized Algorithms. ACM Transactions on Algorithms, 2022, 18 (3), pp.1-30. <10.1145/3519028>. <hal-03835923>

HAL Id: hal-03835923

<https://hal.science/hal-03835923v1>

Submitted on 17 Oct 2023

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.



HAL Authorization

k -apices of minor-closed graph classes. II. Parameterized algorithms¹Ignasi Sau²Giannos Stamoulis³Dimitrios M. Thilikos²**Abstract**

Let \mathcal{G} be a minor-closed graph class. We say that a graph G is a k -*apex* of \mathcal{G} if G contains a set S of at most k vertices such that $G \setminus S$ belongs to \mathcal{G} . We denote by $\mathcal{A}_k(\mathcal{G})$ the set of all graphs that are k -apices of \mathcal{G} . In the first paper of this series we obtained upper bounds on the size of the graphs in the minor-obstruction set of $\mathcal{A}_k(\mathcal{G})$, i.e., the minor-minimal set of graphs not belonging to $\mathcal{A}_k(\mathcal{G})$. In this article we provide an algorithm that, given a graph G on n vertices, runs in time $2^{\text{poly}(k)} \cdot n^3$ and either returns a set S certifying that $G \in \mathcal{A}_k(\mathcal{G})$, or reports that $G \notin \mathcal{A}_k(\mathcal{G})$. Here poly is a polynomial function whose degree depends on the maximum size of a minor-obstruction of \mathcal{G} . In the special case where \mathcal{G} excludes some apex graph as a minor, we give an alternative algorithm running in $2^{\text{poly}(k)} \cdot n^2$ -time.

Keywords: graph minors; parameterized algorithms; graph modification problems; irrelevant vertex technique; Flat Wall Theorem.

¹A conference version of this paper appeared in the *Proceedings of the 47th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 168 of *LIPICs*, pages 95:1–95:20, 2020.

²LIRMM, Université de Montpellier, CNRS, Montpellier, France. Supported by the ANR projects DEMOGRAPH (ANR-16-CE40-0028), ESIGMA (ANR-17-CE23-0010), ELIT (ANR-20-CE48-0008), the French-German Collaboration ANR/DFG Project UTMA (ANR-20-CE92-0027), and the French Ministry of Europe and Foreign Affairs, via the Franco-Norwegian project PHC AURORA. Emails: ignasi.sau@lirmm.fr, sedthilk@thilikos.info.

³LIRMM, Université de Montpellier, Montpellier, France. Email: gstamoulis@lirmm.fr.

Contents

1	Introduction	3
2	Definitions and preliminary results	6
2.1	Restating the problem	6
2.2	Preliminaries	6
3	Flat walls	8
3.1	Walls and subwalls	8
3.2	Paintings and renditions	10
3.3	Flatness pairs	11
3.4	Homogeneous walls	13
4	Auxiliary algorithmic and combinatorial results	16
4.1	Finding an irrelevant vertex	16
4.2	Combinatorial results for branching	18
5	The general algorithm	19
5.1	Iterative compression	20
5.2	The algorithm	20
6	The apex-minor free case	22
6.1	Finding an apex graph as a minor	22
6.2	Quickly finding a wall	24
6.3	The algorithm	26
6.4	Correctness of the algorithm	28
7	Algorithms for variants of VERTEX DELETION TO \mathcal{G}	29
7.1	The general framework	29
7.2	Variants of VERTEX DELETION TO \mathcal{G}	29
8	Discussion and concluding remarks	31

1 Introduction

Graph modification problems are fundamental in algorithmic graph theory. Typically, such a problem is determined by a graph class \mathcal{G} and some prespecified set \mathcal{M} of *local* modifications, such as vertex/edge removal or edge addition/contraction or combinations of them, and the question is, given a graph G and an integer k , whether it is possible to transform G to a graph in \mathcal{G} by applying k modification operations from \mathcal{M} . A plethora of graph problems can be formulated for different instantiations of \mathcal{G} and \mathcal{M} . Applications span diverse topics such as computational biology, computer vision, machine learning, networking, and sociology [25]. As reported by Roded Sharan in [53], already in 1979 Garey and Johnson mentioned 18 different types of modification problems [26, Section A1.2]. For more on graph modification problems, see [10, 25] as well as the running survey in [14]. In this paper we focus our attention on the vertex deletion operation. We say that a graph G is a k -*apex* of a graph class \mathcal{G} if there is a set $S \subseteq V(G)$ of size at most k such that the removal of S from G results in a graph in \mathcal{G} . In other words, we consider the following meta-problem.

VERTEX DELETION TO \mathcal{G}

Input: A graph G and a non-negative integer k .

Objective: Find, if it exists, a set $S \subseteq V(G)$, certifying that G is k -apex of \mathcal{G} .

To illustrate the expressive power of **VERTEX DELETION TO \mathcal{G}** , if \mathcal{G} is the class of edgeless (resp. acyclic, planar, bipartite, (proper) interval, chordal) graphs, we obtain the **VERTEX COVER** (resp. **FEEDBACK VERTEX SET**, **VERTEX PLANARIZATION**, **ODD CYCLE TRANSVERSAL**, **(PROPER) INTERVAL VERTEX DELETION**, **CHORDAL VERTEX DELETION**) problem.

By the classical result of Lewis and Yannakakis [40], **VERTEX DELETION TO \mathcal{G}** is NP-hard for every non-trivial graph class \mathcal{G} . To circumvent its intractability, we study it from the parameterized complexity point of view and we parameterize it by the number k of vertex deletions. In this setting, the most desirable behavior is the existence of an algorithm running in time $f(k) \cdot n^{\mathcal{O}(1)}$, where f is a computable function depending only on k . Such an algorithm is called *fixed-parameter tractable*, or FPT-algorithm for short, and a parameterized problem admitting an FPT-algorithm is said to belong to the parameterized complexity class FPT. Also, the function f is called *parametric dependence* of the corresponding FPT-algorithm, and the challenge is to design FPT-algorithms with small parametric dependencies [15, 18, 21, 43].

Unfortunately, we cannot hope for the existence of FPT-algorithms for every graph class \mathcal{G} . Indeed, the problem is W-hard¹ for some classes \mathcal{G} that are closed under induced subgraphs [41] or, even worse, NP-hard, for $k = 0$, for every class \mathcal{G} whose recognition problem is NP-hard, such as some classes closed under subgraphs or induced subgraphs (for instance 3-colorable graphs), edge contractions [12], or induced minors [19].

On the positive side, a very relevant subset of classes of graphs *does* allow for FPT-algorithms. These are classes \mathcal{G} that are closed under minors², or *minor-closed*. To see this, we define $\mathcal{A}_k(\mathcal{G})$ as the class of the k -apices of \mathcal{G} , i.e., the **yes**-instances of **VERTEX DELETION TO \mathcal{G}** , and observe that

¹Implying that an FPT-algorithm would result in an unexpected complexity collapse; see [18].

²A graph H is a *minor* of a graph G if it can be obtained from a subgraph of G by contracting edges, see [Subsection 2.2](#) for the formal definitions.

if \mathcal{G} is minor-closed then the same holds for $\mathcal{A}_k(\mathcal{G})$ for every $k \geq 0$. This, in turn, implies that for every k , $\mathcal{A}_k(\mathcal{G})$ can be characterized by a set \mathcal{F}_k of minor-minimal graphs that are not in $\mathcal{A}_k(\mathcal{G})$; we call these graphs the *obstructions* of $\mathcal{A}_k(\mathcal{G})$ and we know that they are finite because of the Robertson and Seymour’s theorem [47]. In other words, we know that the size of the obstruction set of $\mathcal{A}_k(\mathcal{G})$ is bounded by some function of k . Then one can decide whether a graph G belongs to $\mathcal{A}_k(\mathcal{G})$ by checking whether G excludes all members of the obstruction set of $\mathcal{A}_k(\mathcal{G})$, and this can be checked by using the FPT-algorithm in [46] (see also [20]).

As the Robertson and Seymour’s theorem [47] does *not* construct \mathcal{F}_k , the aforementioned argument is not constructive, i.e., it is not able to construct the claimed FPT-algorithm. An important step towards the constructibility of such an FPT-algorithm was done by Adler et al. [2], who proved that \mathcal{F}_k is effectively computable. In the first paper of this series [52] we give an *explicit upper bound* on the size of the graphs in \mathcal{F}_k , namely we prove that every graph in \mathcal{F}_k has size bounded by an exponential tower of height four of a polynomial function in k , whose degree depends on the size of the minor-obstructions of \mathcal{G} . The focus of the current paper is on the parametric dependence of FPT-algorithms to solve the VERTEX DELETION TO \mathcal{G} problem, i.e., for recognizing the class $\mathcal{A}_k(\mathcal{G})$.

The task of specifying (or even optimizing) this parametric dependence for different instantiations of \mathcal{G} occupied a considerable part of research in parameterized algorithms. The most general result in this direction states that, for every t , there is some constant c such that if the graphs in \mathcal{G} have treewidth at most t , then VERTEX DELETION TO \mathcal{G} admits an FPT-algorithm that runs in time $c^k \cdot n^{\mathcal{O}(1)}$ [23, 35]. Reducing the constant c in this running time has attracted research on particular problems such as VERTEX COVER [13] (with $c = 1.2738$), FEEDBACK VERTEX SET [37] (with $c = 3.619$), APEX-PSEUDOFORREST [11] (with $c = 3$), PATHWIDTH 1 VERTEX DELETION (with $c = 4.65$) [16], or PUMPKIN VERTEX DELETION [30]. The first step towards a parameterized algorithm for VERTEX DELETION TO \mathcal{G} for cases where \mathcal{G} has unbounded treewidth was done in [42] and later in [31] for the VERTEX PLANARIZATION problem, and the best parameterized dependence for this problem is $2^{\mathcal{O}(k \cdot \log k)} \cdot n$, achieved by Jansen et al. [29]. These results were later extended by Kociumaka and Marcin Pilipczuk [38], who proved that if \mathcal{G}_g is the class of graphs of Euler genus at most g , then VERTEX DELETION TO \mathcal{G}_g admits a $2^{\mathcal{O}_g(k^2 \cdot \log k)} \cdot n^{\mathcal{O}(1)}$ -time³ algorithm.

Our results. In this paper we give an explicit FPT-algorithm for VERTEX DELETION TO \mathcal{G} for every fixed minor-closed graph class \mathcal{G} . In particular, our main results are the following.

Theorem 1. *If \mathcal{G} is a minor-closed graph class, then VERTEX DELETION TO \mathcal{G} admits an algorithm running in time $2^{\text{poly}(k)} \cdot n^3$, for some polynomial poly whose degree depends on \mathcal{G} .*

We say that a graph H is an *apex* graph if it is a 1-apex of the class of planar graphs.

Theorem 2. *If \mathcal{G} is a minor-closed graph class excluding some apex graph, then VERTEX DELETION TO \mathcal{G} admits an algorithm running in time $2^{\text{poly}(k)} \cdot n^2$, for some polynomial poly whose degree depends on \mathcal{G} .*

In Section 7 we explain how the algorithms of Theorem 1 and Theorem 2 can be modified in order to apply to a series of variants of VERTEX DELETION TO \mathcal{G} .

³Given a tuple $\mathbf{t} = (x_1, \dots, x_\ell) \in \mathbb{N}^\ell$ and two functions $\chi, \psi : \mathbb{N} \rightarrow \mathbb{N}$, we write $\chi(n) = \mathcal{O}_{\mathbf{t}}(\psi(n))$ in order to denote that there exists a computable function $\phi : \mathbb{N}^\ell \rightarrow \mathbb{N}$ such that $\chi(n) = \mathcal{O}(\phi(\mathbf{t}) \cdot \psi(n))$.

Our techniques. We provide here just a very succinct enumeration of the techniques that we use in order to achieve [Theorem 1](#) and [Theorem 2](#); a more detailed description with the corresponding definitions is provided, along with the algorithms, in the next sections.

Our starting point to prove [Theorem 1](#) is to use the standard iterative compression technique of Reed et al. [[45](#)] ([Lemma 21](#)). This allows us to assume that we have at hand a slightly too large set $S \subseteq V(G)$ such that $G \setminus S \in \mathcal{G}$. We run the algorithm of [Proposition 10](#) from [[51](#)] that (since $G \setminus S \in \mathcal{G}$) either concludes that the treewidth of G is polynomially bounded by k , or finds a large *flat* wall W together with an apex set A . In the first case, we use the main algorithmic result of Baste et al. [[5](#)] ([Proposition 4](#)) to solve the problem parameterized by treewidth, achieving the claimed running time. [Proposition 10](#) is an improved version of the original “Flat Wall Theorem” of Robertson and Seymour [[46](#)], whose proof is based on the recent results of Kawarabayashi et al. [[33](#)], which we state using the framework that we recently introduced in [[51](#)]. This framework is presented in [Section 3](#) and provides the formal definitions of a series of combinatorial concepts such as paintings and renditions ([Subsection 3.2](#)), flatness pairs and tilts ([Subsection 3.3](#)), as well as a notion of wall homogeneity ([Subsection 3.4](#)) alternative to the one given in [[46](#)]. All these concepts are extensively used in our proofs, as well as in those in the first article of this series [[52](#)].

Once we have the large flat wall W and the apex set A , we see how many vertices of $S \cup A$ have enough neighbors in the “interior” of W . Two possible scenarios may occur. If the “interior” of W has enough neighbors in the set $S \cup A$, we apply a combinatorial result of [[52](#)] ([Proposition 18](#)), based on the notion of *canonical partition* of a wall, that guarantees that every possible solution should intersect $S \cup A$, and we can branch on it.

On the other hand, if the interior of W has few neighbors in $S \cup A$, we find in W a packing of an appropriate number of pairwise disjoint large enough subwalls ([Proposition 19](#)) and we find a subwall whose interior has few (a function not depending on k) neighbors in $S \cup A$. We then argue that we can define from it a *flat* wall in which we can apply the irrelevant vertex technique of Robertson and Seymour [[46](#)] ([Corollary 17](#)). We stress that this flat subwall is not precisely a subwall of W but a tiny “tilt” of a subwall of W , a concept introduced in [[51](#)] that is necessary for our proofs. In order to apply the irrelevant vertex technique, the main combinatorial tool is [Proposition 15](#), which as been proved in [[52](#)] and that is an enhancement of a result of Baste et al. [[5](#)], as we discuss in [Subsection 4.1](#).

In order to achieve the improved running time claimed in [Theorem 2](#), we do *not* use iterative compression. Instead, we directly invoke [Lemma 26](#), which is a variation of [[51](#), [Lemma 11](#)] and whose proof uses [[1](#), [3](#), [32](#), [44](#)], that either reports that we have a no-instance, or concludes that the treewidth of G is polynomially bounded by k , or finds a large wall W in G . If the treewidth is small, we proceed as above. If a large wall is found, we apply [Proposition 10](#) and we now distinguish two cases. If a large flat wall is found, we find an irrelevant vertex using again [Lemma 16](#). Otherwise, inspired by an idea of Marx and Schlotter [[42](#)], we exploit the fact that \mathcal{G} excludes an apex graph, and we use flow techniques to either find a vertex that should belong to the solution, or to conclude that we are dealing with a no-instance.

Organization of the paper. In [Section 2](#) we give some basic definitions and preliminary results. In [Section 3](#) we introduce flat walls along with all the concepts and results around the Flat Wall Theorem, using the framework of [[51](#)]. In [Section 4](#) we present several algorithmic and combinatorial

results that will be used in the algorithms, when finding an irrelevant vertex or when applying the branching step. In [Section 5](#) and [Section 6](#) we present the main algorithms claimed in [Theorem 1](#) and [Theorem 2](#), respectively. In [Section 7](#) we explain how to modify our algorithms so to deal with a series of variants of the VERTEX DELETION TO \mathcal{G} problem. We conclude in [Section 8](#) with some directions for further research.

2 Definitions and preliminary results

Our first step is to restate the problem in a more convenient way. We next give some basic definitions and preliminary results.

2.1 Restating the problem

Let \mathcal{F} be a finite non-empty collection of non-empty graphs. We use $\mathcal{F} \leq_m G$ to denote that some graph in \mathcal{F} is a minor of G .

Given a graph class \mathcal{G} , its *minor obstruction set* is defined as the set of all minor-minimal graphs that are not in \mathcal{G} , and is denoted by $\mathbf{obs}(\mathcal{G})$. Given a finite non-empty collection of non-empty graphs \mathcal{F} , we denote by $\mathbf{exc}(\mathcal{F})$ as the set containing every graph G that excludes all graphs in \mathcal{F} as minors.

Let \mathcal{G} be a minor-closed graph class and \mathcal{F} be its obstruction set. Clearly, VERTEX DELETION TO \mathcal{G} is the same problem as asking, given a graph G and some $k \in \mathbb{N}$, for a vertex set S of at most k vertices such that $G \setminus S \in \mathbf{exc}(\mathcal{F})$. Following the terminology of [\[5–8, 23, 24, 35, 36\]](#), we call this problem \mathcal{F} -M-DELETION.

Some conventions. In what follows we always denote by \mathcal{F} the set $\mathbf{obs}(\mathcal{G})$ of the instantiation of VERTEX DELETION TO \mathcal{G} that we consider. Notice that, given a graph G and an integer k , (G, k) is a *yes-instance* of \mathcal{F} -M-DELETION if and only if $G \in \mathcal{A}_k(\mathbf{exc}(\mathcal{F}))$. Given a graph G , we define its *apex number* to be the smallest integer a for which G is an a -apex of the class of planar graphs. Also, we define the *detail* of G , denoted by $\mathbf{detail}(G)$, to be the maximum among $|E(G)|$ and $|V(G)|$. We define three constants depending on \mathcal{F} that will be used throughout the paper whenever we consider such a collection \mathcal{F} . We define $a_{\mathcal{F}}$ as the minimum apex number of a graph in \mathcal{F} , we set $s_{\mathcal{F}} = \max\{|V(H)| \mid H \in \mathcal{F}\}$, and we set $\ell_{\mathcal{F}} = \max\{\mathbf{detail}(H) \mid H \in \mathcal{F}\}$. Unless stated otherwise, we denote by n and m the number of vertices and edges, respectively, of the graph under consideration. We can always assume that G has $\mathcal{O}_{s_{\mathcal{F}}}(k \cdot n)$ edges, otherwise we can directly conclude that (G, k) is a *no-instance* (for this, use the fact that graphs excluding some graph as a minor are sparse [\[39, 55\]](#)).

2.2 Preliminaries

Sets and integers. We denote by \mathbb{N} the set of non-negative integers. Given two integers p and q , the set $[p, q]$ contains every integer r such that $p \leq r \leq q$. For an integer $p \geq 1$, we set $[p] = [1, p]$ and $\mathbb{N}_{\geq p} = \mathbb{N} \setminus [0, p - 1]$. Given a non-negative integer x , we denote by $\mathbf{odd}(x)$ the minimum odd number that is not smaller than x . For a set S , we denote by 2^S the set of all subsets of S and, given an integer $r \in [|S|]$, we denote by $\binom{S}{r}$ the set of all subsets of S of size r and by $\binom{S}{< r}$ the set

of all subsets of S of size at most r . If \mathcal{S} is a collection of objects where the operation \cup is defined, then we denote $\mathbf{U}\mathcal{S} = \cup_{X \in \mathcal{S}} X$.

Basic concepts on graphs. All graphs considered in this paper are undirected, finite, and without loops or multiple edges. We use standard graph-theoretic notation and we refer the reader to [17] for any undefined terminology. Let G be a graph. We say that a pair $(L, R) \in 2^{V(G)} \times 2^{V(G)}$ is a *separation* of G if $L \cup R = V(G)$ and there is no edge in G between $L \setminus R$ and $R \setminus L$. Given a vertex $v \in V(G)$, we denote by $N_G(v)$ the set of vertices of G that are adjacent to v in G . A vertex $v \in V(G)$ is *isolated* if $N_G(v) = \emptyset$. For $S \subseteq V(G)$, we set $G[S] = (S, E \cap \binom{S}{2})$ and use the shortcut $G \setminus S$ to denote $G[V(G) \setminus S]$. Given a vertex $v \in V(G)$ of degree two with neighbors u and w , we define the *dissolution* of v to be the operation of deleting v and, if u and w are not adjacent, adding the edge $\{u, w\}$. Given two graphs H, G , we say that H is a *dissolution* of G if H can be obtained from G after dissolving vertices of G . Given an edge $e = \{u, v\} \in E(G)$, we define the *subdivision* of e to be the operation of deleting e , adding a new vertex w and making it adjacent to u and v . Given two graphs H and G , we say that H is a *subdivision* of G if H can be obtained from G after subdividing edges of G .

Treewidth. A *tree decomposition* of a graph G is a pair (T, χ) where T is a tree and $\chi : V(T) \rightarrow 2^{V(G)}$ such that

- $\cup_{t \in V(T)} \chi(t) = V(G)$,
- for every edge e of G there is a $t \in V(T)$ such that $\chi(t)$ contains both endpoints of e , and
- for every $v \in V(G)$, the subgraph of T induced by $\{t \in V(T) \mid v \in \chi(t)\}$ is connected.

The *width* of (T, χ) is equal to $\max\{|\chi(t)| - 1 \mid t \in V(T)\}$ and the *treewidth* of G , denoted by $\text{tw}(G)$, is the minimum width over all tree decompositions of G .

To compute a tree decomposition of a graph of bounded treewidth, in the proof of [Lemma 26](#) in [Section 6](#) we will use the single-exponential 5-approximation algorithm for treewidth of Bodlaender et al. [9, Theorem VI].

Proposition 3. *There is an algorithm that, given an graph G and an integer k , outputs either a report that $\text{tw}(G) > k$, or a tree decomposition of G of width at most $5k + 4$. Moreover, this algorithm runs in $2^{\mathcal{O}(k)} \cdot n$ -time.*

Contractions and minors. The *contraction* of an edge $e = \{u, v\}$ of a simple graph G results in a simple graph G' obtained from $G \setminus \{u, v\}$ by adding a new vertex uv adjacent to all the vertices in the set $N_G(u) \cup N_G(v) \setminus \{u, v\}$. A graph G' is a *minor* of a graph G , denoted by $G' \leq_m G$, if G' can be obtained from G by a sequence of vertex removals, edge removals, and edge contractions. If only edge contractions are allowed, we say that G' is a *contraction* of G . Given two graphs H and G , if H is a minor of G then for every vertex $v \in V(H)$ there is a set of vertices in G that are the endpoints of the edges of G contracted towards creating v . We call this set *model* of v in G . Recall that, given a finite collection of graphs \mathcal{F} and a graph G , we use notation $\mathcal{F} \leq_m G$ to denote that some graph in \mathcal{F} is a minor of G .

We present here the main result of Baste et al. [5], which we will use in order to solve \mathcal{F} -M-DELETION on instances of treewidth bounded by an appropriate function of k .

Proposition 4. *Let \mathcal{F} be a finite collection of graphs. There exists an algorithm that, given a triple (G, tw, k) where G is a graph of treewidth at most tw and k is a non-negative integer, it outputs, if it exists, a vertex set S of G of size at most k such that $G \setminus S \in \text{exc}(\mathcal{F})$. Moreover, this algorithm runs in $2^{\mathcal{O}_{s\mathcal{F}}(\text{tw} \cdot \log \text{tw})} \cdot n$ -time.*

3 Flat walls

In this section we deal with flat walls, using the framework of [51]. More precisely, in [Subsection 3.1](#), we introduce walls and several notions concerning them. In [Subsection 3.2](#), we provide the definitions of a rendition and a painting. Using the above notions, in [Subsection 3.3](#), we define flat walls and provide some results about them, including the Flat Wall Theorem (namely, the version proved by Kawarabayashi et al. [33]) and its algorithmic version restated in the “more accurate” framework of [51]. Finally, in [Subsection 3.4](#), we present the notion of homogeneity and an algorithm from [51] that allows us to detect a homogenous flat wall “inside” a given flat wall of “big enough” height. We note that the definitions of this section can also be found in [51, 52].

3.1 Walls and subwalls

We start with some basic definitions about walls.

Walls. Let $k, r \in \mathbb{N}$. The $(k \times r)$ -grid is the graph whose vertex set is $[k] \times [r]$ and two vertices (i, j) and (i', j') are adjacent if and only if $|i - i'| + |j - j'| = 1$. An *elementary r -wall*, for some odd integer $r \geq 3$, is the graph obtained from a $(2r \times r)$ -grid with vertices $(x, y) \in [2r] \times [r]$, after the removal of the “vertical” edges $\{(x, y), (x, y + 1)\}$ for odd $x + y$, and then the removal of all vertices of degree one. Notice that, as $r \geq 3$, an elementary r -wall is a planar graph that has a unique (up to topological isomorphism) embedding in the plane \mathbb{R}^2 such that all its finite faces are incident to exactly six edges. The *perimeter* of an elementary r -wall is the cycle bounding its infinite face, while the cycles bounding its finite faces are called *bricks*. Also, the vertices in the perimeter of an elementary r -wall that have degree two are called *pegs*, while the vertices $(1, 1), (2, r), (2r - 1, 1), (2r, r)$ are called *corners* (notice that the corners are also pegs).

An r -wall is any graph W obtained from an elementary r -wall \bar{W} after subdividing edges (see [Figure 1](#)). A graph W is a *wall* if it is an r -wall for some odd $r \geq 3$ and we refer to r as the *height* of W . Given a graph G , a *wall of G* is a subgraph of G that is a wall. We insist that, for every r -wall, the number r is always odd.

We call the vertices of degree three of a wall W *3-branch vertices*. A cycle of W is a *brick* (resp. the *perimeter*) of W if its 3-branch vertices are the vertices of a brick (resp. the perimeter) of \bar{W} . We denote by $\mathcal{C}(W)$ the set of all cycles of W . We use $D(W)$ in order to denote the perimeter of the wall W . A brick of W is *internal* if it is disjoint from $D(W)$.

Subwalls. Given an elementary r -wall \bar{W} , some odd $i \in \{1, 3, \dots, 2r - 1\}$, and $i' = (i + 1)/2$, the i' -th *vertical path* of \bar{W} is the one whose vertices, in order of appearance, are $(i, 1), (i, 2), (i +$

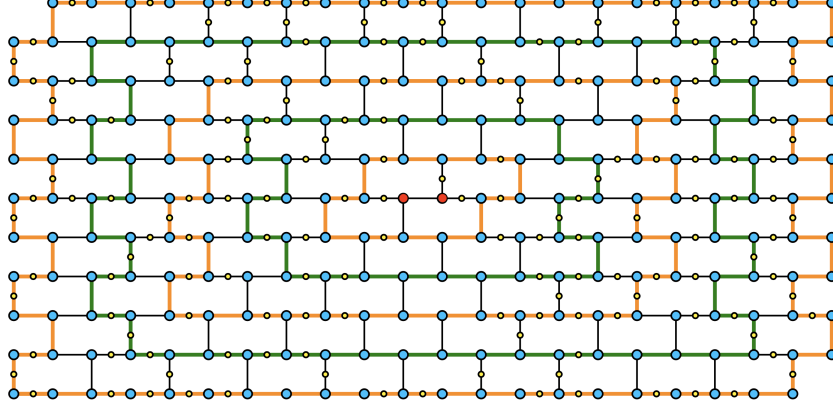


Figure 1: An 11-wall and its five layers, depicted in alternating orange and green. The central vertices of the wall are depicted in red.

$1, 2), (i + 1, 3), (i, 3), (i, 4), (i + 1, 4), (i + 1, 5), (i, 5), \dots, (i, r - 2), (i, r - 1), (i + 1, r - 1), (i + 1, r)$. Also, given some $j \in [2, r - 1]$ the j -th horizontal path of \bar{W} is the one whose vertices, in order of appearance, are $(1, j), (2, j), \dots, (2r, j)$.

A vertical (resp. horizontal) path of an r -wall W is one that is a subdivision of a vertical (resp. horizontal) path of \bar{W} . Notice that the perimeter of an r -wall W is uniquely defined regardless of the choice of the elementary r -wall \bar{W} . A subwall of W is any subgraph W' of W that is an r' -wall, with $r' \leq r$, and such the vertical (resp. horizontal) paths of W' are subpaths of the vertical (resp. horizontal) paths of W .

Layers. The layers of an r -wall W are recursively defined as follows. The first layer of W is its perimeter. For $i = 2, \dots, (r - 1)/2$, the i -th layer of W is the $(i - 1)$ -th layer of the subwall W' obtained from W after removing from W its perimeter and removing recursively all occurring vertices of degree one. We refer to the $(r - 1)/2$ -th layer as the inner layer of W . The central vertices of an r -wall are its two branch vertices that do not belong to any of its layers. See Figure 1 for an illustration of the notions defined above.

Central walls. Given an r -wall W and an odd $q \in \mathbb{N}_{\geq 3}$ where $q \leq r$, we define the central q -subwall of W , denoted by $W^{(q)}$, to be the q -wall obtained from W after removing its first $(r - q)/2$ layers and all occurring vertices of degree one.

Tilts. The interior of a wall W is the graph obtained from W if we remove from it all edges of $D(W)$ and all vertices of $D(W)$ that have degree two in W . Given two walls W and \tilde{W} of a graph G , we say that \tilde{W} is a tilt of W if \tilde{W} and W have identical interiors.

The following result is derived from [1]. We will use it in the improved algorithm of Theorem 2 in Section 6, in order to find a wall in a graph of bounded treewidth, given a tree decomposition of it.

Proposition 5. *There is an algorithm that, given a graph G on m edges, a graph H on h edges without isolated vertices, and a tree decomposition of G of width at most k , it outputs, if it exists, a minor of G isomorphic to H . Moreover, this algorithm runs in $2^{\mathcal{O}(k \log k)} \cdot h^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(h)} \cdot m$ -time.*

3.2 Paintings and renditions

In this subsection we present the notions of renditions and paintings, originating in the work of Robertson and Seymour [46]. The definitions presented here were introduced by Kawarabayashi et al. [33] (see also [5, 51]).

Paintings. A *closed* (resp. *open*) *disk* is a set homeomorphic to the set $\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$ (resp. $\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 < 1\}$). Let Δ be a closed disk. Given a subset X of Δ , we denote its closure by \bar{X} and its boundary by $\text{bd}(X)$. A Δ -*painting* is a pair $\Gamma = (U, N)$ where

- N is a finite set of points of Δ ,
- $N \subseteq U \subseteq \Delta$, and
- $U \setminus N$ has finitely many arcwise-connected components, called *cells*, where, for every cell c ,
 - the closure \bar{c} of c is a closed disk and
 - $|\bar{c}| \leq 3$, where $\bar{c} := \text{bd}(c) \cap N$.

We use the notation $U(\Gamma) := U$, $N(\Gamma) := N$ and denote the set of cells of Γ by $C(\Gamma)$. For convenience, we may assume that each cell of Γ is an open disk of Δ .

Notice that, given a Δ -painting Γ , the pair $(N(\Gamma), \{\bar{c} \mid c \in C(\Gamma)\})$ is a hypergraph whose hyperedges have cardinality at most three and Γ can be seen as a plane embedding of this hypergraph in Δ .

Renditions. Let G be a graph and let Ω be a cyclic permutation of a subset of $V(G)$ that we denote by $V(\Omega)$. By an Ω -*rendition* of G we mean a triple (Γ, σ, π) , where

- (a) Γ is a Δ -painting for some closed disk Δ ,
- (b) $\pi : N(\Gamma) \rightarrow V(G)$ is an injection, and
- (c) σ assigns to each cell $c \in C(\Gamma)$ a subgraph $\sigma(c)$ of G , such that
 - (1) $G = \bigcup_{c \in C(\Gamma)} \sigma(c)$,
 - (2) for distinct $c, c' \in C(\Gamma)$, $\sigma(c)$ and $\sigma(c')$ are edge-disjoint,
 - (3) for every cell $c \in C(\Gamma)$, $\pi(\bar{c}) \subseteq V(\sigma(c))$,
 - (4) for every cell $c \in C(\Gamma)$, $V(\sigma(c)) \cap \bigcup_{c' \in C(\Gamma) \setminus \{c\}} V(\sigma(c')) \subseteq \pi(\bar{c})$, and
 - (5) $\pi(N(\Gamma) \cap \text{bd}(\Delta)) = V(\Omega)$, such that the points in $N(\Gamma) \cap \text{bd}(\Delta)$ appear in $\text{bd}(\Delta)$ in the same ordering as their images, via π , in Ω .

3.3 Flatness pairs

In this subsection we define the notion of a flat wall. The definitions given in this subsection are originating in [51]. We refer the reader to that paper for a more detailed exposition of these definitions and the reasons for which we introduced them. We use the more accurate framework of [51] concerning flat walls, instead of that of [33], in order to be able to use tools that are developed in [51] and [52] and will be useful in future applications as well.

Flat walls. Let G be a graph and let W be an r -wall of G , for some odd integer $r \geq 3$. We say that a pair $(P, C) \subseteq D(W) \times D(W)$ is a *choice of pegs and corners for W* if W is the subdivision of an elementary r -wall \bar{W} where P and C are the pegs and the corners of \bar{W} , respectively (clearly, $C \subseteq P$). To get more intuition, notice that a wall W can occur in several ways from the elementary wall \bar{W} , depending on the way the vertices in the perimeter of \bar{W} are subdivided. Each of them gives a different selection (P, C) of pegs and corners of W .

We say that W is a *flat r -wall* of G if there is a separation (X, Y) of G and a choice (P, C) of pegs and corners for W such that:

- $V(W) \subseteq Y$,
- $P \subseteq X \cap Y \subseteq V(D(W))$, and
- if Ω is the cyclic ordering of the vertices $X \cap Y$ as they appear in $D(W)$, then there exists an Ω -rendition (Γ, σ, π) of $G[Y]$.

We say that W is a *flat wall* of G if it is a flat r -wall for some odd integer $r \geq 3$.

Flatness pairs. Given the above, we say that the choice of the 7-tuple $\mathfrak{R} = (X, Y, P, C, \Gamma, \sigma, \pi)$ certifies that W is a flat wall of G . We call the pair (W, \mathfrak{R}) a *flatness pair* of G and define the *height* of the pair (W, \mathfrak{R}) to be the height of W . We use the term *cell of \mathfrak{R}* in order to refer to the cells of Γ .

We call the graph $G[Y]$ the \mathfrak{R} -compass of W in G , denoted by $\text{compass}_{\mathfrak{R}}(W)$. We can assume that $\text{compass}_{\mathfrak{R}}(W)$ is connected, updating \mathfrak{R} by removing from Y the vertices of all the connected components of $\text{compass}_{\mathfrak{R}}(W)$ except of the one that contains W and including them in X (Γ, σ, π can also be easily modified according to the removal of the aforementioned vertices from Y). We define the *flaps* of the wall W in \mathfrak{R} as $\text{flaps}_{\mathfrak{R}}(W) := \{\sigma(c) \mid c \in C(\Gamma)\}$. Given a flap $F \in \text{flaps}_{\mathfrak{R}}(W)$, we define its *base* as $\partial F := V(F) \cap \pi(N(\Gamma))$. A cell c of \mathfrak{R} is *untidy* if $\pi(\tilde{c})$ contains a vertex x of W such that two of the edges of W that are incident to x are edges of $\sigma(c)$. Notice that if c is untidy then $|\tilde{c}| = 3$. A cell c of \mathfrak{R} is *tidy* if it is not untidy.

Cell classification. Given a cycle C of $\text{compass}_{\mathfrak{R}}(W)$, we say that C is \mathfrak{R} -normal if it is not a subgraph of a flap $F \in \text{flaps}_{\mathfrak{R}}(W)$. Given an \mathfrak{R} -normal cycle C of $\text{compass}_{\mathfrak{R}}(W)$, we call a cell c of \mathfrak{R} *C -perimetric* if $\sigma(c)$ contains some edge of C . Notice that if c is C -perimetric, then $\pi(\tilde{c})$ contains two points $p, q \in N(\Gamma)$ such that $\pi(p)$ and $\pi(q)$ are vertices of C where one, say P_c^{in} , of the two $(\pi(p), \pi(q))$ -subpaths of C is a subgraph of $\sigma(c)$ and the other, denoted by P_c^{out} , $(\pi(p), \pi(q))$ -subpath contains at most one internal vertex of $\sigma(c)$, which should be the (unique) vertex z in

$\partial\sigma(c) \setminus \{\pi(p), \pi(q)\}$. We pick a (p, q) -arc A_c in $\hat{c} := c \cup \tilde{c}$ such that $\pi^{-1}(z) \in A_c$ if and only if P_c^{in} contains the vertex z as an internal vertex.

We consider the circle $K_C = \mathbf{U}\{A_c \mid c \text{ is a } C\text{-perimetric cell of } \mathfrak{R}\}$ and we denote by Δ_C the closed disk bounded by K_C that is contained in Δ . A cell c of \mathfrak{R} is called *C-internal* if $c \subseteq \Delta_C$ and is called *C-external* if $\Delta_C \cap c = \emptyset$. Notice that the cells of \mathfrak{R} are partitioned into *C-internal*, *C-perimetric*, and *C-external* cells.

Let c be a tidy *C-perimetric* cell of \mathfrak{R} where $|\tilde{c}| = 3$. Notice that $c \setminus A_c$ has two arcwise-connected components and one of them is an open disk D_c that is a subset of Δ_C . If the closure \overline{D}_c of D_c contains only two points of \tilde{c} then we call the cell c *C-marginal*.

Influence. For every \mathfrak{R} -normal cycle C of $\text{compass}_{\mathfrak{R}}(W)$ we define the set

$$\text{influence}_{\mathfrak{R}}(C) = \{\sigma(c) \mid c \text{ is a cell of } \mathfrak{R} \text{ that is not } C\text{-external}\}.$$

A wall W' of $\text{compass}_{\mathfrak{R}}(W)$ is *\mathfrak{R} -normal* if $D(W')$ is \mathfrak{R} -normal. Notice that every wall of W (and hence every subwall of W) is an \mathfrak{R} -normal wall of $\text{compass}_{\mathfrak{R}}(W)$. We denote by $\mathcal{S}_{\mathfrak{R}}(W)$ the set of all \mathfrak{R} -normal walls of $\text{compass}_{\mathfrak{R}}(W)$. Given a wall $W' \in \mathcal{S}_{\mathfrak{R}}(W)$ and a cell c of \mathfrak{R} , we say that c is *W'-perimetric/internal/external/marginal* if c is $D(W')$ -perimetric/internal/external/marginal, respectively. We also use $K_{W'}$, $\Delta_{W'}$, $\text{influence}_{\mathfrak{R}}(W')$ as shortcuts for $K_{D(W')}$, $\Delta_{D(W')}$, $\text{influence}_{\mathfrak{R}}(D(W'))$, respectively.

Regular flatness pairs. We call a flatness pair (W, \mathfrak{R}) of a graph G *regular* if none of its cells is *W-external*, *W-marginal*, or *untidy*.

Tilts of flatness pairs. Let (W, \mathfrak{R}) and $(\tilde{W}', \tilde{\mathfrak{R}}')$ be two flatness pairs of a graph G and let $W' \in \mathcal{S}_{\mathfrak{R}}(W)$. We assume that $\mathfrak{R} = (X, Y, P, C, \Gamma, \sigma, \pi)$ and $\tilde{\mathfrak{R}}' = (X', Y', P', C', \Gamma', \sigma', \pi')$. We say that $(\tilde{W}', \tilde{\mathfrak{R}}')$ is a *W'-tilt* of (W, \mathfrak{R}) if

- $\tilde{\mathfrak{R}}'$ does not have \tilde{W}' -external cells,
- \tilde{W}' is a tilt of W' ,
- the set of \tilde{W}' -internal cells of $\tilde{\mathfrak{R}}'$ is the same as the set of W' -internal cells of \mathfrak{R} and their images via σ' and σ are also the same,
- $\text{compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')$ is a subgraph of $\mathbf{U}\text{influence}_{\mathfrak{R}}(W')$, and
- if c is a cell in $C(\Gamma') \setminus C(\Gamma)$, then $|\tilde{c}| \leq 2$.

The next observation follows from the third item above and the fact that the cells corresponding to flaps containing a central vertex of W' are all internal (recall that the height of a wall is always at least three).

Observation 6. Let (W, \mathfrak{R}) be a flatness pair of a graph G and $W' \in \mathcal{S}_{\mathfrak{R}}(W)$. For every W' -tilt $(\tilde{W}', \tilde{\mathfrak{R}}')$ of (W, \mathfrak{R}) , the central vertices of W' belong to the vertex set of $\text{compass}_{\tilde{\mathfrak{R}}'}(\tilde{W}')$.

Also, given a regular flatness pair (W, \mathfrak{R}) of a graph G and a $W' \in \mathcal{S}_{\mathfrak{R}}(W)$, for every W' -tilt $(\tilde{W}', \tilde{\mathfrak{R}}')$ of (W, \mathfrak{R}) , by definition none of its cells is \tilde{W}' -external, \tilde{W}' -marginal, or untidy – thus, $(\tilde{W}', \tilde{\mathfrak{R}}')$ is regular. Therefore, regularity of a flatness pair is a property that its tilts “inherit”.

Observation 7. If (W, \mathfrak{R}) is a regular flatness pair, then for every $W' \in \mathcal{S}_{\mathfrak{R}}(W)$, every W' -tilt of (W, \mathfrak{R}) is also regular.

We next present one of the main results of [51].

Proposition 8. *There exists an algorithm that, given a graph G , a flatness pair (W, \mathfrak{R}) of G , and a wall $W' \in \mathcal{S}_{\mathfrak{R}}(W)$, outputs a W' -tilt of (W, \mathfrak{R}) in $\mathcal{O}(n + m)$ -time.*

We present here the Flat Wall Theorem and, in particular, the version proved by Kawarabayashi et al. [33, Theorem 1.5]. This result will be used in the proof of correctness of the algorithm of Theorem 2.

Proposition 9. *There are two functions $f_1 : \mathbb{N} \rightarrow \mathbb{N}$ and $f_2 : \mathbb{N} \rightarrow \mathbb{N}$, where the images of f_1 are odd numbers, such that if r is an odd integer in $\mathbb{N}_{\geq 3}$, $t \in \mathbb{N}_{\geq 1}$, G is a graph that does not contain K_t as a minor, and W is an $f_1(t) \cdot r$ -wall of G , then there is a set $A \subseteq V(G)$ where $|A| \leq f_2(t)$ and a flatness pair $(\tilde{W}', \tilde{\mathfrak{R}}')$ of $G \setminus A$ of height r . Moreover $f_1(t) = \mathcal{O}(t^{26})$ and $f_2(t) = \mathcal{O}(t^{24})$.*

We conclude this subsection with the following result from [51] that allows us to find a regular flatness pair in a minor-free graph of “big enough” treewidth.

Proposition 10. *There is a function $f_3 : \mathbb{N} \rightarrow \mathbb{N}$ and an algorithm that receives as input a graph G , an odd integer $r \geq 3$, and a $t \in \mathbb{N}_{\geq 1}$, and outputs, in time $2^{\mathcal{O}_t(r^2)} \cdot n$, one of the following:*

- a report that K_t is a minor of G ,
- a tree decomposition of G of width at most $f_3(t) \cdot r$, or
- a set $A \subseteq V(G)$ with $|A| \leq f_2(t)$ and a regular flatness pair (W, \mathfrak{R}) of $G \setminus A$ of height r . (Here $f_2(t)$ is the function of Proposition 9.)

Moreover, $f_3(t) = 2^{\mathcal{O}(t^2 \log t)}$.

We note that the result of [51] also returns a tree decomposition of the flatness pair. However, this additional output is not needed in the algorithms of this paper.

3.4 Homogeneous walls

We first present some definitions on boundaried graphs and folios that will be used to define the notion of homogeneous walls. Following this, we present some results concerning homogeneous walls that are key ingredients in the application of the irrelevant vertex technique in our proofs.

Boundaried graphs. Let $t \in \mathbb{N}$. A t -boundaried graph is a triple $\mathbf{G} = (G, B, \rho)$ where G is a graph, $B \subseteq V(G)$, $|B| = t$, and $\rho : B \rightarrow [t]$ is a bijection. We say that $\mathbf{G}_1 = (G_1, B_1, \rho_1)$ and $\mathbf{G}_2 = (G_2, B_2, \rho_2)$ are *isomorphic* if there is an isomorphism from G_1 to G_2 that extends the bijection $\rho_2^{-1} \circ \rho_1$. The triple (G, B, ρ) is a *boundaried graph* if it is a t -boundaried graph for some $t \in \mathbb{N}$. As in [46] (see also [5]), we define the *detail* of a boundaried graph $\mathbf{G} = (G, B, \rho)$ as $\text{detail}(\mathbf{G}) := \max\{|E(G)|, |V(G) \setminus B|\}$. We denote by $\mathcal{B}^{(t)}$ the set of all (pairwise non-isomorphic) t -boundaried graphs and by $\mathcal{B}_\ell^{(t)}$ the set of all (pairwise non-isomorphic) t -boundaried graphs with detail at most ℓ . We also set $\mathcal{B} = \bigcup_{t \in \mathbb{N}} \mathcal{B}^{(t)}$.

We define the *treewidth* of a boundaried graph $\mathbf{G} = (G, B, \rho)$, denoted by $\text{tw}(\mathbf{G})$, as the minimum width of a tree decomposition (T, χ) of G for which there is some $u \in V(T)$ such that $B \subseteq \chi(u)$. Notice that the treewidth of a t -boundaried graph is always lower-bounded by $t - 1$.

Folios. We say that (M, T) is a *tm-pair* if M is a graph, $T \subseteq V(M)$, and all vertices in $V(M) \setminus T$ have degree two. We denote by $\text{diss}(M, T)$ the graph obtained from M by dissolving all vertices in $V(M) \setminus T$. A *tm-pair* of a graph G is a *tm-pair* (M, T) where M is a subgraph of G . We call the vertices in T *branch* vertices of (M, T) . We need to deal with topological minors for the notion of homogeneity defined below, on which the statement of [5, Theorem 5.2] relies. If $\mathbf{M} = (M, B, \rho) \in \mathcal{B}$ and $T \subseteq V(M)$ with $B \subseteq T$, we call (\mathbf{M}, T) a *btm-pair* and we define $\text{diss}(\mathbf{M}, T) = (\text{diss}(M, T), B, \rho)$. Note that we do not permit dissolution of boundary vertices, as we consider all of them to be branch vertices. If $\mathbf{G} = (G, B, \rho)$ is a boundaried graph and (M, T) is a *tm-pair* of G where $B \subseteq T$, then we say that (\mathbf{M}, T) , where $\mathbf{M} = (M, B, \rho)$, is a *btm-pair* of $\mathbf{G} = (G, B, \rho)$. Let $\mathbf{G}_1, \mathbf{G}_2$ be two boundaried graphs. We say that \mathbf{G}_1 is a *topological minor* of \mathbf{G}_2 , denoted by $\mathbf{G}_1 \preceq_{\text{tm}} \mathbf{G}_2$, if \mathbf{G}_2 has a *btm-pair* (\mathbf{M}, T) such that $\text{diss}(\mathbf{M}, T)$ is isomorphic to \mathbf{G}_1 . Given a $\mathbf{G} \in \mathcal{B}$ and a positive integer ℓ , we define the ℓ -folio of \mathbf{G} as

$$\ell\text{-folio}(\mathbf{G}) = \{\mathbf{G}' \in \mathcal{B} \mid \mathbf{G}' \preceq_{\text{tm}} \mathbf{G} \text{ and } \mathbf{G}' \text{ has detail at most } \ell\}.$$

The number of distinct ℓ -folios of t -boundaried graphs is upper-bounded in the following result, proved first in [6] and used also in [5].

Proposition 11. *There exists a function $f_4 : \mathbb{N}^2 \rightarrow \mathbb{N}$ such that for every $t, \ell \in \mathbb{N}$, $|\{\ell\text{-folio}(\mathbf{G}) \mid \mathbf{G} \in \mathcal{B}_\ell^{(t)}\}| \leq f_4(t, \ell)$. Moreover, $f_4(t, \ell) = 2^{2^{\mathcal{O}((t+\ell) \cdot \log(t+\ell))}}$.*

Augmented flaps. Let G be a graph, A be a subset of $V(G)$ of size a , and (W, \mathfrak{R}) be a flatness pair of $G \setminus A$. For each flap $F \in \text{flaps}_{\mathfrak{R}}(W)$ we consider a labeling $\ell_F : \partial F \rightarrow \{1, 2, 3\}$ such that the set of labels assigned by ℓ_F to ∂F is one of $\{1\}$, $\{1, 2\}$, $\{1, 2, 3\}$. Also, let $\tilde{a} \in [a]$. For every set $\tilde{A} \in \binom{A}{\tilde{a}}$, we consider a bijection $\rho_{\tilde{A}} : \tilde{A} \rightarrow [\tilde{a}]$. The labelings in $\mathcal{L} = \{\ell_F \mid F \in \text{flaps}_{\mathfrak{R}}(W)\}$ and the labelings in $\{\rho_{\tilde{A}} \mid \tilde{A} \in \binom{A}{\tilde{a}}\}$ will be useful for defining a set of boundaried graphs that we will call augmented flaps. We first need some more definitions.

Given a flap $F \in \text{flaps}_{\mathfrak{R}}(W)$, we define an ordering $\Omega(F) = (x_1, \dots, x_q)$, with $q \leq 3$, of the vertices of ∂F so that

- (x_1, \dots, x_q) is a counter-clockwise cyclic ordering of the vertices of ∂F as they appear in the corresponding cell of $C(\Gamma)$. Notice that this cyclic ordering is significant only when $|\partial F| = 3$,

in the sense that (x_1, x_2, x_3) remains invariant under shifting, i.e., (x_1, x_2, x_3) is the same as (x_2, x_3, x_1) but not under inversion, i.e., (x_1, x_2, x_3) is not the same as (x_3, x_2, x_1) , and

- for $i \in [q]$, $\ell_F(x_i) = i$.

Notice that the second condition is necessary for completing the definition of the ordering $\Omega(F)$, and this is the reason why we set up the labelings in \mathcal{L} .

For each set $\tilde{A} \in \binom{A}{\tilde{a}}$ and each $F \in \text{flaps}_{\mathfrak{R}}(W)$ with $t_F := |\partial F|$, we fix $\rho_F : \partial F \rightarrow [\tilde{a} + 1, \tilde{a} + t_F]$ such that $(\rho_F^{-1}(\tilde{a} + 1), \dots, \rho_F^{-1}(\tilde{a} + t_F)) = \Omega(F)$. Also, we define the boundaried graph

$$\mathbf{F}^{\tilde{A}} := (G[\tilde{A} \cup F], \tilde{A} \cup \partial F, \rho_{\tilde{A}} \cup \rho_F)$$

and we denote by $F^{\tilde{A}}$ the underlying graph of $\mathbf{F}^{\tilde{A}}$. We call $\mathbf{F}^{\tilde{A}}$ an \tilde{A} -augmented flap of the flatness pair (W, \mathfrak{R}) of $G \setminus A$ in G .

Palettes and homogeneity. For each \mathfrak{R} -normal cycle C of $\text{compass}_{\mathfrak{R}}(W)$ and each set $\tilde{A} \subseteq A$, we define (\tilde{A}, ℓ) -palette(C) = $\{\ell\text{-folio}(\mathbf{F}^{\tilde{A}}) \mid F \in \text{influence}_{\mathfrak{R}}(C)\}$. Given a set $\tilde{A} \subseteq A$, we say that the flatness pair (W, \mathfrak{R}) of $G \setminus A$ is ℓ -homogeneous with respect to \tilde{A} if every internal brick of W has the same (\tilde{A}, ℓ) -palette (seen as a cycle of $\text{compass}_{\mathfrak{R}}(W)$). Also, given a collection $\mathcal{S} \subseteq 2^A$, we say that the flatness pair (W, \mathfrak{R}) of $G \setminus A$ is ℓ -homogeneous with respect to \mathcal{S} if it is ℓ -homogeneous with respect to every $\tilde{A} \in \mathcal{S}$.

The following observation is a consequence of the fact that, given a wall W and a subwall W' of W , every internal brick of a tilt W'' of W' is also an internal brick of W .

Observation 12. Let $\ell \in \mathbb{N}$, G be a graph, $A \subseteq V(G)$, $\mathcal{S} \subseteq 2^A$, and (W, \mathfrak{R}) be a flatness pair of $G \setminus A$. If (W, \mathfrak{R}) is ℓ -homogeneous with respect to \mathcal{S} , then for every subwall W' of W , every W' -tilt of (W, \mathfrak{R}) is also ℓ -homogeneous with respect to \mathcal{S} .

Let $a, \tilde{a}, \ell \in \mathbb{N}$, where $\tilde{a} \leq a$. Also, let G be a graph, A be a subset of $V(G)$ of size at most a , and (W, \mathfrak{R}) be a flatness pair of $G \setminus A$. For every flap $F \in \text{flaps}_{\mathfrak{R}}(W)$, we define the function $\text{var}_F^{(A, \tilde{a}, \ell)} : \binom{A}{\leq \tilde{a}} \rightarrow \{\ell\text{-folio}(\mathbf{G}) \mid \mathbf{G} \in \bigcup_{i \in [\tilde{a}+3]} \mathcal{B}^{(i)}\}$ that maps each set $\tilde{A} \in \binom{A}{\leq \tilde{a}}$ to the set $\ell\text{-folio}(\mathbf{F}^{\tilde{A}})$.

We also use the following result that follows from [Proposition 11](#) and the fact that $|\binom{A}{\leq \tilde{a}}| = \mathcal{O}(|A|^{\tilde{a}})$ (see also [\[52\]](#)).

Lemma 13. *There exists a function $f_5 : \mathbb{N}^3 \rightarrow \mathbb{N}$ such that if $a, \tilde{a}, \ell \in \mathbb{N}$, where $\tilde{a} \leq a$, G is a graph, A is a subset of $V(G)$ of size at most a , and (W, \mathfrak{R}) is a flatness pair of $G \setminus A$, then*

$$|\{\text{var}_F^{(A, \tilde{a}, \ell)} \mid F \in \text{flaps}_{\mathfrak{R}}(W)\}| \leq f_5(a, \tilde{a}, \ell).$$

Moreover, $f_5(a, \tilde{a}, \ell) = 2^{a \cdot 2^{\mathcal{O}((\tilde{a} + \ell) \cdot \log(\tilde{a} + \ell))}}$.

[Lemma 13](#) allows us to define an injective function $\sigma : \{\text{var}_F^{(A, \tilde{a}, \ell)} \mid F \in \text{flaps}_{\mathfrak{R}}(W)\} \rightarrow [f_5(a, \tilde{a}, \ell)]$ that maps each function in $\{\text{var}_F^{(A, \tilde{a}, \ell)} \mid F \in \text{flaps}_{\mathfrak{R}}(W)\}$ to an integer in $[f_5(a, \tilde{a}, \ell)]$. Using σ , we define a function $\zeta_{A, \tilde{a}, \ell} : \text{flaps}_{\mathfrak{R}}(W) \rightarrow [f_5(a, \tilde{a}, \ell)]$, that maps each flap $F \in \text{flaps}_{\mathfrak{R}}(W)$ to the integer $\sigma(\text{var}_F^{(A, \tilde{a}, \ell)})$. In [\[51\]](#), given a $w \in \mathbb{N}$, the notion of homogeneity is defined with respect to a flap-coloring ζ of (W, \mathfrak{R}) with w colors, that is a function from $\text{flaps}_{\mathfrak{R}}(W)$ to $[w]$. This function gives rise

to the ζ -palette of each \mathfrak{R} -normal cycle of $\text{compass}_{\mathfrak{R}}(W)$ which, in turn, is used to define the notion of a ζ -homogeneous flatness pair. Hence, using the terminology of [51], $\zeta_{A,\tilde{a},\ell}$ is a flap-coloring of (W, \mathfrak{R}) with $f_5(a, \tilde{a}, \ell)$ colors, that “colors” each flap $F \in \text{flaps}_{\mathfrak{R}}(W)$ by mapping it to the integer $\sigma(\text{var}_F^{(A,\tilde{a},\ell)})$, and the notion of ℓ -homogeneity with respect to $\binom{A}{\leq \tilde{a}}$ defined here can be alternatively interpreted as $\zeta_{A,\tilde{a},\ell}$ -homogeneity. The following result, which is the application of a result of Sau et al. [51, Lemma 13] for the flap-coloring $\zeta_{A,\tilde{a},\ell}$, provides an algorithm that, given a flatness pair of “big enough” height, outputs a homogeneous flatness pair.

Proposition 14. *There is a function $f_6 : \mathbb{N}^4 \rightarrow \mathbb{N}$, whose images are odd integers, and an algorithm that receives as input an odd integer $r \geq 3$, $\tilde{a}, a, \ell \in \mathbb{N}$, where $\tilde{a} \leq a$, a graph G , a set $A \subseteq V(G)$ of size at most a , and a flatness pair (W, \mathfrak{R}) of $G \setminus A$ of height $f_6(r, a, z, \ell)$, and outputs a flatness pair $(\hat{W}, \hat{\mathfrak{R}})$ of $G \setminus A$ of height r that is ℓ -homogeneous with respect to $\binom{A}{\leq \tilde{a}}$ and is a W' -tilt of (W, \mathfrak{R}) for some subwall W' of W . Moreover, $f_6(r, a, \tilde{a}, \ell) = \mathcal{O}(r^{f_5(a,\tilde{a},\ell)})$ and the algorithm runs in time $2^{\mathcal{O}(f_5(a,\tilde{a},\ell) \cdot r \log r)} \cdot (n + m)$.*

The price of homogeneity. As Proposition 14 indicates, finding a homogeneous flat wall inside a flat wall has a price, corresponding to the function $f_6(r, a, \tilde{a}, \ell)$ of the required height of the given flat wall. The “polynomial gap” between the height of the given flatness pair (W, \mathfrak{R}) of $G \setminus A$ and the homogenous flat wall that is returned is determined by the function f_5 of Observation 12, that bounds the number of different folios that can be rooted through the augmented flaps of (W, \mathfrak{R}) , for each possible augmentation of each flap with a subset of A of size at most \tilde{a} . In this paper, we will use Proposition 14 in order to compute a flat wall that is homogeneous with respect to 2^A , that is, for $\tilde{a} = a$. We set $c_{a,\ell} = f_5(a, a, \ell)$ and we point out that, in general, it follows that $c_{a,\ell} = 2^{2^{\mathcal{O}((a+\ell) \cdot \log(a+\ell))}}$. However, by using the notion of representatives instead of folios as in [5], we can obtain a smaller bound of $c_{a,\ell} = 2^{\mathcal{O}((a+\ell) \cdot \log(a+\ell))}$. We call $c_{a_{\mathcal{F}}, \ell_{\mathcal{F}}}$ the *palette-variety* of \mathcal{F} .

4 Auxiliary algorithmic and combinatorial results

In this section we provide some algorithmic and combinatorial results that will support the main algorithms of this paper. In Subsection 4.1 we provide an algorithm that finds an irrelevant vertex inside a “large enough” homogeneous flat wall, while, in Subsection 4.2, we define canonical partitions of walls and we present some combinatorial results that allow our algorithms to branch.

4.1 Finding an irrelevant vertex

The *irrelevant vertex technique* was introduced in [46] for providing an FPT-algorithm for the DISJOINT PATHS problem. Moreover, this technique has appeared to be quite versatile and is now a standard tool of parameterized algorithm design (see e.g., [15, 54]). The applicability of this technique for \mathcal{F} -M-DELETION is materialized in this section by the algorithm of Lemma 16.

For the proof of Lemma 16 we need the next combinatorial result, Proposition 15, whose proof is presented in [52]. Proposition 15 intuitively states that, given a graph G and a homogeneous flatness pair (W, \mathfrak{R}) of G of “big enough” height, it holds that for every $W^{(q)}$ -tilt $(\hat{W}, \hat{\mathfrak{R}})$ of (W, \mathfrak{R}) , $\text{compass}_{\mathfrak{R}}(\hat{W})$ can be “safely” removed from the input graph G , in the sense that (G, k) and $(G \setminus V(\text{compass}_{\mathfrak{R}}(\hat{W})), k)$ are equivalent instances of \mathcal{F} -M-DELETION. In [52] we insisted on a proof of

Proposition 15 that requires homogeneity with respect to 2^A (that is, with respect to every possible subset of A) in order to find an irrelevant wall, no matter the choice of the hitting set. This is an enhancement of the result of [5, Theorem 5.2], which allows to reroute minors outside a part of the wall that is homogeneous with respect to a *particular* apex set. When aiming to detect a wall that is irrelevant for \mathcal{F} -M-DELETION, we do not know a priori which is the hitting set, and therefore we need to ask, firstly, for a wall that is irrelevant for every choice of a hitting set S and, secondly, for homogeneity that captures all possible remaining (after the deletion of S) apex sets of the flat wall in order to apply [5, Theorem 5.2] for such an apex set. For these reasons, this enhancement of [5, Theorem 5.2] is essential for our case.

The running time of the next result depends on the function f_{ul} coming from the Unique Linkage Theorem from [34] (see also [48, 49]). Recall that $\ell_{\mathcal{F}} = \max\{\text{detail}(H) \mid H \in \mathcal{F}\}$.

Proposition 15. *There exist two functions $f_7 : \mathbb{N}^4 \rightarrow \mathbb{N}$ and $f_8 : \mathbb{N}^2 \rightarrow \mathbb{N}$, where the images of f_7 are odd numbers, such that for every $a, k \in \mathbb{N}$, every odd $q \in \mathbb{N}_{\geq 3}$, and every graph G , if A is a subset of $V(G)$ of size at most a and (W, \mathfrak{R}) is a regular flatness pair of $G \setminus A$ of height at least $f_7(a, \ell_{\mathcal{F}}, q, k)$ that is $f_8(a, \ell_{\mathcal{F}})$ -homogeneous with respect to 2^A , then for every $W^{(q)}$ -tilt $(\hat{W}, \hat{\mathfrak{R}})$ of (W, \mathfrak{R}) , it holds that (G, k) and $(G \setminus V(\text{compass}_{\hat{\mathfrak{R}}}(\hat{W})), k)$ are equivalent instances of \mathcal{F} -M-DELETION. Moreover, $f_7(a, \ell_{\mathcal{F}}, q, k) = \mathcal{O}(k \cdot (f_{ul}(16a + 12\ell_{\mathcal{F}}))^3 + q)$, where f_{ul} is the function of the Unique Linkage Theorem, and $f_8(a, \ell_{\mathcal{F}}) = a + \ell_{\mathcal{F}} + 3$.*

By applying **Proposition 8** on top of **Proposition 15**, in order to find a tilt that is guaranteed to be irrelevant by **Proposition 15**, we directly get the following algorithm, which outputs a flatness pair $(\hat{W}, \hat{\mathfrak{R}})$ of an input graph G such that (G, k) and $(G \setminus V(\text{compass}_{\hat{\mathfrak{R}}}(\hat{W})), k)$ are equivalent instances of \mathcal{F} -M-DELETION. In fact, in the rest of the paper, we use a slightly weaker version of **Lemma 16**, referred as **Corollary 17**, that outputs just an irrelevant vertex. Here, we prove this more general result for future use.

Lemma 16. *There exists an algorithm with the following specifications:*

Find-Irrelevant-Wall $(k, q, a, G, A, W, \mathfrak{R})$

Input: Three integers $k, q, a \in \mathbb{N}$, with odd $q \geq 3$, a graph G , a set $A \subseteq V(G)$ of size at most a , and a regular flatness pair (W, \mathfrak{R}) of $G \setminus A$ of height at least $f_7(a, \ell_{\mathcal{F}}, q, k)$ that is $f_8(a, \ell_{\mathcal{F}})$ -homogeneous with respect to 2^A .

Output: A flatness pair $(\hat{W}, \hat{\mathfrak{R}})$ of $G \setminus A$ that is a $W^{(q)}$ -tilt of (W, \mathfrak{R}) and such that (G, k) and $(G \setminus V(\text{compass}_{\hat{\mathfrak{R}}}(\hat{W})), k)$ are equivalent instances of \mathcal{F} -M-DELETION.

Moreover, this algorithm runs in $\mathcal{O}(n + m)$ -time.

Notice that **Lemma 16** together with **Observation 6** imply **Corollary 17** if we set $q = 3$ and output a central vertex of the obtained 3-wall.

Corollary 17. *There exists an algorithm with the following specifications:*

Find-Irrelevant-Vertex $(k, a, G, A, W, \mathfrak{R})$

Input: Two integers $k, a \in \mathbb{N}$, a graph G , a set $A \subseteq V(G)$ of size at most a , and a regular flatness pair (W, \mathfrak{R}) of $G \setminus A$ of height at least $f_7(a, \ell_{\mathcal{F}}, 3, k)$ that is $f_8(a, \ell_{\mathcal{F}})$ -homogeneous with respect to 2^A .

Output: A vertex $v \in V(G)$ such that (G, k) and $(G \setminus v, k)$ are equivalent instances of \mathcal{F} -M-DELETION.

Moreover, this algorithm runs in $\mathcal{O}(n + m)$ -time.

4.2 Combinatorial results for branching

In this subsection, we present the notion of a canonical partition and provide two combinatorial results that will justify a branching step of our algorithm and, if such a step cannot be applied, the existence of a wall that will allow the application of the irrelevant vertex technique. Canonical partitions were introduced in [52].

Canonical partitions. Let $r \geq 3$ be an odd integer. Let W be an r -wall and let P_1, \dots, P_r (resp. L_1, \dots, L_r) be its vertical (resp. horizontal) paths. For every even (resp. odd) $i \in [2, r - 1]$ and every $j \in [2, r - 1]$, we define $A^{(i,j)}$ to be the subpath of P_i that starts from a vertex of $P_i \cap L_j$ and finishes at a neighbor of a vertex in L_{j+1} (resp. L_{j-1}), such that $P_i \cap L_j \subseteq A^{(i,j)}$ and $A^{(i,j)}$ does not intersect L_{j+1} (resp. L_{j-1}). Similarly, for every $i, j \in [2, r - 1]$, we define $B^{(i,j)}$ to be the subpath of L_j that starts from a vertex of $P_i \cap L_j$ and finishes at a neighbor of a vertex in P_{i-1} , such that $P_i \cap L_j \subseteq A^{(i,j)}$ and $A^{(i,j)}$ does not intersect P_{i-1} .

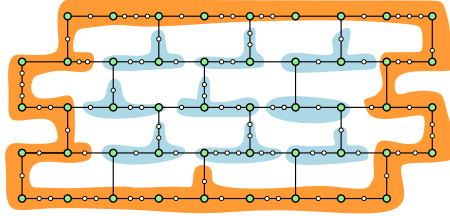


Figure 2: A 5-wall and its canonical partition \mathcal{Q} . The orange bag is the external bag Q_{ext} .

For every $i, j \in [2, r - 1]$, we denote by $Q^{(i,j)}$ the graph $A^{(i,j)} \cup B^{(i,j)}$ and Q_{ext} to be the graph $W \setminus \bigcup_{i,j \in [2, r - 1]} Q_{i,j}$. Now consider the collection $\mathcal{Q} = \{Q_{\text{ext}}\} \cup \{Q_{i,j} \mid i, j \in [2, r - 1]\}$ and observe that the graphs in \mathcal{Q} are connected subgraphs of W and their vertex sets form a partition of $V(W)$. We call \mathcal{Q} the *canonical partition* of W . Also, we call every $Q_{i,j}, i, j \in [2, r - 1]$ an *internal bag* of \mathcal{Q} , while we refer to Q_{ext} as the *external bag* of \mathcal{Q} . See Figure 2 for an illustration of the notions defined above. For every $i \in [(r - 1)/2]$, we say that a set $Q \in \mathcal{Q}$ is an *i -internal bag* of \mathcal{Q} if $V(Q)$ does not contain any vertex of the first i layers of W . Notice that the 1-internal bags of \mathcal{Q} are the internal bags of \mathcal{Q} .

Let (W, \mathfrak{R}) be a flatness pair of a graph G . Consider the canonical partition \mathcal{Q} of W . We enhance the graphs of \mathcal{Q} so to include in them all the vertices of G by applying the following procedure. We set $\tilde{\mathcal{Q}} := \mathcal{Q}$ and, as long as there is a vertex $x \in V(\text{compass}_{\mathfrak{R}}(W)) \setminus V(\mathbf{U}\tilde{\mathcal{Q}})$ that is adjacent to a vertex of a graph $Q \in \tilde{\mathcal{Q}}$, update $\tilde{\mathcal{Q}} := \tilde{\mathcal{Q}} \setminus \{Q\} \cup \{\tilde{Q}\}$, where $\tilde{Q} = \text{compass}_{\mathfrak{R}}(W)[\{x\} \cup V(Q)]$. Since $\text{compass}_{\mathfrak{R}}(W)$ is a connected graph, in this way we define a partition of the vertices of $\text{compass}_{\mathfrak{R}}(W)$ into subsets inducing connected graphs. We call the $\tilde{Q} \in \tilde{\mathcal{Q}}$ that contains Q_{ext} as a subgraph the *external bag* of $\tilde{\mathcal{Q}}$, and we denote it by \tilde{Q}_{ext} , while we call *internal bags* of $\tilde{\mathcal{Q}}$ all graphs in $\tilde{\mathcal{Q}} \setminus \{\tilde{Q}_{\text{ext}}\}$. Moreover, we enhance $\tilde{\mathcal{Q}}$ by adding all vertices of $G \setminus V(\text{compass}_{\mathfrak{R}}(W))$ in its external bag, i.e., by

updating $\tilde{Q}_{\text{ext}} := G[V(\tilde{Q}_{\text{ext}}) \cup V(G \setminus V(\text{compass}_{\mathfrak{R}}(W)))]$. We call such a partition \tilde{Q} a (W, \mathfrak{R}) -canonical partition of G . Notice that a (W, \mathfrak{R}) -canonical partition of G is not unique, since the sets in \tilde{Q} can be “expanded” arbitrarily when introducing vertex x .

Let (W, \mathfrak{R}) be a flatness pair of a graph G of height r , for some $r \geq 3$ and \tilde{Q} be a (W, \mathfrak{R}) -canonical partition of G . For every $i \in [(r-1)/2]$, we say that a set $Q \in \tilde{Q}$ is an i -internal bag of \tilde{Q} if it contains an i -internal bag of \tilde{Q} as a subgraph.

Next we identify a combinatorial structure that guarantees the existence of a set of $q = \mathcal{O}_{s_{\mathcal{F}}}(k)$ vertices that intersects every solution S of \mathcal{F} -M-DELETION with input (G, k) . This will permit branching on q simpler instances of the form $(G', k-1)$. Recall that $a_{\mathcal{F}}$ is the minimum apex number of a graph in \mathcal{F} . The following result is proved in [52].

Proposition 18. *There exist three functions $f_9, f_{10}, f_{11} : \mathbb{N}^3 \rightarrow \mathbb{N}$, such that if \mathcal{F} is a finite set of graphs, G is a graph, $k \in \mathbb{N}$, A is a subset of $V(G)$, (W, \mathfrak{R}) is a flatness pair of $G \setminus A$ of height at least $f_9(a_{\mathcal{F}}, s_{\mathcal{F}}, k)$, \tilde{Q} is a (W, \mathfrak{R}) -canonical partition of $G \setminus A$, A' is a subset of vertices of A that are adjacent, in G , to vertices of at least $f_{10}(a_{\mathcal{F}}, s_{\mathcal{F}}, k)$ $f_{11}(a_{\mathcal{F}}, s_{\mathcal{F}}, k)$ -internal bags of \tilde{Q} , and $|A'| \geq a_{\mathcal{F}}$, then for every set $S \subseteq V(G)$ of size at most k such that $G \setminus S \in \text{exc}(\mathcal{F})$ it holds that $S \cap A' \neq \emptyset$. Moreover, $f_9(a, s, k) = \mathcal{O}(2^a \cdot s^{5/2} \cdot k^{5/2})$, $f_{10}(a, s, k) = \mathcal{O}(2^a \cdot s^3 \cdot k^3)$, and $f_{11}(a, s, k) = \mathcal{O}((a^2 + k) \cdot s)$, where $a = a_{\mathcal{F}}$ and $s = s_{\mathcal{F}}$.*

The next result is also proved in [52] and intuitively states that, given a flatness pair (W, \mathfrak{R}) of “big enough” height and a (W, \mathfrak{R}) -canonical partition \tilde{Q} of G , we can find a “packing” of subwalls of W that are inside some central part of W and that the vertex set of every internal bag of \tilde{Q} intersects the vertices of the flaps in the influence of at most one of these walls. We will use this result in the case where the set A' of Proposition 18 is “small”, i.e., there are only “few” vertices in A that have “big enough” degree with respect to the central part of the canonical partition, and therefore Proposition 18 cannot justify branching. Following the latter condition and Proposition 19, we will be able to find a flatness pair with “few” apices so as to build irrelevant vertex arguments inside its compass.

Proposition 19. *There exists a function $f_{12} : \mathbb{N}^3 \rightarrow \mathbb{N}$ such that if $p, z \in \mathbb{N}_{\geq 1}$, $x \in \mathbb{N}_{\geq 3}$ is an odd integer, G is a graph, (W, \mathfrak{R}) is a flatness pair of G of height at least $f_{12}(z, x, p)$, and \tilde{Q} is a (W, \mathfrak{R}) -canonical partition of G , then there is a collection $\mathcal{W} = \{W_1, \dots, W_z\}$ of x -subwalls of W such that*

- for every $i \in [z]$, $\mathbf{U}\text{influence}_{\mathfrak{R}}(W_i)$ is a subgraph of $\mathbf{U}\{Q \mid Q \text{ is a } p\text{-internal bag of } \tilde{Q}\}$ and
- for every $i, j \in [z]$, with $i \neq j$, there is no internal bag of \tilde{Q} that contains vertices of both $V(\mathbf{U}\text{influence}_{\mathfrak{R}}(W_i))$ and $V(\mathbf{U}\text{influence}_{\mathfrak{R}}(W_j))$.

Moreover, $f_{12}(z, x, p) = \mathcal{O}(\sqrt{z} \cdot x + p)$.

5 The general algorithm

In this section we present the general algorithm for \mathcal{F} -M-DELETION. The existence of this algorithm proves Theorem 1. In Subsection 5.1, we explain how to employ the *iterative compression technique* so as to ask for an algorithm for a new, more convenient to solve, problem and, in Subsection 5.2, we develop an algorithm for this new problem.

5.1 Iterative compression

In order to prove [Theorem 1](#), we apply the iterative compression technique (introduced in [\[45\]](#); see also [\[15\]](#)) and we give a $2^{\text{poly}(k)} \cdot n^2$ -time algorithm for the following problem.

\mathcal{F} -M-DELETION-COMPRESSION

Input: A graph G , a $k \in \mathbb{N}$, and a set S of size $k + 1$ such that $G \setminus S \in \mathbf{exc}(\mathcal{F})$.

Objective: Find, if exists, a set $S' \subseteq V(G)$ of size at most k such that $G \setminus S' \in \mathbf{exc}(\mathcal{F})$.

In other words, given an input (G, k, S) of \mathcal{F} -M-DELETION-COMPRESSION, we have at hand a graph G and a “slightly larger than k ” hitting set S , and we aim to find a hitting set of size at most k , that is a certificate that (G, k) is a yes-instance of \mathcal{F} -M-DELETION. Given this set S , we can directly assume that $G \setminus S$ does not contain a big clique as a minor and therefore we can deal with this minor-free graph, and thus, due to [Proposition 10](#), we can obtain either a tree decomposition of G of “small” width (and solve the problem using the dynamic programming algorithm of [\[5\]](#)), or a flat wall on top of which we build our branching and irrelevant vertex technique arguments. In this way, we manage to avoid the “big clique” possible output of [Proposition 10](#). However, this swifiting from \mathcal{F} -M-DELETION to \mathcal{F} -M-DELETION-COMPRESSION comes together with an extra linear factor in the running time of the algorithm, as observed in the following (see [\[15\]](#)).

Observation 20. If there is an algorithm solving \mathcal{F} -M-DELETION-COMPRESSION in $f(k) \cdot n^c$ -time, then there exists an algorithm solving \mathcal{F} -M-DELETION in $\mathcal{O}(f(k) \cdot n^{c+1})$ -time.

In [Subsection 5.2](#) we prove that \mathcal{F} -M-DELETION-COMPRESSION can be solved in $2^{\text{poly}(k)} \cdot n^2$ -time ([Lemma 21](#)). This along with [Observation 20](#) yield [Theorem 1](#).

5.2 The algorithm

In this subsection we present the algorithm solving \mathcal{F} -M-DELETION-COMPRESSION.

We set $\tilde{c}_{a,\ell} := f_4(a, f_8(a, \ell)) = 2^{2^{\mathcal{O}((a+\ell) \cdot \log(a+\ell))}}$, where f_4 is the number of different folios given in [Proposition 11](#) and f_8 is the function given in [Proposition 15](#), in order to find an irrelevant vertex.

Lemma 21. *Let \mathcal{F} be a finite collection of graphs. There is an algorithm solving \mathcal{F} -M-DELETION-COMPRESSION in $2^{\mathcal{O}_s(k^{2 \cdot (c+2)})} \cdot n^2$ -time, where $a = a_{\mathcal{F}}$, $s = s_{\mathcal{F}}$, $\ell = \ell_{\mathcal{F}}$, and $c = \tilde{c}_{a,\ell}$.*

Proof. For simplicity, in this proof, we use c instead of $\tilde{c}_{a_{\mathcal{F}}, \ell_{\mathcal{F}}}$, s instead of $s_{\mathcal{F}}$, ℓ instead of $\ell_{\mathcal{F}}$, a instead of $a_{\mathcal{F}}$, and recall that $\ell = \mathcal{O}(s^2)$ and $a \leq s$. Also, we set

$$\begin{aligned} z &= f_7(a - 1, \ell, 3, k), & d &= f_8(a, \ell), & b &= f_6(z, a, a, d) = \mathcal{O}_{s_{\mathcal{F}}}(k^c), \\ m &= f_9(a, s, k), & x &= f_{10}(a, s, k), & l &= (f_2(s) + k + 1) \cdot x, \\ p &= f_{11}(a, s, k), & h &= f_{12}(l + 1, b, p), & \text{and } r &= \text{odd}(\max\{m, h\}) = \mathcal{O}_s(k^{c+2}). \end{aligned}$$

We present the algorithm **Solve-Compression**, whose input is a quadruple (G, k', k, S) where G is a graph, k' and k are non-negative integers with $k' \leq k$, and S is a subset of $V(G)$ such that $|S| = k$ and $G \setminus S \in \mathbf{exc}(\mathcal{F})$. The algorithm returns, if it exists, a solution for \mathcal{F} -M-DELETION on (G, k') . Certainly, we may assume that $k' < k$, otherwise S is already a solution and we are done. The steps of the algorithm are the following:

Step 1. Run the algorithm of [Proposition 10](#) with input $(G \setminus S, r, s)$. Since $G \setminus S \in \text{exc}(\mathcal{F})$ and $\mathcal{F} \leq_m K_s$, the algorithm outputs, in time $2^{\mathcal{O}_s(r^2)} \cdot n = 2^{\mathcal{O}_s(k^{2 \cdot (c+2)})} \cdot n$, either a tree decomposition of $G \setminus S$ of width at most $f_3(s) \cdot r$, or a set $A \subseteq V(G)$ with $|A| \leq f_2(s)$ and a regular flatness pair (W, \mathfrak{R}) of $G \setminus A$ of height r . In the first case, we solve \mathcal{F} -M-DELETION-COMPRESSION in time $2^{\mathcal{O}_s(r \log r)} \cdot n = 2^{\mathcal{O}_s(k^{c+2} \log k)} \cdot n$ using the algorithm of [Proposition 4](#). In what follows we examine the second case, where the algorithm of [Proposition 10](#) outputs a set $A \subseteq V(G)$ with $|A| \leq f_2(s)$ and a regular flatness pair (W, \mathfrak{R}) of $G \setminus A$ of height r .

We consider a (W, \mathfrak{R}) -canonical partition $\tilde{\mathcal{Q}}$ of $G \setminus (S \cup A)$. We compute, in $\mathcal{O}(n)$ -time, the set

$$A^* = \{v \in S \cup A \mid v \text{ is adjacent, in } G, \text{ to vertices of at least } x \text{ } p\text{-internal bags of } \tilde{\mathcal{Q}}\}$$

and we proceed to the second step.

Step 2. The algorithm examines two cases depending on the size of the A^* . In the first case, the *branching case*, the outcome is a set of vertices, the set $S \cup A$, that should intersect every possible solution. In the second case, the *irrelevant vertex case*, the outcome is an irrelevant vertex.

[*Branching case*]. It holds that $|A^*| \geq a$. In this case the algorithm recursively calls **Solve-Compression** with input $(G \setminus x, k' - 1, |S \setminus x|, S \setminus x)$ for every $x \in A^*$, and if one of these new instances is a yes-instance, certified by a set \bar{S} , then returns $\bar{S} \cup \{x\}$, otherwise it returns that (G, k') is a no-instance.

[Proposition 18](#) implies that the above branching step of the algorithm is correct.

[*Irrelevant vertex case*]. It holds that $|A^*| < a$. We consider a family $\mathcal{W} = \{W_1, \dots, W_{l+1}\}$ of $l + 1$ b -subwalls of W such that for every $i \in [l + 1]$, $\mathbf{U}\text{influence}_{\mathfrak{R}}(W_i)$ is a subgraph of $\mathbf{U}\{Q \mid Q \text{ is a } p\text{-internal bag of } \tilde{\mathcal{Q}}\}$ and for every $i, j \in [l + 1]$, where $i \neq j$, there is no internal bag $Q \in \tilde{\mathcal{Q}}$ that contains vertices of both $V(\mathbf{U}\text{influence}_{\mathfrak{R}}(W_i))$ and $V(\mathbf{U}\text{influence}_{\mathfrak{R}}(W_j))$. The existence of \mathcal{W} follows from the fact that $r \geq h = f_{12}(l + 1, b, p)$ and [Proposition 19](#).

Notice that the vertices in $(S \cup A) \setminus A^*$ are adjacent, in G , to vertices of at most $x \cdot |(S \cup A) \setminus A^*| \leq x \cdot (f_2(s) + k + 1) = l$ p -internal bags of $\tilde{\mathcal{Q}}$. Hence, taking into account the aforementioned properties of the walls W_1, \dots, W_{l+1} , there exists an $i \in [l + 1]$ such that no vertex in $(S \cup A) \setminus A^*$ is adjacent to vertices of $V(\mathbf{U}\text{influence}_{\mathfrak{R}}(W_i))$. In other words, if there exists a vertex $v \in V(\mathbf{U}\text{influence}_{\mathfrak{R}}(W_i))$ that is adjacent, in G , to a vertex $u \in S \cup A$, then $u \in A^*$. The fact that $|A^*| < a$ implies that, for this i , there are less than a vertices in $S \cup A$ that are adjacent to vertices of $V(\mathbf{U}\text{influence}_{\mathfrak{R}}(W_i))$.

Since W_i is a b -subwall of W and (W, \mathfrak{R}) is a flatness pair of $G \setminus (S \cup A)$, we apply the algorithm of [Proposition 8](#), and obtain, in linear time, a flatness pair $(\tilde{W}_i, \tilde{\mathfrak{R}}_i)$ of $G \setminus (S \cup A)$ that is a W_i -tilt of (W, \mathfrak{R}) . Notice that since $(\tilde{W}_i, \tilde{\mathfrak{R}}_i)$ is a W_i -tilt of (W, \mathfrak{R}) , $\text{compass}_{\tilde{\mathfrak{R}}_i}(\tilde{W}_i)$ is a subgraph of $\mathbf{U}\text{influence}_{\mathfrak{R}}(\tilde{W}_i)$ and, due to [Observation 7](#), $(\tilde{W}_i, \tilde{\mathfrak{R}}_i)$ is regular. This implies that, if A_i is the set of vertices from $S \cup A$ that are adjacent to vertices of $\text{compass}_{\tilde{\mathfrak{R}}_i}(\tilde{W}_i)$ in G , then $A_i \subseteq A^*$ and therefore $|A_i| < a$. Notice that, by adding the vertices of $(S \cup A) \setminus A_i$ to $G \setminus (S \cup A)$, we obtain a flatness pair $(\check{W}_i, \check{\mathfrak{R}}_i)$ of $G \setminus A_i$ such that $\text{compass}_{\check{\mathfrak{R}}_i}(\check{W}_i) = \text{compass}_{\tilde{\mathfrak{R}}_i}(\tilde{W}_i)$. Applying the algorithm of [Proposition 14](#) for $(b, a, a, d, G, A_i, \check{W}_i, \check{\mathfrak{R}}_i)$, we obtain a flatness pair $(\check{W}'_i, \check{\mathfrak{R}}'_i)$ of $G \setminus A_i$ of height z that is d -homogeneous with respect to 2^{A_i} and is a \check{W}'_i -tilt of $(\check{W}_i, \check{\mathfrak{R}}_i)$ for some subwall \check{W}'_i of \check{W}_i . Due to [Observation 7](#), $(\check{W}'_i, \check{\mathfrak{R}}'_i)$ is also regular. This algorithm runs in $2^{\mathcal{O}_s(k \log k)} \cdot n$ -time.

We now apply **Find-Irrelevant-Vertex** of [Corollary 17](#) for $(k, a, G, A_i, \check{W}_i, \check{\mathfrak{R}}_i)$ and obtain a vertex v such that (G, k) and $(G \setminus v, k)$ are equivalent instances of **\mathcal{F} -M-DELETION**. According to [Corollary 17](#), this vertex can be detected in linear time, and the algorithm correctly calls recursively **Solve-Compression** with input $(G \setminus v, k', k, S)$. This completes the *irrelevant vertex case*.

Recall that $|S \cup A| \leq k + 1 + f_2(s) = \mathcal{O}_s(k)$. Therefore, if $T(n, k', k)$ is the running time of the above algorithm, then

$$T(n, k', k) \leq 2^{\mathcal{O}_s(k^2 \cdot (c+2))} \cdot n + \max\{T(n-1, k', k), \mathcal{O}_s(k) \cdot T(n, k-1, k)\}$$

that, given that $k' \leq k$, implies that $T(n, k', k) = 2^{\mathcal{O}_s(k^2 \cdot (c+2))} \cdot n^2$.

Notice now that the output of **Solve-Compression** on $(G, k, k+1, S)$ gives a solution for **\mathcal{F} -M-DELETION-COMPRESSON** on this instance. \square

6 The apex-minor free case

In this section we present an improved algorithm solving **\mathcal{F} -M-DELETION** in the case where $a_{\mathcal{F}} = 1$. The existence of this algorithm proves [Theorem 2](#). In [Subsection 6.1](#), we show that a graph that contains a flat wall that is “highly connected” to a vertex in its apex set, also contains any apex graph as a minor. In [Subsection 6.2](#), we provide an algorithm that will allow us to detect a wall inside a graph G in linear time. In [Subsection 6.3](#), we provide the improved algorithm that solves **\mathcal{F} -M-DELETION** in the case where $a_{\mathcal{F}} = 1$ and, in [Subsection 6.4](#), we prove its correctness.

6.1 Finding an apex graph as a minor

Grids. Let $k \in \mathbb{N}_{\geq 2}$. We use the term *k-grid* to refer to the $(k \times k)$ -grid. We say that a graph is a *partially triangulated r-grid* if it can be obtained from an r -grid after adding edges in such a way that the remaining graph remains planar.

Let $k, r \in \mathbb{N}_{\geq 2}$. A vertex of a $(k \times r)$ -grid is called *internal* if it has degree four, and otherwise it is called *external*. We define the *perimeter* of a $(k \times r)$ -grid to be the unique cycle of the grid of length at least three that does not contain internal vertices.

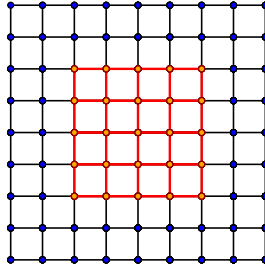


Figure 3: A 9-grid and its central 5-grid.

Let $r \in \mathbb{N}_{\geq 2}$ and H be an r -grid. Given an $i \in [\frac{r}{2}]$, we define the *i-th layer* of H recursively as follows. The first layer of H is its perimeter, while, if $i \geq 2$, the *i-th layer* of H is the $(i-1)$ -th layer of the grid created if we remove from H its perimeter. Given two odd integers $r, q \in \mathbb{N}_{\geq 3}$ such

that $q \leq r$ and an r -grid H , we define the *central q -grid* of H to be the graph obtained from H if we remove from H its $\frac{r-q}{2}$ first layers. See [Figure 3](#) for an illustration of the notions defined above. Given a partially triangulated r -grid H , we call *central q -grid* of H the subgraph of H induced by the vertices of the central q -grid of the underlying grid of H .

Given a graph G and a vertex $v \in V(G)$, we say that a graph H is a *v -fixed contraction* of G if H can be obtained from G after contracting edges that are not incident with v . A graph H is a *v -apex partially triangulated r -grid* if it can be obtained from a partially triangulated r -grid Γ after adding a new vertex v and some edges between v and vertices in $V(\Gamma)$. A *complete v -apex partially triangulated r -grid* is a graph obtained from a v -apex partially triangulated r -grid by adding every edge between v and the vertices of the grid.

The following result is a special case of [\[51, Lemma 29\]](#).

Proposition 22. *There exist three functions f_{13}, f_{14} , and $f_{15} : \mathbb{N} \rightarrow \mathbb{N}$, such that if $r \in \mathbb{N}$, H is a v -apex partially triangulated h -grid, where $v \in V(H)$ and $h \geq f_{13}(r) + 2 \cdot f_{15}(r)$, and vertex v has at least $f_{14}(r)$ neighbors in the central $f_{13}(r)$ -grid of $H \setminus \{v\}$, then H contains as a v -fixed contraction a complete v -apex partially triangulated r -grid. Moreover, $f_{13}(r) = \mathcal{O}(r^5)$, $f_{14}(r) = \mathcal{O}(r^6)$, and $f_{15}(r) = \mathcal{O}(r^2)$.*

The following easy observation intuitively states that every planar graph H is a minor of a big enough grid, where the relationship between the size of the grid and $|V(H)|$ is linear (see e.g., [\[50\]](#)).

Proposition 23. *There is a function $f_{16} : \mathbb{N} \rightarrow \mathbb{N}$ such that every planar graph on n vertices is a minor of the $f_{16}(n)$ -grid. Moreover, $f_{16}(n) = \mathcal{O}(n)$.*

In the proof of [Theorem 2](#), we will need the following result.

Lemma 24. *There exist three functions $f_{17}, f_{18}, f_{19} : \mathbb{N} \rightarrow \mathbb{N}$, such that if \mathcal{F} is a finite set of graphs containing an apex graph, G is a graph, A is a subset of $V(G)$, (W, \mathfrak{R}) is a flatness pair of $G \setminus A$ of height at least $f_{17}(s_{\mathcal{F}})$, $\tilde{\mathcal{Q}}$ is a (W, \mathfrak{R}) -canonical partition of $G \setminus A$, and there is a vertex in A that is adjacent, in G , to at least $f_{18}(s_{\mathcal{F}}) f_{19}(s_{\mathcal{F}})$ -internal bags of $\tilde{\mathcal{Q}}$, then $\mathcal{F} \leq_m G$.*

Proof. Let f_{13}, f_{14} , and f_{15} be the functions of [Proposition 22](#) and f_{16} be the function of [Proposition 23](#). We set $r = f_{16}(s_{\mathcal{F}} - 1)$, $f_{17}(s_{\mathcal{F}}) = f_{13}(r) + 2 \cdot f_{15}(r) + 2$, $f_{18}(s_{\mathcal{F}}) = f_{14}(r)$, and $f_{19}(s_{\mathcal{F}}) = f_{15}(r)$. Let G be a graph, $A \subseteq V(G)$, (W, \mathfrak{R}) be a flatness pair of $G \setminus A$ of height h , where $h \geq f_{17}(s_{\mathcal{F}})$, $\tilde{\mathcal{Q}}$ be a (W, \mathfrak{R}) -canonical partition of $G \setminus A$, and v be a vertex in A that is adjacent, in G , to at least $f_{18}(s_{\mathcal{F}}) f_{19}(s_{\mathcal{F}})$ -internal bags of $\tilde{\mathcal{Q}}$.

We contract every bag in $\tilde{\mathcal{Q}}$ to a vertex. Observe that this results in a planar graph (since (W, \mathfrak{R}) is a flatness pair) that is a partially triangulated $(h - 2)$ -grid $\bar{\Gamma}$ (whose vertices correspond to the internal bags of $\tilde{\mathcal{Q}}$) together with an extra vertex u_{ext} (which corresponds to the external bag of $\tilde{\mathcal{Q}}$) that is adjacent to all the vertices in the perimeter of $\bar{\Gamma}$. We contract an edge between u_{ext} and a vertex in the perimeter of $\bar{\Gamma}$ and we denote by Γ the obtained partially triangulated $(h - 2)$ -grid. We set Γ^{+v} to be the graph obtained from Γ by adding the vertex v and the edges $\{v, u\}$, if u is a vertex of Γ that corresponds to a bag $Q \in \tilde{\mathcal{Q}}$ that contains a vertex adjacent, in G , to v . Notice that Γ^{+v} is a v -apex partially triangulated $(h - 2)$ -grid that is a minor of G . Moreover, observe that since v is adjacent, in G , to vertices of an $f_{19}(s_{\mathcal{F}})$ -internal bag of $\tilde{\mathcal{Q}}$, then, since $f_{19}(s_{\mathcal{F}}) = f_{15}(r)$ and $h - 2 \geq f_{13}(r) + 2 \cdot f_{15}(r)$, vertex v is also adjacent to a vertex in the central

$f_{19}(s_{\mathcal{F}})$ -grid of $\Gamma = \Gamma^{+v} \setminus \{v\}$. Thus, v has at least $f_{18}(s_{\mathcal{F}})$ neighbors in the central $f_{19}(s_{\mathcal{F}})$ -grid of Γ . By [Proposition 22](#), Γ^{+v} contains as a v -fixed contraction a complete v -apex r -grid and therefore, since $r = f_{16}(s_{\mathcal{F}} - 1)$, by [Proposition 23](#) every apex graph on at most $s_{\mathcal{F}}$ vertices is a minor of G . Thus, $\mathcal{F} \leq_m G$, and the lemma follows. \square

6.2 Quickly finding a wall

In this subsection we prove [Lemma 26](#) that intuitively states that there is an algorithm that, given a graph G and two non-negative integers r and k , outputs either that (G, k) is a no-instance of \mathcal{F} -M-DELETION, or a report that the treewidth of G is polynomially bounded by r and k , or an r -wall of G . Before stating [Lemma 26](#), we present the following result of Kawarabayashi and Kobayashi [\[32\]](#), which provides a *linear* relation between the treewidth and the height of a largest wall in a minor-free graph.

Proposition 25. *There is a function $f_{20} : \mathbb{N} \rightarrow \mathbb{N}$ such that, for every $t, r \in \mathbb{N}$ and every graph G that does not contain K_t as a minor, if $\text{tw}(G) \geq f_{20}(t) \cdot r$, then G contains an r -wall as a subgraph. In particular, one may choose $f_{20}(t) = 2^{\mathcal{O}(t^2 \cdot \log t)}$.*

[Lemma 26](#) is a variation of [\[51, Lemma 11\]](#) that we prove in this subsection. The version presented here will be useful for the design of the algorithm of [Theorem 2](#). Recall that $s_{\mathcal{F}} = \max\{|V(H)| \mid H \in \mathcal{F}\}$.

Lemma 26. *There exists an algorithm with the following specifications:*

Find-Wall(G, r, k)

Input: A graph G , an odd $r \in \mathbb{N}_{\geq 3}$, and a $k \in \mathbb{N}$.

Output: One of the following:

- Either a report that G has treewidth at most $f_{20}(s_{\mathcal{F}}) \cdot r + k$, or
- an r -wall W of G , or
- a report that (G, k) is a no-instance of \mathcal{F} -M-DELETION.

Moreover, this algorithm runs in $2^{\mathcal{O}_{s_{\mathcal{F}}}(r^2 + (k+r) \cdot \log(k+r))} \cdot n$ -time.

The algorithm of [Lemma 26](#) is a recursive one. Namely, given an instance of this algorithm, we compute a smaller-sized instance and recurse. This is achieved by using the following result that is derived from [\[44\]](#). For a detailed analysis of the results of [\[44\]](#), see [\[3\]](#).

Proposition 27. *There exists an algorithm with the following specifications:*

Input: A graph G and a $t \in \mathbb{N}$ such that $|V(G)| \geq 12t^3$.

Output: A graph G^* such that $|V(G^*)| \leq (1 - \frac{1}{16t^2}) \cdot |V(G)|$ and:

- Either G^* is a subgraph of G such that $\text{tw}(G) = \text{tw}(G^*)$, or
- G^* is obtained from G after contracting the edges of a matching in G .

Moreover, this algorithm runs in $2^{\mathcal{O}(t)} \cdot n$ -time.

We now have all the ingredients to prove [Lemma 26](#).

Proof of Lemma 26. We set $c := f_{20}(s_{\mathcal{F}}) \cdot r + k$. We now describe a recursive algorithm as follows.

We first argue for the base case, namely when $|V(G)| < 12c^3$. To check whether $\text{tw}(G) \leq c$, we use the algorithm of Arnborg et al. [\[4\]](#), which runs in time $\mathcal{O}(|V(G)|^{c+2}) = 2^{\mathcal{O}_{s_{\mathcal{F}}}((r+k) \cdot \log(r+k))}$, and if this is the case, we report the same and stop. If not, we aim to find an r -wall of G or conclude that we are dealing with a no-instance. We first consider an arbitrary ordering $(v_1, \dots, v_{|V(G)|})$ of the vertices of G . For each $i \in [|V(G)|]$, we set G_i to be the graph induced by the vertices v_1, \dots, v_i . We iteratively run the algorithm of [Proposition 3](#) on G_i and c for increasing values of i . This algorithm runs in $2^{\mathcal{O}(c)} \cdot |V(G)| = 2^{\mathcal{O}_{s_{\mathcal{F}}}(r+k)}$ -time. Let $j \in [|V(G)|]$ be the smallest integer such that the above algorithm outputs a report that $\text{tw}(G_j) > c$ and notice that there exists a tree decomposition (\mathcal{T}_j, χ_j) of G_j (obtained by the one of G_{j-1} by adding the vertex v_j to all the bags) of width at most $5c + 5$. Thus, we can call the algorithm of [Proposition 4](#) with input $(G_j, 5c + 5, k)$ (which runs in $2^{\mathcal{O}_{s_{\mathcal{F}}}(c \cdot \log c)} \cdot |V(G_j)| = 2^{\mathcal{O}_{s_{\mathcal{F}}}((r+k) \cdot \log(r+k))}$ -time) in order to find, if it exists, a set $S_j \subseteq V(G_j)$ such that $|S_j| \leq k$ and $\mathcal{F} \not\leq_m G_j \setminus S_j$. We distinguish two cases.

- If such a set S_j does not exist, then we can safely report that (G, k) is a no-instance.
- If such a set S_j exists, then we call the algorithm of [Proposition 5](#) for $G_j \setminus S_j$ (and the decomposition of $G_j \setminus S_j$ obtained from (\mathcal{T}_j, χ_j) by removing the vertices of S_j from the all the bags in order to check whether it contains an elementary r -wall W as a minor. This algorithm runs in $2^{\mathcal{O}(c \cdot \log c)} \cdot r^{\mathcal{O}(c)} \cdot 2^{\mathcal{O}(r^2)} \cdot |V(G_j \setminus S_j)| = 2^{\mathcal{O}_{s_{\mathcal{F}}}((r+k) \cdot \log(r+k))} \cdot r^{\mathcal{O}_{s_{\mathcal{F}}}(r+k)} \cdot 2^{\mathcal{O}(r^2)} = 2^{\mathcal{O}_{s_{\mathcal{F}}}(r^2 + (r+k) \cdot \log(r+k))}$ -time, since $|E(W)| = \mathcal{O}(r^2)$. Since $G_j \setminus S_j$ does not contain $K_{s_{\mathcal{F}}}$ as a minor and $\text{tw}(G_j \setminus S_j) \geq c - k = f_{20}(s_{\mathcal{F}}) \cdot r$ and because of [Proposition 25](#), this algorithm will output an elementary r -wall W of $G_j \setminus S_j$. We also return W as a wall of G .

Therefore, in the case where $|V(G)| < 12c^3$, we obtain one of the three possible outputs in time $2^{\mathcal{O}_{s_{\mathcal{F}}}(r^2 + (r+k) \log(r+k))}$.

If $|V(G)| \geq 12c^3$, then we call the algorithm of [Proposition 27](#) with input (G, c) , which outputs a graph G^* such that $|V(G^*)| \leq (1 - \frac{1}{16c^2}) \cdot |V(G)|$ and

- either G^* is a subgraph of G such that $\text{tw}(G) = \text{tw}(G^*)$, or
- G^* is obtained from G after contracting the edges of a matching in G .

In both cases, we recursively call the algorithm on G^* and we distinguish the following two cases.

Case 1: G^* is a subgraph of G such that $\text{tw}(G) = \text{tw}(G^*)$.

- If the recursive call on G^* reports that $\text{tw}(G^*) \leq c$, then we return that $\text{tw}(G) \leq c$.
- If the recursive call on G^* outputs an r -wall W of G^* , then we return W as a wall of G .
- If (G^*, k) is a no-instance, then we report that (G, k) is also a no-instance.

Case 2: G^* is obtained from G after contracting the edges of a matching in G .

- (a) If the recursive call on G^* reports that $\text{tw}(G^*) \leq c$, then we do the following. We first notice that the fact that $\text{tw}(G^*) \leq c$ implies that $\text{tw}(G) \leq 2c$, since we can obtain a tree decomposition (\mathcal{T}, χ) of G from a tree decomposition (\mathcal{T}^*, χ^*) of G^* , by replacing, in every $t \in \mathcal{T}^*$, every occurrence of a vertex of G^* that is a result of an edge contraction by its endpoints in G . Thus, we can call the algorithm of [Proposition 4](#) with input $(G, 2c, k)$ (which runs in $2^{\mathcal{O}_{s_{\mathcal{F}}}(c \log c)} \cdot n$ -time) in order to find, if it exists, a set S such that $|S| \leq k$ and $\mathcal{F} \not\leq_m G \setminus S$. We distinguish again two cases.
- If such a set S does not exist, then the algorithm reports that (G, k) is a **no-instance**.
 - If such a set S exists, then we apply the algorithm of [Proposition 3](#) with input $(G \setminus S, 2c)$ (which runs in $2^{\mathcal{O}(c)} \cdot n$ -time) and we get a tree decomposition of $G \setminus S$ of width at most $10c + 4$. Using this decomposition, we call the algorithm of [Proposition 5](#) for $G \setminus S$ in order to check whether it contains an elementary r -wall W as a minor. This algorithm runs in $2^{\mathcal{O}(c \cdot \log c)} \cdot r^{\mathcal{O}(c)} \cdot 2^{\mathcal{O}(r^2)} \cdot n = 2^{\mathcal{O}_{s_{\mathcal{F}}}((r+k) \cdot \log(r+k))} \cdot r^{\mathcal{O}_{s_{\mathcal{F}}}(r+k)} \cdot 2^{\mathcal{O}(r^2)} \cdot n = 2^{\mathcal{O}_{s_{\mathcal{F}}}(r^2 + (r+k) \cdot \log(r+k))} \cdot n$ -time, since $|E(G \setminus S)| = \mathcal{O}(n)$ and $|E(W)| = \mathcal{O}(r^2)$. If this algorithm outputs an elementary r -wall W of $G \setminus S$, then we output W . Otherwise, we can safely report, because of [Proposition 25](#), that $\text{tw}(G) \leq f_{20}(s_{\mathcal{F}}) \cdot r + k = c$.
- (b) If the recursive call on G^* outputs an r -wall W^* of G^* , then by uncontracting the edges of M in W^* we can return an r -wall of G .
- (c) If (G^*, k) is a **no-instance**, then we report that (G, k) is also a **no-instance**.

It is easy to see that the running time of the above algorithm is

$$T(n, k, r) \leq T\left(\left(1 - \frac{1}{12c^2}\right) \cdot n, k, r\right) + 2^{\mathcal{O}_{s_{\mathcal{F}}}(r^2 + (r+k) \log(r+k))} \cdot n,$$

which implies that $T(n, k, r) = 2^{\mathcal{O}_{s_{\mathcal{F}}}(r^2 + (r+k) \log(r+k))} \cdot n$, as claimed. \square

6.3 The algorithm

In this subsection we prove that, in the case where $a_{\mathcal{F}} = 1$, there is an algorithm that solves \mathcal{F} -M-DELETION in time $2^{\mathcal{O}_{s_{\mathcal{F}}}(k^2(c+2))} \cdot n^2$, where $c = c_{a, \ell_{\mathcal{F}}}$ and $a = f_2(s_{\mathcal{F}})$. Note that the existence of such an algorithm implies [Theorem 2](#).

Let G be graph and let W be a wall of G . The *drop*, denoted by $D_{W'}$, of a subwall W' of W is defined as follows. If contract in G the perimeter of W to a single vertex v , $D_{W'}$ is the unique 2-connected component of the resulting graph that contains the interior of W' . We call the vertex v the *pole* of the drop $D_{W'}$.

Our algorithm avoids iterative compression in a similar fashion as done by Marx and Schlotter in [\[42\]](#) for the VERTEX PLANARIZATION problem. The algorithm has three main steps. We first set $a = f_2(s_{\mathcal{F}})$ and we define $d = f_8(a, \ell_{\mathcal{F}})$,

$$\begin{aligned} \beta &= f_7(0, \ell_{\mathcal{F}}, 3, k), & \lambda &= f_{18}(s_{\mathcal{F}}) \cdot (a + 1), & q &= f_{19}(s_{\mathcal{F}}), \\ \eta &= f_{12}(\lambda + 1, \beta, q), & z &= \text{odd}(\max\{f_{17}(s_{\mathcal{F}}), \eta\}), & w &= f_6(z, a, 0, d), \\ b &= 3 + f_3(s_{\mathcal{F}}) \cdot w, & l &= f_{10}(1, s_{\mathcal{F}}, k) \cdot (k + a), & p &= f_{11}(1, s_{\mathcal{F}}, k), \\ h &= f_{12}(l + 1, b, p), & r &= \text{odd}(\max\{f_9(1, s_{\mathcal{F}}, k), h\}), \text{ and } & R &= \text{odd}(f_1(s_{\mathcal{F}}) \cdot r + k) = \mathcal{O}_{s_{\mathcal{F}}}(k^{c+2}). \end{aligned}$$

Step 1. Run the algorithm of Lemma 26 with input (G, R, k) and, in $2^{\mathcal{O}_{s_{\mathcal{F}}}(k^{2(c+2)})} \cdot n$ -time, either report a no-answer, or conclude that $\text{tw}(G) \leq f_{20}(s_{\mathcal{F}}) \cdot R + k$ and solve \mathcal{F} -M-DELETION in $2^{\mathcal{O}_{s_{\mathcal{F}}}(k^{c+2} \cdot \log k)} \cdot n$ -time using the algorithm of Proposition 4, or obtain an R -wall \tilde{W} of G . In the third case, consider all the $\binom{R}{b}^2 = 2^{\mathcal{O}_{s_{\mathcal{F}}}(k^c \log k)}$ b -subwalls of \tilde{W} and for each one of them, say W , construct its drop D_W , and run the algorithm of Proposition 10 with input $(D_W \setminus \{v_W\}, w, s_{\mathcal{F}})$, where v_W is the pole of D_W . This takes time $2^{\mathcal{O}_{s_{\mathcal{F}}}(k^c)} \cdot n$. If for some of these drops the result is a set $A \subseteq V(D_W \setminus \{v_W\})$ with $|A| \leq a$ and a regular flatness pair (W', \mathfrak{R}') of $(D_W \setminus \{v_W\}) \setminus A$ of height w , then proceed to Step 2, otherwise proceed to Step 3.

Step 2. We apply the algorithm of Proposition 14 with input $(z, a, 0, d, D_W \setminus \{v_W\}, A, W', \mathfrak{R}')$, which outputs a flatness pair $(\check{W}, \check{\mathfrak{R}})$ of $(D_W \setminus \{v_W\}) \setminus A$ of height z that is d -homogeneous with respect to $\binom{A}{\leq 0} = \{\emptyset\}$ and is a W^* -tilt of (W', \mathfrak{R}') for some subwall W^* of W' . This takes $2^{\mathcal{O}_{s_{\mathcal{F}}}(k \log k)} \cdot n$ -time. By Observation 7, $(\check{W}, \check{\mathfrak{R}})$ is regular. Let $A^* := A \cup \{v_W\}$ and keep in mind that $(D_W \setminus \{v_W\}) \setminus A = D_W \setminus A^*$. Consider all the β -subwalls of \check{W} , which are at most $\binom{z}{\beta}^2$ many, and for each of them, say \hat{W} , check in linear time whether there is an edge, in D_W , between A^* and $V(\mathbf{Uinfluence}_{\check{\mathfrak{R}}}(\hat{W}))$. If this is the case for every such a subwall, then proceed to Step 3. If not, let \hat{W} be a β -subwall of \check{W} such that no vertex of A^* is adjacent to $V(\mathbf{Uinfluence}_{\check{\mathfrak{R}}}(\hat{W}))$. By applying Proposition 8 for the flatness pair $(\check{W}, \check{\mathfrak{R}})$ of $D_W \setminus A^*$ and the subwall \hat{W} of \check{W} , we obtain in linear time a \hat{W} -tilt (W'', \mathfrak{R}'') of $(\check{W}, \check{\mathfrak{R}})$. Keep in mind that (W'', \mathfrak{R}'') is a flatness pair of $D_W \setminus A^*$ which is also regular and d -homogeneous with respect to $\{\emptyset\}$, due to Observation 7 and Observation 12, respectively. Also, notice that, since (W'', \mathfrak{R}'') is a \hat{W} -tilt of (W', \mathfrak{R}') , $\text{compass}_{\mathfrak{R}''}(W'')$ is a subgraph of $\mathbf{Uinfluence}_{\check{\mathfrak{R}}}(\hat{W})$ and therefore no vertex of A^* is adjacent, in D_W , to a vertex of $\text{compass}_{\mathfrak{R}''}(W'')$. The latter implies that we can obtain a 7-tuple $\tilde{\mathfrak{R}}''$ from \mathfrak{R}'' by adding all vertices of $G \setminus V(\text{compass}_{\mathfrak{R}''}(W''))$ such that $\text{compass}_{\tilde{\mathfrak{R}}''}(W'') = \text{compass}_{\mathfrak{R}''}(W'')$ and $(W'', \tilde{\mathfrak{R}}'')$ is a flatness pair of G . We apply Find-Irrelevant-Vertex of Corollary 17 with input $(k, 0, G, \emptyset, W'', \tilde{\mathfrak{R}}'')$ and obtain, in linear time, an irrelevant vertex v such that (G, k) and $(G \setminus v, k)$ are equivalent instances of \mathcal{F} -M-DELETION. Then the algorithm runs recursively on the equivalent instance $(G \setminus v, k)$.

Notice that Step 2 can be seen as the *irrelevant vertex case* of our algorithm.

Step 3. Consider all the r -subwalls of \tilde{W} , which are at most $\binom{R}{r}^2 = 2^{\mathcal{O}_{s_{\mathcal{F}}}(k^c \log k)}$ many, and for each of them, compute its canonical partition \mathcal{Q} . Then, for each p -internal bag Q of \mathcal{Q} , add a new vertex v_Q and make it adjacent to all vertices in Q , then add a new vertex x_{all} and make it adjacent to all x_Q 's, and in the resulting graph, for every vertex y of G that is not in the union of the internal bags of \mathcal{Q} , check, in time $\mathcal{O}(k \cdot |E(G)|) = \mathcal{O}_{s_{\mathcal{F}}}(k \cdot n)$ (using standard flow techniques), whether there are $f_{10}(1, s_{\mathcal{F}}, k)$ internally vertex-disjoint paths from x_{all} to y . If this is indeed the case for some y , then y should belong to every solution of \mathcal{F} -M-DELETION for the instance (G, k) , and the algorithm runs recursively on the equivalent instance $(G \setminus y, k - 1)$. If no such a vertex y exists, then report that (G, k) is a no-instance of \mathcal{F} -M-DELETION.

Note that Step 3 can be seen as a trivial *branching case* where the only choice is vertex y .

Notice that the third step of the algorithm, when applied takes time $2^{\mathcal{O}_{s_{\mathcal{F}}}(k^c \log k)} \cdot n^2$. However, it cannot be applied more than k times during the course of the algorithm. As the first step runs in time $2^{\mathcal{O}_{s_{\mathcal{F}}}(k^{2(c+2)} \log k)} \cdot n$, and the second step runs in time $2^{\mathcal{O}_{s_{\mathcal{F}}}(k \log k)} \cdot n$, they may be applied

at most n times, and the claimed time complexity follows.

6.4 Correctness of the algorithm

In this subsection we prove the correctness of the algorithm presented in [Subsection 6.3](#).

Recall that $a = f_2(s_{\mathcal{F}})$, $d = f_8(a, \ell_{\mathcal{F}})$,

$$\begin{aligned} \beta &= f_7(0, \ell_{\mathcal{F}}, 3, k), & \lambda &= f_{18}(s_{\mathcal{F}}) \cdot (a + 1), & q &= f_{19}(s_{\mathcal{F}}), \\ \eta &= f_{12}(\lambda + 1, \beta, q), & z &= \text{odd}(\max\{f_{17}(s_{\mathcal{F}}), \eta\}), & w &= f_6(z, a, 0, d), \\ b &= 3 + f_3(s_{\mathcal{F}}) \cdot w, & l &= f_{10}(1, s_{\mathcal{F}}, k) \cdot (k + a), & p &= f_{11}(1, s_{\mathcal{F}}, k), \\ h &= f_{12}(l + 1, b, p), & r &= \text{odd}(\max\{f_9(1, s_{\mathcal{F}}, k), h\}), \text{ and } & R &= \text{odd}(f_1(s_{\mathcal{F}}) \cdot r + k) = \mathcal{O}_{s_{\mathcal{F}}}(k^{c+2}). \end{aligned}$$

Let (G, k) be a yes-instance and let S be a solution, i.e., a subset of $V(G)$ of size at most k such that $G \setminus S \in \text{exc}(\mathcal{F})$ and let \tilde{W} be an R -wall of G . Then, since $R \geq f_1(s_{\mathcal{F}}) \cdot r + k$, there is an $(f_1(s_{\mathcal{F}}) \cdot r)$ -subwall of \tilde{W} , say W^* , that does not contain vertices of S . The wall W^* is an $(f_1(s_{\mathcal{F}}) \cdot r)$ -wall of $G \setminus S$, and therefore by [Proposition 9](#) there is a set $A \subseteq V(G \setminus S)$, with $|A| \leq a$, and a flatness pair (W, \mathfrak{R}) of $G \setminus (S \cup A)$ of height r .

Let \tilde{Q} be a (W, \mathfrak{R}) -canonical partition of $G \setminus (S \cup A)$. For each p -internal bag Q of \tilde{Q} , add a new vertex v_Q and make it adjacent to all vertices in Q , then add a new vertex x_{all} and make it adjacent to all v_Q 's. In the resulting graph, if there are $f_{10}(1, s_{\mathcal{F}}, k)$ internally vertex-disjoint paths from x_{all} to a vertex $v \in S \cup A$, then this is checked in Step 3 (since connectivity of the internal bags implies that every such a path can be rerouted in order to intersect the wall) and the algorithm correctly (due to [Proposition 18](#)) runs recursively on the equivalent instance $(G \setminus v, k - 1)$. If this is not the case, then for each vertex v of $S \cup A$ there are less than $f_{10}(1, s_{\mathcal{F}}, k)$ p -internal bags of \tilde{Q} that contain vertices adjacent to v . This means that the p -internal bags of \tilde{Q} that contain vertices adjacent to some vertex of $S \cup A$ are at most $f_{10}(1, s_{\mathcal{F}}, k) \cdot (k + a) = l$.

We consider a family $\mathcal{W} = \{W_1, \dots, W_{l+1}\}$ of $l + 1$ b -subwalls of W such that for every $i \in [l + 1]$, $\mathbf{U}\text{influence}_{\mathfrak{R}}(W_i)$ is a subgraph of $\mathbf{U}\{Q \mid Q \text{ is a } p\text{-internal bag of } \tilde{Q}\}$ and for every $i, j \in [l + 1]$, with $i \neq j$, there is no internal bag of \tilde{Q} that contains vertices of both $V(\mathbf{U}\text{influence}_{\mathfrak{R}}(W_i))$ and $V(\mathbf{U}\text{influence}_{\mathfrak{R}}(W_j))$. The existence of \mathcal{W} follows from the fact that $r \geq h = f_{12}(l + 1, b, p)$ and [Proposition 19](#).

The fact that the p -internal bags of \tilde{Q} that contain vertices adjacent to some vertex of $S \cup A$ are at most l implies that there exists an $i \in [l + 1]$ such that no vertex of $V(\mathbf{U}\text{influence}_{\mathfrak{R}}(W_i))$ is adjacent, in G , to a vertex in $S \cup A$. Thus, if D_{W_i} is the drop of W_i , and $D_{\bar{W}_i} := D_{W_i} \setminus \{v_{W_i}\}$, where v_{W_i} is the pole of D_{W_i} , then $D_{\bar{W}_i} \leq_m G \setminus (S \cup A)$. This, in turn, implies that $D_{\bar{W}_i} \in \text{exc}(\mathcal{F})$ and therefore $D_{\bar{W}_i}$ does not contain $K_{s_{\mathcal{F}}}$ as a minor. Additionally, we notice that $D_{\bar{W}_i}$ contains the central $(b - 2)$ -subwall \bar{W}_i of W_i as a subgraph and since \bar{W}_i has height $b - 2 = f_3(s_{\mathcal{F}}) \cdot w + 1$, it holds that $\text{tw}(D_{\bar{W}_i}) > f_3(s_{\mathcal{F}}) \cdot w$. Therefore, by applying the algorithm of [Proposition 10](#) with input $(D_{\bar{W}_i}, w, s_{\mathcal{F}})$, we must find a set $A \subseteq V(D_{\bar{W}_i} \setminus \{v_{W_i}\})$ with $|A| \leq a$ and a regular flatness pair (W', \mathfrak{R}') of $D_{\bar{W}_i} \setminus A$ of height w . This should be detected in Step 1.

We apply the algorithm of [Proposition 14](#) with input $(z, a, 0, d, D_{\bar{W}_i}, A, W', \mathfrak{R}')$, which outputs a flatness pair $(\tilde{W}, \tilde{\mathfrak{R}})$ of $D_{\bar{W}_i} \setminus A$ of height z that is d -homogeneous with respect to $\binom{A}{\leq 0} = \{\emptyset\}$ and is a W^* -tilt of (W', \mathfrak{R}') for some subwall W^* of W' . We set $A^* := A \cup \{v_{W_i}\}$ and keep in mind

that $D_{W_i} \setminus A^* = D_{W_i}^- \setminus A$. Let \tilde{Q}' be a $(\check{W}, \check{\mathfrak{R}})$ -canonical partition of $D_{W_i} \setminus A^*$. Since D_{W_i} is a subgraph of $D_{W_i}^-$ and $D_{W_i}^- \in \mathbf{exc}(\mathcal{F})$, we observe that $D_{W_i} \in \mathbf{exc}(\mathcal{F})$. Hence, as a consequence of [Lemma 24](#), every vertex in A^* has neighbors in less than $f_{18}(s_{\mathcal{F}})$ q -internal bags of Q'' . Therefore, since $\lambda = f_{18}(s_{\mathcal{F}}) \cdot (a + 1) \geq f_{18}(s_{\mathcal{F}}) \cdot |A^*|$ and $\rho \geq \eta = f_{12}(\lambda + 1, \beta, q)$, it follows, due to [Proposition 19](#), that there is a β -subwall \bar{W} of W' such that no vertex of $V(\mathbf{Uinfluence}_{\mathfrak{Y}'}(\bar{W}))$ is adjacent, in D_{W_i} , to a vertex in A^* . Therefore, this β -subwall \bar{W} of W' should be detected in Step 2.

7 Algorithms for variants of VERTEX DELETION TO \mathcal{G}

We now present how our approach can be modified so to obtain FPT-algorithms for several variants of the VERTEX DELETION TO \mathcal{G} problem.

7.1 The general framework

Notice that both algorithms in [Section 5](#) and [Section 6](#) are based on one of the following three scenarios for VERTEX DELETION TO \mathcal{G} with input (G, k) .

[*Bounded treewidth case*] A tree decomposition of G of width $k^{\mathcal{O}(1)}$, or

[*Branching case*] a set B with $|B| = \mathcal{O}(k)$ such that (G, k) is a **yes**-instance if and only if, for some $x \in B$, $(G \setminus x, k - 1)$ is a **yes**-instance, or

[*Irrelevant vertex case*] a vertex x such that (G, k) is a **yes**-instance if and only if $(G \setminus x, k)$ is a **yes**-instance,

For each of the variants of VERTEX DELETION TO \mathcal{G} that we consider, the algorithm recursively runs on an equivalent instance with one vertex less (*irrelevant vertex case*), or branches on $\mathcal{O}(k)$ equivalent instances where both k and the size of the graph are one less (*branching case*). The eventual outcome is to reduce the problem to the *bounded treewidth case*, producing a tree decomposition of G of width $k^{\mathcal{O}(1)}$ (*bounded treewidth case*). In each variant of the problem, the *bounded treewidth case* can be treated by a suitable modification of the dynamic programming algorithm of [\[6\]](#), taking into account the main combinatorial result in [\[5\]](#). For each variant that we treat, the algorithm of [Section 5](#) assumes that we have at hand a solution of VERTEX DELETION TO \mathcal{G} of size k , which can be found by the algorithm in [Theorem 1](#).

We next present the problem variants and explain how to adapt the *branching case* and the *irrelevant vertex case* for each of them.

7.2 Variants of VERTEX DELETION TO \mathcal{G}

A common part of the inputs of all problems below is the pair (G, k) , where G is a graph and k is a non-negative integer, i.e., an input of VERTEX DELETION TO \mathcal{G} .

Annotated. In the *annotated version* of VERTEX DELETION TO \mathcal{G} , the input is a triple (G, k, R) , where $R \subseteq V(G)$, and the problem asks for a solution S with $S \subseteq R$.

[*Branching case*]: we branch on $(G \setminus x, k - 1, R \setminus x)$ for all the annotated vertices of B , i.e., for every $x \in B \cap R$. If there is no such a vertex, we report that (G, k, R) is a no-instance.

[*Irrelevant vertex case*]: we recurse on $(G \setminus x, k, R \setminus x)$, as every irrelevant vertex for the original problem is also an irrelevant vertex for its annotated variant.

Modulo. In the *modulo version* of VERTEX DELETION TO \mathcal{G} , the input is a quadruple (G, k, q, p) where q, p are integers, p is a prime, and $q < p$. The question is whether there is a solution S of size at most k where $|S| \equiv q \pmod{p}$.

[*Branching case*]: we branch on $(G \setminus x, k - 1, q - 1 \pmod{p}, p)$ for every $x \in B$.

[*Irrelevant vertex case*]: it is the same, as every irrelevant vertex for the original problem is also an irrelevant vertex for this variant.

Weighted. In the *weighted version* of VERTEX DELETION TO \mathcal{G} , the input is a triple (G, k, \mathbf{w}) where $\mathbf{w} : V(G) \rightarrow \mathbb{R}$ is a weight function assigning positive real weights to the vertices of G . The problem asks for a solution S with $\sum_{v \in S} \mathbf{w}(v) \leq k$.

[*Branching case*]: we branch on $(G \setminus x, k - \mathbf{w}(x), \mathbf{w} \setminus \{(x, \mathbf{w}(x))\})$, for every $x \in B$.

[*Irrelevant vertex case*]: it is the same, as every irrelevant vertex for the original problem is also an irrelevant vertex for its weighted variant.

For the above problem, if $\varepsilon = \min\{\mathbf{w}(x) \mid x \in V(G)\}$, then the parametric dependence of the derived algorithm is $2^{\text{poly}(k/\varepsilon)}$, as the size of the solution S is at most k/ε .

Counting. In the *counting version* of VERTEX DELETION TO \mathcal{G} with input (G, k) , the output is the number $\#\mathcal{G}(G, k)$ of all solutions of VERTEX DELETION TO \mathcal{G} of size (at most) k . We treat the case where we count solutions of size exactly k as the “ $\leq k$ ”-case can be easily reduced to it.

[*Branching case*]: return $\sum_{x \in B} \#\mathcal{G}(G \setminus x, k - 1)$.

[*Irrelevant vertex case*]: return $\#\mathcal{G}(G \setminus x, k - 1) + \#\mathcal{G}(G \setminus x, k)$.

The above creates $T(n, k)$ subproblems on bounded treewidth graphs, where

$$T(n, k) = \max\{\mathcal{O}(k) \cdot T(n - 1, k - 1), T(n - 1, k - 1) + T(n - 1, k)\}.$$

This makes a total of $2^{\mathcal{O}(k)} \cdot n$ problems, each solvable in $2^{k^{\mathcal{O}(1)}} \cdot n$ -time by the counting version of the dynamic programming algorithm of [6], taking into account the analysis of [5].

Colored. In the *colored version* of VERTEX DELETION TO \mathcal{G} , the input is a triple (G, k, χ) where $\chi : V(G) \rightarrow [k]$ is a function assigning colors from $[k]$ to the vertices of G . The problem asks for a solution S to VERTEX DELETION TO \mathcal{G} that carries all k colors, i.e., for each $i \in [k]$, $|S \cap \chi^{-1}(i)| = 1$. (Notice that the requested solution must have size exactly k .) To deal with this problem, we deal with its annotated version where we permit $\chi : V(G) \rightarrow \{0, 1, \dots, k\}$, i.e., the vertices in $R := \bigcup_{i \in [k]} \chi^{-1}(i)$ are annotated, while the vertices in $\chi^{-1}(0)$ cannot participate in a solution (we call these vertices *black* vertices).

[*Branching case*]: we branch on $(G \setminus x, k - 1, \chi|_x)$, for every $x \in B \cap R$, where

$$\chi|_x = \{(v, \chi(v)) \mid v \in V(G) \setminus \chi^{-1}(\chi(x))\} \cup \{(v, 0) \mid v \in \chi^{-1}(\chi(x)) \setminus \{x\}\}.$$

The new coloring $\chi|_x$ turns black all vertices carrying the color of x . If $B \cap R = \emptyset$, i.e., all vertices in B are black, then we have a no-instance.

[*Irrelevant vertex case*]: Before we apply the irrelevant vertex case, we check whether there is some $i \in [k]$ with $|\chi^{-1}(i)| \leq 1$, i.e., there is a color in $[k]$ that appears once or is not used at all. If $|\chi^{-1}(i)| = 0$, then we return that we have a no-instance. If $\chi^{-1}(i) = \{x\}$, then x should belong to every possible solution and, in this case, we recurse on $(G \setminus x, k - 1, \chi \setminus \{(x, \chi(x))\})$. If now each color is used at least twice, we recurse on $(G \setminus x, k, \chi \setminus \{(x, \chi(x))\})$, i.e., apply the irrelevant vertex case.

8 Discussion and concluding remarks

Apices of topological minors. Very recently, Fomin et al. [24] gave an FPT-algorithm running in time $\mathcal{O}_{s,k}(n^4)$ for the following problem: for a fixed finite family of graphs \mathcal{F} , each on at most s vertices, decide whether an n -vertex input graph G is a k -apex of the class of graphs that exclude the graphs in \mathcal{F} as *topological minors*⁴. For every graph H , there is a finite set \mathcal{H} of graphs such that a graph G contains H as a minor if and only if G contains a graph in \mathcal{H} as a topological minor. Based on this observation, the result of Fomin et al. [24] implies that for every minor-closed graph class \mathcal{G} , VERTEX DELETION TO \mathcal{G} admits an $\mathcal{O}(h(k, s) \cdot n^4)$ -time FPT-algorithm, where s is the maximum size of an obstruction of \mathcal{G} . Notice that this implication is a solid improvement on VERTEX DELETION TO \mathcal{G} with respect to the result of [2], where only the computability of h is proved. However, as mentioned in [24], even for fixed values of s , the dependence of h on k is humongous. Therefore, Theorem 1 can be seen as orthogonal to the result of [24]. An interesting question is whether the ideas of this paper can be useful towards improving the the parametric dependence of the algorithm of [24].

Limitations of the irrelevant vertex technique. An intriguing open question is whether VERTEX DELETION TO \mathcal{G} admits a $2^{\mathcal{O}_{s\mathcal{F}}(k^c)} \cdot n^{\mathcal{O}(1)}$ -time algorithm for some universal constant c that does *not* depend on the class \mathcal{G} . Clearly, this is not the case of the algorithms of Theorem 1 and Theorem 2, running in time $2^{\mathcal{O}_{s\mathcal{F}}(k^{2(c+2)})} \cdot n^3$ and $2^{\mathcal{O}_s(k^{2(c+2)})} \cdot n^2$, respectively, where c is the *palette-variety* of the minor-obstruction set \mathcal{F} of \mathcal{G} which, from the corresponding proofs, is estimated to

⁴The definition is as minors, except that only edges incident with at least a degree-two vertex can be contracted.

be $c = 2^{2^{\mathcal{O}(s^2 \cdot \log s)}}$ and $c = 2^{2^{\mathcal{O}(s^{24} \cdot \log s)}}$, respectively (recall that s is the maximum size of a minor-obstruction of \mathcal{G}). We tend to believe that this dependence is unavoidable if we want to use the *irrelevant vertex technique*, as it reflects the *price of homogeneity*, mentioned in [Subsection 3.4](#). Having homogeneous walls is critical for the application of this technique (see [Lemma 16](#)) when \mathcal{G} is more general than surface-embeddable graphs (in the bounded genus case, all subwalls are already homogeneous). Is there a way to prove that this behavior is unavoidable subject to some complexity assumption? An interesting result of this flavor, concerning the existence of polynomial kernels for [VERTEX DELETION TO \$\mathcal{G}\$](#) , was given by Giannopoulou et al. [\[27\]](#) who proved that, even for minor-closed families \mathcal{G} that exclude a planar graph, the dependence on \mathcal{G} of the degree of the polynomial kernel, which exists because of [\[23\]](#), is unavoidable subject to reasonable complexity assumptions.

Kernelization. As mentioned above, Giannopoulou et al. [\[27\]](#) provided a polynomial kernel for [VERTEX DELETION TO \$\mathcal{G}\$](#) in the case where \mathcal{G} excludes a planar graph. To the best of our knowledge, the existence of a polynomial kernel is open for every family \mathcal{G} whose obstructions are all non-planar. In particular, no polynomial kernel is known even for the [VERTEX PLANARIZATION](#) problem.

Other modification operations. Another direction is to consider graph modification to a minor-closed graph class for different modification operations. Our approach becomes just simpler in the case where the modification operation is edge removal or edge contraction. In these two cases, we immediately get rid of the branching part of our algorithms and only the irrelevant vertex part needs to be applied. Another challenge is to combine all aforementioned modifications. This is more complicated (and tedious) but not more complex. What is really more complex is to consider as well edge additions. We leave it as an open research challenge (a first step was done for the case of planar graphs [\[22\]](#)).

Lower bounds. Concerning lower bounds for [VERTEX DELETION TO \$\mathcal{G}\$](#) under the Exponential Time Hypothesis [\[28\]](#), we are not aware of any lower bound stronger than $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ for any minor-closed class \mathcal{G} . This lower bound already applies when $\mathcal{F} = \{K_2\}$, i.e., for the [VERTEX COVER](#) problem [\[8, 28\]](#).

Acknowledgements. We wish to thank the anonymous reviewers of the conference version of this paper for their comments and remarks that improved the presentation of this manuscript. Moreover, we wish to thank Dániel Marx for his valuable remarks and advises regarding the variants of the problem discussed in [Section 7](#).

References

- [1] Isolde Adler, Frederic Dorn, Fedor V. Fomin, Ignasi Sau, and Dimitrios M. Thilikos. Faster parameterized algorithms for minor containment. *Theoretical Computer Science*, 412(50):7018–7028, 2011. doi:10.1016/j.tcs.2011.09.015. [5](#), [9](#)

- [2] Isolde Adler, Martin Grohe, and Stephan Kreutzer. Computing excluded minors. In *Proc. of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 641–650, 2008. doi:10.5555/1347082.1347153. 4, 31
- [3] Ernst Althaus and Sarah Ziegler. Optimal Tree Decompositions Revisited: A Simpler Linear-Time FPT Algorithm. *CoRR*, abs/1912.09144, 2019. arXiv:1912.09144. 5, 24
- [4] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k -tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987. doi:10.1137/0608024. 25
- [5] Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. A complexity dichotomy for hitting connected minors on bounded treewidth graphs: the chair and the banner draw the boundary. In *Proc. of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 951–970, 2020. doi:10.1137/1.9781611975994.57. 5, 6, 8, 10, 14, 16, 17, 20, 29, 30
- [6] Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. I. General upper bounds. *SIAM Journal on Discrete Mathematics*, 34(3):1623–1648, 2020. doi:10.1137/19M1287146. 6, 14, 29, 30
- [7] Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. II. Single-exponential algorithms. *Theoretical Computer Science*, 814:135–152, 2020. doi:10.1016/j.tcs.2020.01.026. 6
- [8] Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. III. Lower bounds. *Journal of Computer and System Sciences*, 109:56–77, 2020. doi:10.1016/j.jcss.2019.11.002. 6, 32
- [9] Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016. doi:10.1137/130947374. 7
- [10] Hans L. Bodlaender, Pinar Heggernes, and Daniel Lokshtanov. Graph modification problems (dagstuhl seminar 14071). *Dagstuhl Reports*, 4(2):38–59, 2014. doi:10.4230/DagRep.4.2.38. 3
- [11] Hans L. Bodlaender, Hirotaka Ono, and Yota Otachi. A faster parameterized algorithm for pseudoforest deletion. *Discrete Applied Mathematics*, 236:42–56, 2018. doi:10.1016/j.dam.2017.10.018. 4
- [12] Andries E. Brouwer and Henk Jan Veldman. Contractibility and NP-completeness. *Journal of Graph Theory*, 11(1):71–79, 1987. doi:10.1002/jgt.3190110111. 3
- [13] Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theoretical Computer Science*, 411(40-42):3736–3756, 2010. doi:10.1016/j.tcs.2010.06.026. 4
- [14] Christophe Crespelle, Pål Grønås Drange, Fedor V. Fomin, and Petr A. Golovach. A survey of parameterized algorithms and the complexity of edge modification. *CoRR*, abs/2001.06867, 2013. arXiv:2001.06867. 3

- [15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3. 3, 16, 20
- [16] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. An improved FPT algorithm and a quadratic kernel for pathwidth one vertex deletion. *Algorithmica*, 64(1):170–188, 2012. doi:10.1007/s00453-011-9578-2. 4
- [17] Reinhard Diestel. *Graph Theory*, volume 173. Springer-Verlag, 5th edition, 2017. doi:10.1007/978-3-662-53622-3. 7
- [18] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1. 3
- [19] Michael R. Fellows, Jan Kratochvíl, Matthias Middendorf, and Frank Pfeiffer. The complexity of induced minors and related problems. *Algorithmica*, 13(3):266–282, 1995. doi:10.1007/BF01190507. 3
- [20] Michael R. Fellows and Michael A. Langston. Nonconstructive tools for proving polynomial-time decidability. *Journal of the ACM*, 35(3):727–739, 1988. doi:10.1145/44483.44491. 4
- [21] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-x. 3
- [22] Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. Modification to planarity is fixed parameter tractable. In *Proc. of the 36th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 126 of *LIPICs*, pages 28:1–28:17, 2019. doi:10.4230/LIPICs.STACS.2019.28. 32
- [23] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar \mathcal{F} -Deletion: Approximation, Kernelization and Optimal FPT Algorithms. In *Proc. of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 470–479, 2012. doi:10.1109/FOCS.2012.62. 4, 6, 32
- [24] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Hitting topological minors is FPT. In *Proc. of the 52nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 1317–1326, 2020. doi:10.1145/3357713.3384318. 6, 31
- [25] Fedor V. Fomin, Saket Saurabh, and Neeldhara Misra. Graph modification problems: A modern perspective. In *Proc. of the 9th International Workshop on Frontiers in Algorithmics (FAW)*, volume 9130 of *LNCS*, pages 3–6. Springer, 2015. doi:10.1007/978-3-319-19647-3_1. 3
- [26] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. URL: <https://dl.acm.org/doi/10.5555/574848>. 3

- [27] Archontia C. Giannopoulou, Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. Uniform kernelization complexity of hitting forbidden minors. *ACM Transactions on Algorithms*, 13(3):35:1–35:35, 2017. doi:10.1145/3029051. 32
- [28] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774. 32
- [29] Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A near-optimal planarization algorithm. In *Proc. of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1802–1811. SIAM, 2014. doi:10.1137/1.9781611973402.130. 4
- [30] Gwenaël Joret, Christophe Paul, Ignasi Sau, Saket Saurabh, and Stéphan Thomassé. Hitting and harvesting pumpkins. *SIAM Journal on Discrete Mathematics*, 28(3):1363–1390, 2014. doi:10.1137/120883736. 4
- [31] Ken-ichi Kawarabayashi. Planarity allowing few error vertices in linear time. In *Proc. of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 639–648, 2009. doi:10.1109/FOCS.2009.45. 4
- [32] Ken-ichi Kawarabayashi and Yusuke Kobayashi. Linear min-max relation between the treewidth of an H -minor-free graph and its largest grid minor. *Journal of Combinatorial Theory, Series B*, 141:165–180, 2020. doi:10.1016/j.jctb.2019.07.007. 5, 24
- [33] Ken-ichi Kawarabayashi, Robin Thomas, and Paul Wollan. A new proof of the flat wall theorem. *Journal of Combinatorial Theory, Series B*, 129:204–238, 2018. doi:10.1016/j.jctb.2017.09.006. 5, 8, 10, 11, 13
- [34] Ken-ichi Kawarabayashi and Paul Wollan. A Shorter Proof of the Graph Minor Algorithm: The Unique Linkage Theorem. In *Proc. of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 687–694, 2010. doi:10.1145/1806689.1806784. 17
- [35] Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. *ACM Transactions on Algorithms*, 12(2):21:1–21:41, 2016. doi:10.1145/2797140. 4, 6
- [36] Eun Jung Kim, Maria J. Serna, and Dimitrios M. Thilikos. Data-compression for parametrized counting problems on sparse graphs. In *Proc. of the 29th International Symposium on Algorithms and Computation (ISAAC)*, volume 123 of *LIPICs*, pages 20:1–20:13, 2018. doi:10.4230/LIPICs.ISAAC.2018.20. 6
- [37] Tomasz Kociumaka and Marcin Pilipczuk. Faster deterministic feedback vertex set. *Information Processing Letters*, 114(10):556–560, 2014. doi:10.1016/j.ipl.2014.05.001. 4
- [38] Tomasz Kociumaka and Marcin Pilipczuk. Deleting Vertices to Graphs of Bounded Genus. *Algorithmica*, 81(9):3655–3691, 2019. doi:10.1007/s00453-019-00592-7. 4

- [39] Alexandr V. Kostochka. Lower bound of the Hadwiger number of graphs by their average degree. *Combinatorica*, 4:307–316, 1984. doi:10.1007/BF02579141. 6
- [40] John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4. 3
- [41] Daniel Lokshantov. Wheel-Free Deletion Is $W[2]$ -Hard. In *Proc. of the 3rd International Workshop on Parameterized and Exact Computation (IWPEC)*, volume 5018 of *LNCS*, pages 141–147, 2008. doi:10.1007/978-3-540-79723-4_14. 3
- [42] Dániel Marx and Ildikó Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3-4):807–822, 2012. doi:10.1007/s00453-010-9484-z. 4, 5, 26
- [43] Rolf Niedermeier. *Invitation to fixed parameter algorithms*, volume 31. Oxford University Press, 2006. doi:10.1093/ACPROF:OSO/9780198566076.001.0001. 3
- [44] Ljubomir Perkovic and Bruce A. Reed. An improved algorithm for finding tree decompositions of small width. *International Journal of Foundations of Computer Science*, 11(3):365–371, 2000. doi:10.1142/S0129054100000247. 5, 24
- [45] Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004. doi:10.1016/j.orl.2003.10.009. 5, 20
- [46] Neil Robertson and Paul D. Seymour. Graph Minors. XIII. The Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995. doi:10.1006/jctb.1995.1006. 4, 5, 10, 14, 16
- [47] Neil Robertson and Paul D. Seymour. Graph Minors. XX. Wagner’s conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004. doi:10.1016/j.jctb.2004.08.001. 4
- [48] Neil Robertson and Paul D. Seymour. Graph Minors. XXI. Graphs with unique linkages. *Journal of Combinatorial Theory, Series B*, 99(3):583–616, 2009. doi:10.1016/j.jctb.2008.08.003. 17
- [49] Neil Robertson and Paul D. Seymour. Graph Minors. XXII. Irrelevant vertices in linkage problems. *Journal of Combinatorial Theory, Series B*, 102(2):530–563, 2012. doi:10.1016/j.jctb.2007.12.007. 17
- [50] Neil Robertson, Paul D. Seymour, and Robin Thomas. Quickly excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 62(2):323–348, 1994. doi:10.1006/jctb.1994.1073. 23
- [51] Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. A more accurate view of the Flat Wall Theorem. Manuscript submitted for publication, 2021. arXiv:2102.06463. 5, 8, 10, 11, 13, 15, 16, 23, 24

- [52] Ignasi Sau, Giannos Stamoulis, and Dimitrios M. Thilikos. k -apices of minor-closed graph classes. I. Bounding the obstructions. Manuscript submitted for publication, 2021. [arXiv:2103.00882](#). [4](#), [5](#), [8](#), [11](#), [15](#), [16](#), [18](#), [19](#)
- [53] Roded Sharan. *Graph Modification Problems and their Applications to Genomic Research*. PhD thesis, Sackler Faculty of Exact Sciences, School of Computer Science, 2002. URL: <http://www.cs.tau.ac.il/thesis/thesis/Roded-Sharan-phd.pdf>. [3](#)
- [54] Dimitrios M. Thilikos. Graph minors and parameterized algorithm design. In *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *LNCS*, pages 228–256. Springer, 2012. doi:10.1007/978-3-642-30891-8_13. [16](#)
- [55] Andrew Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory, Series B*, 81(2):318–338, 2001. doi:10.1006/jctb.2000.2013. [6](#)