An FPT-Algorithm for Recognizing k-Apices of Minor-Closed Graph Classes

Ignasi Sau, Giannos Stamoulis, Dimitrios M. Thilikos

An FPT-Algorithm for Recognizing k-Apices of Minor-Closed Graph Classes

Ignasi Sau
LIRMM, Université de Montpellier, CNRS, France
ignasi.sau@lirmm.fr

Giannos Stamoulis
Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens, Greece
Inter-university Postgraduate Programme "Algorithms, Logic, and Discrete Mathematics" (ALMA), Athens, Greece
giannosstam@di.uoa.gr

Dimitrios M. Thilikos
LIRMM, Université de Montpellier, CNRS, France
sedthilk@thilikos.info

Abstract

Let $\mathcal{G}$ be a 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 prove that if $\mathcal{G}$ is minor-closed, then there is an algorithm that either returns a set S certifying that G is a k-apex of $\mathcal{G}$ or reports that such a set does not exist, in $2^{\mathsf{poly}(k)}n^3$ time. Here poly is a polynomial function whose degree depends on the maximum size of a minor-obstruction of $\mathcal{G}$, i.e., the minor-minimal set of graphs not belonging to $\mathcal{G}$. In the special case where $\mathcal{G}$ excludes some apex graph as a minor, we give an alternative algorithm running in $2^{\mathsf{poly}(k)}n^2$ time.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Graph modification problems, irrelevant vertex technique, graph minors, parameterized algorithms

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.95

Category Track A: Algorithms, Complexity and Games

Related Version The full version of this paper is permanently available at `https://arxiv.org/abs/2004.12692`.

Funding *Ignasi Sau*: Supported by projects DEMOGRAPH (ANR-16-CE40-0028) and ESIGMA (ANR-17-CE23-0010).
Dimitrios M. Thilikos: Supported by projects DEMOGRAPH (ANR-16-CE40-0028) and ESIGMA (ANR-17-CE23-0010). Supported by the Research Council of Norway and the French Ministry of Europe and Foreign Affairs, via the Franco-Norwegian project PHC AURORA 2019.

Acknowledgements We wish to thank the anonymous reviewers for their comments and remarks that improved the presentation of this paper. Moreover, we wish to thank Dániel Marx for his valuable comments and advise.

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, 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 [24]. As reported by Roded Sharan in [48], already in 1979, Garey and Johnson mentioned 18 different types of modification problems [25, Section A1.2]. For more on graph modification problems, see [9, 24], as well as the running survey in [13]. 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.
> **Question:** Find, if exists, a set $S \subseteq V(G)$ certifying that G is a 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 [39], 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 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 [14, 17, 20, 42].

Unfortunately, we cannot hope for the existence of FPT-algorithms for every graph class $\mathcal{G}$. Indeed, the problem is W-hard[1] for some classes $\mathcal{G}$ that are closed under induced subgraphs [40] 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 [11], or induced minors [18].

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 minors[2], or *minor-closed*. To see this, we define $\mathcal{G}_k$ as the class of the k-apices of $\mathcal{G}$, i.e., the yes-instances of VERTEX DELETION TO $\mathcal{G}$, and observe that if $\mathcal{G}$ is minor-closed then the same holds for $\mathcal{G}_k$, for every k. This, in turn, implies that for every k, $\mathcal{G}_k$ can be characterized by a set $\mathcal{F}_k$ of minor-minimal graphs not in $\mathcal{G}_k$; we call these graphs the *obstructions* of $\mathcal{G}_k$ and we know that they are finite because of the Robertson and Seymour theorem [46]. In other words, we know that the obstruction set of $\mathcal{G}_k$ is bounded by some function of k. Then one can decide whether a graph G belongs to $\mathcal{G}_k$ by checking whether G excludes all members of the obstruction set of $\mathcal{G}_k$, and this can be checked by using the FPT-algorithm in [45] (see also [19]).

As the Robertson and Seymour theorem [46] 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 the parametric dependence of the above FPT-algorithm is indeed a constructible function.

---

[1] Implying that an FPT-algorithm would result in an unexpected complexity collapse; see [17].
[2] A graph H is a *minor* of a graph G if it can be obtained from a subgraph of G by contracting edges.

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 says that, for every t, there is some 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 $c^k \cdot n^{\mathcal{O}(1)}$ time [22, 34]. Reducing the constant c in this running time has attracted research on particular problems such as VERTEX COVER [12] (with $c = 1.2738$), FREEDBACK VERTEX SET [36] (with $c = 3.619$), APEX-PSEUDOFOREST [10] (with $c = 3$), PATHWIDTH 1 VERTEX DELETION (with $c = 4.65$) [15] (see also [29] for further related results). The first step towards a parameterized algorithm for VERTEX DELETION TO $\mathcal{G}$ for cases where $\mathcal{G}$ has unbounded treewidth was done in [41] and later in [30] for the 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. [28]. These results were later extended by Kociumaka and Marcin Pilipczuk [37], 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 $2^{\mathcal{O}_g(k^2 \log k)} \cdot n^{\mathcal{O}(1)}$ step[3] algorithm.

Our results. In this paper we give an explicit FPT-algorithm for VERTEX DELETION TO $\mathcal{G}$ for *every* 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 of time $2^{\mathsf{poly}(k)} \cdot n^3$, 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 of time $2^{\mathsf{poly}(k)} \cdot n^2$, for some polynomial* poly *whose degree depends on $\mathcal{G}$.*

Very recently, Fomin et al. [23] gave an FPT-algorithm of $\mathcal{O}_{s,k}(n^4)$ time 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 contains 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. [23] 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 [23], 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 [23].

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. [44]. 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 then run the algorithm of Lemma 11 (whose proof uses [1, 3, 31, 43]) that either reports that we have a no-instance, or concludes that the

---

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

[4] The definition is as minors, except that only edges incident to degree-two vertices are contracted.

treewidth of G is polynomially bounded by k, or finds a large wall R in G. In the second case, we use the main algorithmic result of Baste et al. [5] (Proposition 3) to solve the problem parameterized by treewidth, achieving the claimed running time. In the latter case, we apply Proposition 12 (whose proof uses [8, 32, 33]) to find in R a large *flat* wall W together with an apex set A. We find in W a packing of an appropriate number of pairwise disjoint large enough subwalls. Two possible scenarios may occur. If the "interior" of each of these subwalls has enough neighbors in the set apex $S \cup A$, we apply a combinatorial result (Lemma 15) that guarantees that every possible solution should intersect $S \cup A$, and we can branch on it. On the other hand, if there exists a subwall whose interior W has few neighbors in $S \cup A$, we argue that we can define from it a *flat* wall in which we can apply the irrelevant vertex technique of Robertson and Seymour [45] (Lemma 14). We stress that this flat subwall is not precisely a subwall of W but a tiny "tilt" of a subwall of W, a new concept that is necessary for our proofs. The application of the irrelevant vertex technique requires a lot of technical care. For this, we use and enhance some of the ingredients introduced by Baste et al. [5].

In order to achieve the improved running time claimed in Theorem 2, we do *not* use iterative compression. Instead, we directly invoke Lemma 11. If the treewidth is small, we proceed as above. If a large wall is found, we apply Proposition 12 and we now distinguish two cases. If a large flat wall whose flaps have bounded treewidth is found, we find an irrelevant vertex using Lemma 14. Otherwise, inspired by an idea of Marx and Schlotter [41], 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. We provide in Section 2 some definitions and preliminary results. In Section 3 we state several algorithmic and combinatorial results that will be used when finding an irrelevant vertex and when applying the branching argument discussed above. In Section 4 we present the algorithms claimed in Theorem 1 and Theorem 2. We conclude in Section 5 with some directions for further research. All the missing proofs are available in the full version of the paper [47].

2 Definitions and preliminary results

Before we explain our techniques, we give some necessary definitions. They concern fundamental tools from the Graph Minors series of Robertson and Seymour that are heavily used in our algorithms and proofs. But first, we restate the problem in a more convenient way.

2.1 Restating the problem

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

Let $\mathcal{G}$ be a minor-closed graph class and $\mathcal{F}$ be the set of its minor-obstructions. Clearly, Vertex Deletion to $\mathcal{G}$ is the same problem as asking, given a graph G and some $k \in \mathbb{N}$, whether G contains a vertex set S of at most k vertices such that $\mathcal{F} \nleq_{\mathsf{m}} G \setminus S$. Following the terminology of [4–7, 22, 23, 34, 35], we call this problem $\mathcal{F}$-M-Deletion. In order to prove Theorem 1, we apply the iterative compression technique (introduced in [44]; see also [14]) and we give a $2^{\mathsf{poly}(k)} \cdot n^2$ 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 $\mathcal{F} \nleq_{\mathsf{m}} G \setminus S$.
Objective: Find, if exists, a set $S' \subseteq V(G)$ of size at most k such that $\mathcal{F} \nleq_{\mathsf{m}} G \setminus S'$.

Some conventions. In what follows we always denote by $\mathcal{F}$ the obstruction set of the minor-closed class $\mathcal{G}$ of the instantiation of VERTEX DELETION TO $\mathcal{G}$ that we consider. Also, 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. 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 $\ell_{\mathcal{F}} = \max\{|E(H)| + |V(H)| \mid H \in \mathcal{F}\}$. We also agree that n is the size of the input graph G. 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 $\mathcal{F}$-minor free graphs are sparse [38, 50]).

We present here the main result of [5]. We will use this in order to solve $\mathcal{F}$-M-DELETION on instances of bounded treewidth.

▶ **Proposition 3.** *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 on n vertices and 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 $\mathcal{F} \nleq_{\mathsf{m}} G \setminus S$. This algorithm runs in $2^{\mathcal{O}_{s_{\mathcal{F}}}(\mathsf{tw} \log \mathsf{tw})} \cdot n$ time.*

2.2 Definitions

We give here a minimal set of definitions and concepts that are necessary to support the description of our results. Some of them are given precisely, and for some of them we just provide enough intuition.

Renditions. Let Δ be a closed disk, i.e., a set homeomorphic to the set $\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$. Given a subset X of Δ, we denote its closure by $\bar{X}$ and its boundary by $\mathbf{bor}(X)$. A Δ-*painting* is a pair $\Gamma = (U, N)$ where N is a finite set of points of Δ, $N \subseteq U \subseteq \Delta$, $U \setminus N$ has finitely many arcwise-connected components, called *cells*, such that, for every cell c, $\bar{c}$ is a closed disk, $\mathbf{bor}(c) \cap \Delta \subseteq N$, and $|\mathbf{bor}(c) \cap N| \leq 3$. We use the notation $U(\Gamma) := U$, $N(\Gamma) := N$ and denote the set of cells of Γ by $C(\Gamma)$. Notice that, given a Δ-painting Γ, the pair $(N(\Gamma), \{c \cap N \mid c \in C(\Gamma)\})$ is a hypergraph whose hyperedges have cardinality at most three, and Γ can be seen as a plane embedding of this hypergraph in Δ. 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) $\pi : N(\Gamma) \to V(G)$ 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 $c \in C(\Gamma)$, $\pi(c \cap N) \subseteq V(\sigma(c))$,
(4) for every cell $c \in C(\Gamma)$, $V(\sigma(c)) \cap \bigcup_{c' \in C(\Gamma) \setminus \{c\}} V(\sigma(c')) \subseteq \pi(c \cap N)$, and
(5) $\pi(N(\Gamma) \cap \mathbf{bor}(\Delta)) = V(\Omega)$, such that the points in $N(\Gamma) \cap \mathbf{bor}(\Delta)$ appear in $\mathbf{bor}(\Delta)$ in the same ordering as their images, via π, in Ω.

We say that an Ω-rendition (Γ, σ, π) of G is *tight* if the following conditions are satisfied:

(1) For every $c \in C(\Gamma)$, the graph $\sigma(c) \setminus \pi(c \cap N)$, when non-null, is connected and the neighborhood of its vertex set in G is $\pi(c \cap N)$.
(2) For every $c \in C(\Gamma)$ there are $|c \cap N|$ vertex-disjoint paths in G from $\pi(c \cap N)$ to the set $V(\Omega)$.
(3) If there are two points x, y of N such that $e = \{\pi(x), \pi(y)\} \in E(G)$, then there is a $c \in C(\Gamma)$ such that $\sigma(c)$ is the two-vertex connected graph $(e, \{e\})$.

It is easy to see that given an Ω-rendition of a graph G where $V(\Omega)$ contains at least three vertices that are in a cycle of G, a tight Ω-rendition of G can be constructed in $\mathcal{O}(n + m)$ steps.

Walls. We avoid here the detailed definition of an r-wall. As an intuitive alternative we provide the graph G in Figure 1 where an *elementary* 7*-wall* $\hat{W}$ is the graph with red and green vertices that has as edges the vertical and horizontal segments between a red and a green vertex. The 7*-wall* W is the spanning subgraph of G that is a subdivision of $\hat{W}$ with

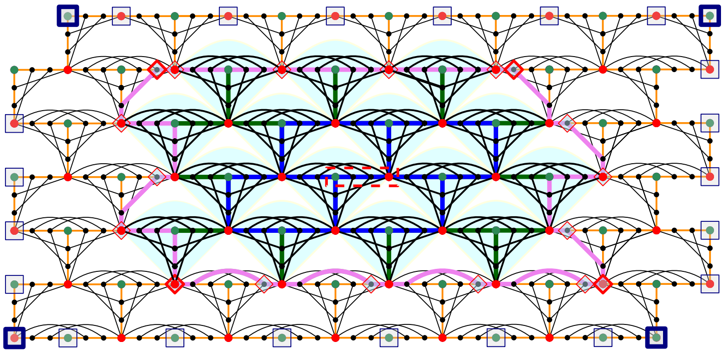

Figure 1 A graph G with a 7-wall and a flat 5-wall in it.

the black vertices as subdivision vertices. The *pegs* of $\hat{W}$ are depicted by the squared vertices, while the *corners* of $\hat{W}$ are the endpoints of the highest and the lowest horizontal path of $\hat{W}$, depicted by the fat squared vertices (the corners are also pegs). Notice that a wall W can occur in several ways from the elementary wall $\hat{W}$, depending on the way the vertices in the perimeter of $\hat{W}$ are subdivided. Each of them gives a different selection (P, C) of pegs and corners of W. We insist that, for every r-wall, the number r is always odd: for this, whenever an r-wall appears with r even, we agree to round it up to the next odd $r + 1$.

Flat walls. Given a graph G, 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 $R \setminus L$. An r-wall has a planar embedding where the boundary/ies of its external/internal face/es define its *perimeter* and its *bricks*. The *center* of the wall W is the path between a red and a green vertex depicted in the dashed red rectangle in Figure 1. Let G be a graph and let W be an r-wall of G. 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 $G[Y]$.

A subwall of W is every subgraph W' of W that is located "orthocanonically" in W and the wall $W^{(r)}$ is the unique r-subwall of W with the same center as W. A subwall is called *internal* if it does not intersect the perimeter of W. In Figure 1, W contains only one internal 5-subwall $W' = W^{(5)}$ and many internal 3-subwalls, among them the wall $W''' = W^{(3)}$ (depicted in blue). Of course, the graph G in Figure 1 contains also other walls as subgraphs such as the wall W'' consisting of the purple, green, and blue edges.

Compass and flaps. Given a flat wall W of a graph G as above, we call $G[Y]$ *the compass* of W in G, denoted by $\mathsf{compass}(W)$. We call (X, Y) the *separation certifying the flat wall* W and $X \cap Y$ is called the *frontier* of W. The *ground set* of W is $\mathsf{ground}(W) := \pi(N(\Gamma))$. We clarify that $\mathsf{ground}(W)$ may consist of vertices of the compass of W that are not necessarily vertices of W (this is not the case in the simple example of Figure 1). We also call the

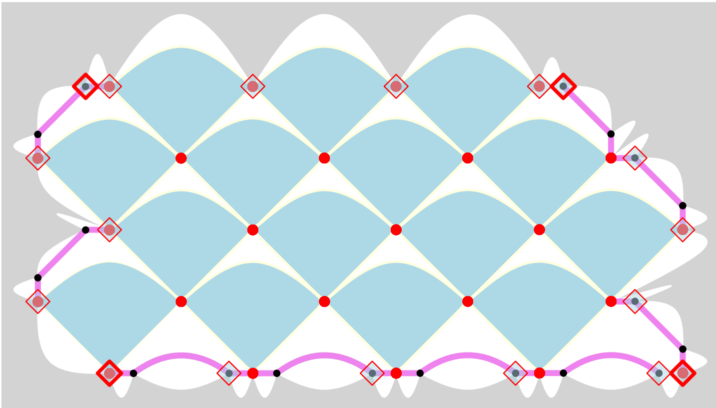

Figure 2 The rendition of the compass of the flat 5-wall W'' of Figure 1. Trivial flaps are depicted in purple.

graphs in $\mathsf{flaps}(W) := \{\sigma(c) \mid c \in C(\Gamma)\}$ *flaps* of the wall W. For each flap $F \in \mathsf{flaps}(W)$ we define its *base* as the set $\partial F := V(F) \cap \mathsf{ground}(W)$. A flap $F \in \mathsf{flaps}(W)$ is *trivial* if $|\partial F| = 2$ and it consists of one edge between the two vertices in ∂F. As an example, the wall W'' in Figure 1, formed by all the fat edges (purple, green, and blue), is a flat wall. The pegs are the diamond vertices, the corners are the fat diamond vertices, and the rendition has two types of flaps: those whose base has three vertices, that are inside the light-blue disks, and those that are trivial flaps and are the purple fat edges that are outside of the light-blue disks (see also Figure 2 for the rendition of W''). Notice that *none* of the internal subwalls of W is a flat wall.

Tilts. Given a wall W', we define its *inpegs* as the vertices of its perimeter that are incident to edges of W that are not in its perimeter. The *interior* of W' is the subgraph of W' induced by the union $V(W' \setminus V(P))$ and its inpegs. We say that a wall W'' is a *tilt* of a wall W' if W'' and W' have identical interiors. For instance, in Figure 1 the wall W'' is a tilt of $W' = W^{(5)}$.

Partially disk-embedded graphs. We say that a graph G is *partially disk-embedded in some closed disk* Δ, if there is some subgraph K of G that is embedded in Δ such that $(V(G) \cap \Delta, V(G) \setminus \mathsf{int}(\Delta))$ is a separation of G, where int is used to denote the interior of a subset of the plane. From now on, we use the term *partially* Δ*-embedded graph* G to denote that a graph G is partially disk-embedded in some closed disk Δ. We call the graph $K = G \cap \Delta$ *compass* of the Δ-embedded graph G and we assume that G is accompanied by an embedding of its compass in Δ, that is the set $G \cap \Delta$. We say that G is a Δ-embedded graph if it is partially Δ-embedded graph and $G \subseteq \Delta$ (the whole G is embedded in Δ).

Levelings. Let W be a flat wall of a graph G. Following [5], we define the *leveling* of W in G, denoted by $\tilde{W}$, as the bipartite graph where one part is the ground set of W, the other part is the set of flaps of W, and, given a pair $(v, F) \in \mathsf{ground}(W) \times \mathsf{flaps}(W)$, the set $\{v, F\}$ is an edge of $\tilde{W}$ if and only if $v \in \partial F$. Again, keep in mind that $\tilde{W}$ may contain (many) vertices that are not in W. Notice that the incidence graph of the plane hypergraph $(N(\Gamma), \{c \cap N \mid c \in C(\Gamma)\})$ is isomorphic to $\tilde{W}$ via an isomorphism that extends π and, moreover, bijectively corresponds cells to flaps. This permits us to treat $\tilde{W}$ as a Δ-embedded graph where $\mathbf{bor}(\Delta) \cap \tilde{W}$ is the frontier of W. We call the vertices of $\mathsf{ground}(W)$ (resp. $\mathsf{flaps}(W)$) *ground-vertices* (resp. *flap-vertices*) of $\tilde{W}$. See Figure 3 for an example of a leveling.

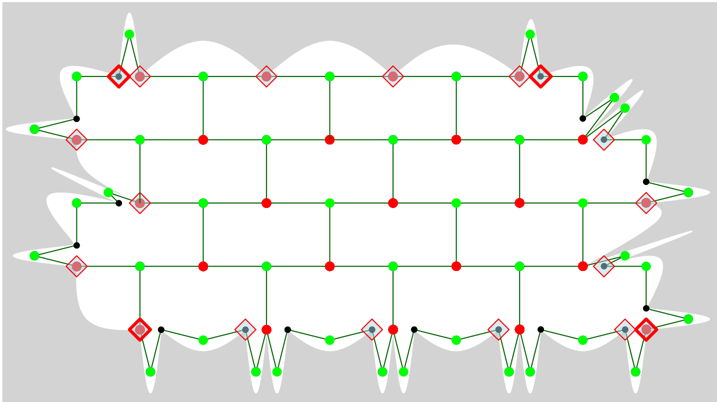

Figure 3 The leveling of the flat 5-wall W'' of Figure 1. The green vertices are the flap-vertices and the non-green vertices are the ground-vertices.

Recall that each edge e of $\mathsf{compass}(W)$ belongs to exactly one flap of W. If both of the endpoints of e are in the boundary of this flap, then this flap should be a trivial one and we say that e is a *short* edge of $\mathsf{compass}(W)$. We define the graph $W^\bullet$ as the graph obtained from W if we subdivide once every short edge in W. The next observation is a consequence of the following three facts: flap-vertices of $\tilde{W}$ have degree at most three, all the vertices of a wall have degree at most three, and every separation (A, B) of order at most three of a wall is trivial.

▶ **Observation 4.** *If W is a flat wall of a graph G, then the leveling $\tilde{W}$ of W in G contains a subgraph W^R that is isomorphic to some subdivision of $W^\bullet$ via an isomorphism that maps each ground vertex to itself.*

We call the graph W^R as in Observation 4 *representation* of the flat wall W in the Δ-embedded graph $\tilde{W}$, and therefore we can see it as a Δ-embedded subgraph of $\tilde{W}$. Notice that the above observation permits to bijectively map each cycle of W to a cycle of W^R that is also a cycle of $\tilde{W}$. That way, each cycle C of W corresponds to a cycle C of W^R denoted by C^R and we call C^R the *representation* of C in $\tilde{W}$. From now on, we reserve the superscript "^{R}"-notation to denote the correspondence between W (resp. C) and W^R (resp. C^R).

We define the function $\mathsf{flaps} : \mathcal{C}(W) \to 2^{\mathsf{flaps}(W)}$ so that, for each cycle C of W, $\mathsf{flaps}(C)$ contains each flap F of W that, when seen as a flap-vertex of the Δ-embedded graph $\tilde{W}$, belongs to the closed disk bounded by C^R. The following result is very similar to [33, Lemma 6.1]. The proof is strongly based on the notion of levelings.

▶ **Lemma 5.** *Let $a, r, r' \in \mathbb{N}$, where $r > r' \geq 3$. Also, let G be a graph, let (A, W) be an (a, r)-apex wall pair of G, and let W' be an internal r'-subwall of W. Then W' has a tilt W'' such that (A, W'') is an (a, r')-apex wall pair of G. Moreover,*

1. *the compass of W'' in $G \setminus A$ is a subgraph of the compass of W in $G \setminus A$ and*
2. *if P' is the perimeter of W', then the vertex set of the compass of W'' in $G \setminus A$ is a subset of $\bigcup \mathsf{flaps}(P')$.*

Morever, given G, (A, W), and W', the (a, r')-apex wall pair (A, W'') can be constructed in $\mathcal{O}(n)$ time.

From now on we refer to an (A, W'') as in Lemma 5 as *an* (a, r')-apex wall pair *generated by* the internal r'-subwall W' of W and we keep in mind that the compasses of all such flat walls W'' may differ *only* on their perimeter. The proof of Lemma 5 also implies the following.

▶ **Observation 6.** *Let W be a flat wall of a graph G, and W^R be the representation of W in the leveling $\tilde{W}$ of W in G. Then for every internal subwall $\bar{W}$ of W^R there exist an internal subwall W' of W and a tilt W'' of W' such that*
- *$\bar{W}$ is the representation of W' in the leveling $\tilde{W}$ of W,*
- *W'' is a flat wall, and*
- *the vertex set of the compass of W'' in G is a subset of $\bigcup \mathsf{flaps}(P')$, where P' is the perimeter of W'.*

Moreover, given G, W, and $\bar{W}$, the flat wall W'' can be constructed in $\mathcal{O}(n)$ steps.

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 $\rho : B \to [t]$ 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 [45], we define the *detail* of a boundaried graph $\mathbf{G} = (G, B, \rho)$ as $\mathsf{detail}(\mathbf{G}) := \max\{|E(G)|, |V(G) \setminus B|\}$. We denote by $\mathcal{B}^{(t)}$ the set of all (pairwise non-isomorphic) t-boundaried graphs. We also set $\mathcal{B} = \bigcup_{t \in \mathbb{N}} \mathcal{B}^{(t)}$.

Folios. We say that a t-boundaried graph $\mathbf{G}_1 = (G_1, B_1, \rho_1)$ is a *minor* of a t-boundaried graph $\mathbf{G}_2 = (G_2, B_2, \rho_2)$, denoted by $\mathbf{G}_1 \preceq_{\mathsf{m}} \mathbf{G}_2$, if there is a sequence of removals of non-boundary vertices, edge removals, and edge contractions in G_2, disallowing contractions of edges with both endpoints in B_2, that transforms $\mathbf{G}_2$ to a boundaried graph that is isomorphic to $\mathbf{G}_1$ (during edge contractions, boundary vertices prevail). Note that this extends the usual definition of minors in graphs without boundary.

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 $\mathsf{diss}(M, T)$ 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 if M is a subgraph of G. We call the vertices in T *branch* vertices of (M, T).

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 $\mathsf{diss}(\mathbf{M}, T) = (\mathsf{diss}(M, T), B, \rho)$. 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 *$\mathsf{btm}$-pair* of $\mathbf{G} = (G, B, \rho)$. Let $\mathbf{G}_i = (G_i, B_i, \rho_i), i \in [2]$. We say that $\mathbf{G}_1$ is a *topological minor* of $\mathbf{G}_2$, denoted by $\mathbf{G}_1 \preceq_{\mathsf{tm}} \mathbf{G}_2$, if $\mathbf{G}_2$ has a btm-pair $(\mathbf{M}, T)$ such that $\mathsf{diss}(\mathbf{M}, T)$ is isomorphic to $\mathbf{G}_1$. We define the *ℓ-folio* of $\mathbf{G} = (G, B, \rho) \in \mathcal{B}$ as $\ell\text{-}\mathsf{folio}(\mathbf{G}) = \{\mathbf{G}' \in \mathcal{B} \mid \mathbf{G}' \preceq_{\mathsf{tm}} \mathbf{G}$ and $\mathbf{G}'$ has detail at most $\ell\}$.

Homogeneous walls. Let G be a graph and W be a flat wall of G. Let also (Γ, σ, π) be a rendition of the compass of W in G. Recall that $\Gamma = (U, N)$ is a Δ-painting for some closed disk Δ. Given a flap F, we denote by $\Omega(F)$ the counter-clockwise ordering of the vertices of ∂F as they appear in the corresponding cell of $C(\Gamma)$. Notice that as $|\partial F| \leq 3$, 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., $(v_1, v_2, v_3) \equiv (v_2, v_3, v_1)$ but not under inversion, i.e., $(v_1, v_2, v_3) \not\equiv (v_3, v_2, v_1)$.

Given a graph G, we say that the pair (A, W) is an *(r, a)-apex wall pair* of G if A is a subset of a vertices from G and W is a flat r-wall of $G \setminus A$. Let G be a graph and let (A, W) be an (a, r)-apex wall pair of G. For each cell $F \in \mathsf{flaps}(W)$ with $t_F = |\partial F|$, we fix $\rho_F : \partial F \to [a+1, a+t_F]$ such that $(\rho_F^{-1}(a+1), \ldots, \rho_F^{-1}(a+t_F)) \equiv \Omega(c)$. We also fix a bijection $\rho_A : A \to [a]$. For each flap $F \in \mathsf{flaps}(W)$, we define the boundaried graph $\mathbf{F}^A := (G[A \cup F], A \cup \partial F, \rho_A \cup \rho_F)$ and we denote by F^A the underlying graph of $\mathbf{F}^A$. We call $\mathbf{F}^A$ *augmented flap* of (A, W). Notice that $G[V(\mathsf{compass}(W)) \cup A] = \bigcup_{F \in \mathsf{flaps}(W)} F^A$.

Given some $\ell \in \mathbb{N}$, we say that two flaps $F_1, F_2 \in \mathsf{flaps}(W)$ are (A,ℓ)-*equivalent*, denoted by $F_1 \sim_{A,\ell} F_2$, if $\ell\text{-}\mathsf{folio}(\mathbf{F}_1^A) = \ell\text{-}\mathsf{folio}(\mathbf{F}_2^A)$. For each $F \in \mathsf{flaps}(W)$, we define the (a,ℓ)-*color* of F, denoted by $(a,\ell)\text{-}\mathsf{color}(F)$, as the equivalence class of $\sim_{A,\ell}$ to which $\mathbf{F}^A$ belongs.

Let $\tilde{W}$ be the leveling of W in $G \setminus A$ and let W^R be the representation of W in $\tilde{W}$. Recall that $\tilde{W}$ is a Δ-embedded graph. For each cycle C of W, we define the (a,ℓ)-*palette* of C, denoted by $(a,\ell)\text{-}\mathsf{palette}(C)$, as the set of all the (a,ℓ)-colors of the flaps that appear as vertices of $\tilde{W}$ in the closed disk bounded by C^R in Δ (recall that by C^R we denote the representation of C in $\tilde{W}$).

Let $a, \ell, r, q \in \mathbb{N}$, where $r > q \geq 3$ and let (A, W) be an (a,r)-apex wall pair of a graph G. We say that (A, W) is an (ℓ, q)-*homogeneous* (a,r)-apex wall pair of G if every internal brick B of W that is not a brick of $W^{(q)}$ has the same (a,ℓ)-palette (seen as a cycle of W). If we drop the demand that "B is not a brick of $W^{(q)}$" then we simply say that (A, W) is an ℓ-*homogeneous* (a,r)-apex wall pair of G.

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

▶ **Observation 7.** *Let $a, r, r' \in \mathbb{N}$, where $r > r' \geq 3$. Also, let G be a graph, let (A, W) be an (a,r)-apex wall pair of G, and let W' be an internal r'-subwall of W. If (A, W) is (ℓ, q)-homogeneous for some $\ell, q \in \mathbb{N}$ where $r > q \geq 3$, then every (a, r')-apex wall pair (A, W'') generated by W' is (ℓ, q)-homogeneous.*

The following result is from [5, Lemma 4.3] and implies that if the wall of an apex wall pair is polynomially large on r, then its compass contains a homogeneous flat r-wall.

▶ **Proposition 8.** *There is a function $f_1 : \mathbb{N}^3 \to \mathbb{N}$ such that if $\ell, r, a \in \mathbb{N}$, where $r \geq 3$, G is a graph, and (A, W) is an $(a, f_1(\ell, r, a))$-apex wall pair of G, then W has a r-subwall W' such that every (a,r)-apex wall pair of G that is generated by W' is ℓ-homogeneous. Moreover, it holds that $f_1(\ell, r, a) = \mathcal{O}(r^{c_{a,\ell}})$, for some constant $c_{a,\ell}$ depending on a and ℓ.*

We refer to the constant $c_{a,\ell}$ of Proposition 8, when $a = a_{\mathcal{F}}$ and $\ell = \ell_{\mathcal{G}}$ as the *palette-variety* of $\mathcal{F}$. This constant reflects the *price of homogeneity*: the degree of the polynomial overhead that we need to pay in order to force homogeneity in a flat wall.

Treewidth. A *tree decomposition* of a graph G is a pair (T, χ) where T is a tree and $\chi : V(T) \to 2^{V(G)}$ such that (1) $\bigcup_{t \in V(T)} \chi(t) = V(G)$, (2) for every edge e of G there is a $t \in V(T)$ such that $\chi(t)$ contains both endpoints of e, and (3) 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 defined as $\mathbf{w}(T, \chi) := \max\big\{ |\chi(t)| - 1 \;\big|\; t \in V(T)\big\}$. The *treewidth of* G is defined as $\mathsf{tw}(G) := \min\big\{ \mathbf{w}(T, \chi) \;\big|\; (T, \chi) \text{ is a tree decomposition of } G\big\}$.

The following is the main result of [8]. We will use it to compute a tree decomposition of a graph of bounded treewidth.

▶ **Proposition 9.** *There is an algorithm that, given an graph G on n vertices and an integer k, it outputs either a report that $\mathsf{tw}(G) > k$, or a tree decomposition of G of width at most $5k + 4$. Moreover, this algorithm runs in $2^{\mathcal{O}(k)} \cdot n$ steps.*

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

▶ **Proposition 10.** *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 $2^{\mathcal{O}(k \log k)} \cdot h^{\mathcal{O}(k)} \cdot 2^{\mathcal{O}(h)} \cdot m$ steps.*

3 Auxiliary algorithmic and combinatorial results

3.1 Two algorithmic results

We are now in position to state some results that will support our algorithms.

▶ **Lemma 11.** *There exist a function $f_2 : \mathbb{N} \to \mathbb{N}$ and an algorithm as follows:*
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 $f_2(s_{\mathcal{F}}) \cdot r + k$, 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 $2^{\mathcal{O}_{s_{\mathcal{F}}}(r^2+(k+r)\log(k+r))} \cdot n$ steps.

The proof of Lemma 11 combines the algorithm of Perkovič and Reed in [43] for computing the treewidth of a graph, as well as the excellent analysis of the algorithm provided in [3]. We also use the upper bound for the treewidth of an K_h-minor free graph without an r-wall by [31], the dynamic programming algorithm of [1] for finding a wall in a graph of bounded treewidth, and the single-exponential FPT-approximation algorithm for treewidth in [8].

The next result follows from [33, Theorem 1.9] and the proof of [32, Theorem 5.2].

▶ **Proposition 12.** *There are functions $f_3 : \mathbb{N} \to \mathbb{N}$, $f_4 : \mathbb{N} \to \mathbb{N}$ and an algorithm as follows:*
Clique-Or-Flat-Wall(G, r, t, W)
Input: *A graph G on n vertices and m edges, an odd integer $r \geq 3$, a $t \in \mathbb{N}_{\geq 1}$, and an R-wall W in G, where $R = f_3(t) \cdot r$.*
Output: *Either a minor of G isomorphic to K_t, or*
(1) *a set $A \subseteq V(G)$ of size at most $12288t^{24}$,*
(2) *a flat r-wall $\bar{W}$ of $G \setminus A$ such that $V(\bar{W}) \cap A = \emptyset$, and*
(3) *a separation (X, Y) of $G \setminus A$ that certifies that $\bar{W}$ is a flat wall and an Ω-rendition of $G[Y]$ with flaps of treewidth at most $f_4(t) \cdot r$, where Ω is a cyclic ordering of $X \cap Y$ determined by the order on the perimeter of $\bar{W}$.*

Moreover, this algorithm runs in $2^{\mathcal{O}_t(r)}(m + n)$ time.

Proposition 12, without the bound on the treewidth of the flaps, has been proven in [33, Theorem 1.9]. However, in [33, Theorem 1.9] $\bar{W}$ is a tilt of some r-subwall of W with a different function $f_3{}'$ for the relation between R and r and has running time $\mathcal{O}(t^{24}(n+m))$.

In Proposition 12 the bound on the treewidth of the flaps is obtained if we plug [33, Theorem 1.9] in the proof of [32, Theorem 5.2], taking into account the linear dependence between R and r. The parameterized dependence $2^{\mathcal{O}_t(r)}$ of the algorithm follows because of the use of the linear FPT-approximation algorithm for treewidth in [8] so as to compute the tree decompositions of the flaps. Another stronger version of Proposition 12, where no R-wall W is given in the input, appears in [23, Lemma 3.2], running in $2^{\mathcal{O}_t(r^{58})} \cdot n \log^2 n$ time. We do not need this stronger version as, for our problem, the R-wall W will be found by the algorithm of Lemma 11.

3.2 Finding an irrelevant vertex

The *irrelevant vertex technique* was introduced in [45] for providing an FPT-algorithm for the Disjoint Paths problem. Moreover, this technique has appeared to by quite versatile and is now a standard tool of parameterized algorithm design (see e.g., [14, 49]). The applicability of this technique for $\mathcal{F}$-M-Deletion is materialized by the algorithm of Lemma 14.

Given a graph G, a set $A \subseteq V(G), |A| = a$, and an r-wall W of G, we say that (A, W) is an (a, r)-*apex-wall* pair if W is a flat r-wall of $G \setminus A$.

▶ **Lemma 13.** *There is a function $f_5 : \mathbb{N}^4 \to \mathbb{N}$ such that if $a, \ell, q, k \in \mathbb{N}$, $q \geq 3$, G is a graph, and (A, W) is an ℓ-homogeneous $(a, f_5(a, \ell, q, k))$-apex wall pair of G, then for every (a, q)-apex wall pair $(A, \hat{W})$ generated by $W^{(q)}$, it holds that (G, k) and $(G \setminus V(\mathsf{compass}(\hat{W})), k)$ are equivalent instances of $\mathcal{F}$-M-Deletion. Moreover, $f_6(a, \ell, q, k) = O_{a,\ell,q}(k)$.*

Sketch of the proof. The proof considers $k+1$ subwalls of W which, in turn, generate $k+1$ flat walls (by taking their tilts) whose compasses are all disjoint except at some territory T of constant size containing the center of W that is common for all these walls. If now S is a solution to $\mathcal{F}$-M-Deletion for the instance (G, k) then one, say W_i, of these $k+1$ flat walls will have a compass that does not intersect S (except from some constant size territory $T_i$around the center of W_i that contains T). We then claim that $S' = S \setminus T_i$ is a new solution of $\mathcal{F}$-M-Deletion that avoids the center of W. To prove that this is the case, we assume to the contrary that some graph L in $\mathcal{F}$ appears as a minor in $G' = G \setminus S'$. But this means that part of the realization of H in G' meets some vertex in the center of W and therefore it traverses $\mathsf{compass}(W_i)$. We arrive to a contradiction by providing an alternative realization of L that is routed away from T_i. For this rerouting we use the main combinatorial result of [5] that guarantees that there is a function $f_6 : \mathbb{N}^3 \to \mathbb{N}$ such that, for every $a, \ell, q \in \mathbb{N}$ and every graph G, if (A, W) is an (ℓ, q)-homogeneous $(a, f_6(a, \ell, q))$-apex wall pair of G, there is an (a, q)-apex wall pair of G generated by $W^{(q)}$ whose compass is irrelevant. For this it is enough to pick $f_5(a, \ell, q, k) = \mathcal{O}(k \cdot f_6(a, \ell, q) + q)$. ◀

▶ **Lemma 14.** *There exist a function $f_7 : \mathbb{N}^4 \to \mathbb{N}$ and an algorithm with the following specifications:*

Find-Irrelevant-Wall(G, q, k, b, A, W)
Input: *A graph G on n vertices, two integers $k, q \in \mathbb{N}$, a $b \in \mathbb{N}_{\geq 3}$, and an $(a, f_7(a, \ell_{\mathcal{F}}, b, k))$-apex wall pair (A, W) of G whose all flaps have treewidth at most q.*
Output: *A flat b-wall $\hat{W}$ of $G \setminus A$ such that (G, k) and $(G \setminus V(\mathsf{compass}(\hat{W})), k)$ are equivalent instances of $\mathcal{F}$-M-Deletion.*
Moreover, $f_7(a, \ell_{\mathcal{F}}, b, k) = \mathcal{O}_{a,\ell_{\mathcal{F}}}((k+b)^{c_{a,\ell_{\mathcal{F}}}})$ for some constant $c_{a,\ell_{\mathcal{F}}}$ depending on a and $\ell_{\mathcal{F}}$. This algorithm runs in $2^{\mathcal{O}_{a,\ell_{\mathcal{F}}}(q \log q + (k+b)\log(k+b))} \cdot n$ time.

Proof. We set $f_7(a, \ell_{\mathcal{F}}, b, k) := f_1(\ell_{\mathcal{F}}, r, a)$, where $r = f_5(a, \ell, b, k)$. The algorithm considers each one of the $\binom{f_7(a,\ell_{\mathcal{F}},b,k)}{r}$ internal r-subwalls W' of W and constructs an (a, r)-wall pair (A, W'') generated by W'. This can be done in $\mathcal{O}(n)$ time because of Lemma 5.

From Proposition 8 there is a choice of W' such that (A, W'') is $\ell_{\mathcal{F}}$-homogeneous. To check whether (A, W'') is $\ell_{\mathcal{F}}$-homogeneous we do the following. Let $\mathcal{B}$ be the set of all flaps of W'' that, when seen as flap vertices of $\tilde{W}$, appear in the closed disk bounded by the representation of B in $\tilde{W}$, where B is an internal brick of W'' that is not a brick of $W''^{(b)}$. For every flap $F \in \mathcal{B}$, we consider the boundaried graph $\mathbf{F}^A$. Using the fact that $\mathsf{tw}(F^A) \leq q + a$, we apply the algorithm of Proposition 9 which outputs a tree decomposition of F^A of width at most $5(q+a)+4$. Then by applying the algorithm of Proposition 10, we compute $\ell_{\mathcal{F}}$-$\mathsf{folio}(\mathbf{F}^A)$ in $2^{\mathcal{O}_{a,\ell_{\mathcal{F}}}(q \log q)}$ time. Then, it is easy to check in linear time whether (A, W'') is $\ell_{\mathcal{F}}$-homogeneous.

After we find W', we again use Lemma 5 in order to construct a flat b-wall $\hat{W}$ of $G \setminus A$ generated by $W'^{(b)}$. The algorithm outputs $\hat{W}$, and this is correct because of Lemma 13. ◀

3.3 Combinatorial results for branching

We now give a combinatorial result that will justify a branching step of our algorithm, i.e., its recursive application on a set of $\mathcal{O}_s(k)$ vertices. Given a graph G and a set $A \subseteq V(G)$, we say that a graph H is an *A-fixed minor of G* if H can be obtained from a subgraph G' of

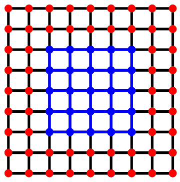

Figure 4 A 9-grid and its central 5-subgrid.

G where $A \subseteq V(G')$, after contracting edges without endpoints in A (see Figure 4 for the definition of an r-grid and its central subgrids that we will need later). A graph H is an *A-apex r-grid* if it can be obtained by an r-grid Γ after adding a set A of new vertices and some edges between the vertices of A and $V(\Gamma)$. We call Γ *underlying grid* of H.

Next we identify a combinatorial structure that guarantees the existence of a set of $q = \mathcal{O}_s(k)$ vertices that intersects every solution S of $\mathcal{F}$-M-Deletion on imput (G, k). This will permit branching on q simpler instances of the form $(G', k-1)$.

▶ **Lemma 15.** *There exist three functions $f_8, f_9, f_{10} : \mathbb{N}^2 \to \mathbb{N}$, such that if $\mathcal{F}$ is a finite set of graphs, G is a graph, $k \in \mathbb{N}$, and $A \subseteq V(G)$, $|A| = a_{\mathcal{F}}$, such that G contains as an A-fixed minor an A-apex $f_8(s_{\mathcal{F}}, k)$-grid H where each vertex $v \in A$ has at least $f_9(s_{\mathcal{F}}, k)$ neighbors in the central $(f_8(s_{\mathcal{F}}, k) - f_{10}(s_{\mathcal{F}}, k))$-grid of $H \setminus A$, then for every solution S of $\mathcal{F}$-M-Deletion for the instance (G, k), it holds that $S \cap A \neq \emptyset$. Moreover, $f_8(s_{\mathcal{F}}, k) = \mathcal{O}_{s_{\mathcal{F}}}(k^{3/2})$, $f_9(s_{\mathcal{F}}, k) = \mathcal{O}_{s_{\mathcal{F}}}(k^3)$, and $f_{10}(s_{\mathcal{F}}, k) = \mathcal{O}_{s_{\mathcal{F}}}(k)$.*

We conjecture that Lemma 15 is tight, in the sense that it cannot be proved for some $f_9(s_{\mathcal{F}}, k) = \mathcal{O}_{s_{\mathcal{F}}}(k^{3-\epsilon})$.

Notice that in the special case where $a_{\mathcal{F}} = 1$, then Lemma 15 can be improved by using the main combinatorial result of [16]. In particular [16, Lemma 3.1] easily implies that, in this case, $f_8(s_{\mathcal{F}}, k) = \mathcal{O}_{s_{\mathcal{F}}}(k)$, $f_9(s_{\mathcal{F}}, k) = \mathcal{O}_{s_{\mathcal{F}}}(k^2)$, and $f_{10}(s_{\mathcal{F}}, k) = \mathcal{O}_{s_{\mathcal{F}}}(\sqrt{k})$. In [47], we prove that these bounds can be improved to $f_8(s_{\mathcal{F}}, k) = \mathcal{O}_{s_{\mathcal{F}}}(\sqrt{k})$, $f_9(s_{\mathcal{F}}, k) = \mathcal{O}_{s_{\mathcal{F}}}(k)$, and $f_{10}(s_{\mathcal{F}}, k) = \mathcal{O}_{s_{\mathcal{F}}}(1)$. We will use these improved bounds for the proof of Theorem 2.

We conclude this subsection with one additional definition that will be useful for the application of Lemma 15 in our main algorithm.

Canonical partitions. We define the *canonical partition* of an r-wall W to be a collection $\mathcal{Q} = \{Q_{\text{ext}}, Q_1, \dots, Q_q\}$ of $(r-2)^2 + 1$ connected subgraphs of W such that their vertex sets form a partition of $V(W)$ as indicated in Figure 5.

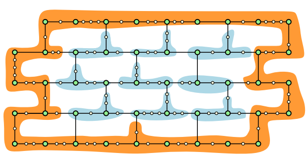

Figure 5 A 5-wall and its canonical partition $\mathcal{Q}$. The orange bag is the external bag Q_{ext}.

Let (A, W) be an (a, r)-apex wall pair of a graph G and let $\tilde{W}$ be the Δ-embedded graph that is the leveling $\tilde{W}$ of W in $G \setminus A$. Let also W^R be the representation of W in $\tilde{W}$. Consider a canonical partition $\mathcal{Q}$ of W^R. We enhance the graphs of $\mathcal{Q}$ so to include in them all the vertices of $\tilde{W}$ by applying the following procedure. We set $\tilde{\mathcal{Q}} := \mathcal{Q}$ and, as long as there is a vertex $x \in V(\tilde{W}) \setminus V(\bigcup \tilde{\mathcal{Q}})$[5] 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 $\tilde{Q} = \tilde{W}[\{x\} \cup V(Q)]$. We just defined a partition of the vertices of $\tilde{W}$ into subsets inducing connected graphs. We call such a partition *canonical partition* of $\tilde{W}$. Notice that a canonical partition of $\tilde{W}$ is not unique (since the sets in $\mathcal{Q}$ can be "expanded" arbitrarily when introducing the vertex x).

4 The algorithms

4.1 The general algorithm

▶ **Lemma 16.** *Let $\mathcal{F}$ be a finite collection of graphs. There is an algorithm solving $\mathcal{F}$-M-Deletion-Compression in $2^{\mathcal{O}_s(k^{2(c_{\mathcal{F}}+2)} \log k)} \cdot n^2$ time.*

Proof. For simplicity, in this proof, we use c instead of $c_{\mathcal{F}}$, s instead of $s_{\mathcal{F}}$, ℓ instead of $\ell_{\mathcal{F}}$, a instead of $a_{\mathcal{F}}$, and remember that $\ell = \mathcal{O}_s(1)$ and $a = \mathcal{O}_s(1)$. We set

$$b = f_7(k, a, \ell) = \mathcal{O}(k^c), \qquad x = f_9(s, k), \qquad l = (12288 s^{24} + k + 1) \cdot x,$$
$$m = f_8(s, k), \qquad p = f_{10}(s, k), \qquad h = \max\{m - p, \lceil \sqrt{l+1} \cdot b \rceil\},$$
$$r = h + p + 2, \qquad R = f_3(s) \cdot r, \qquad \text{and notice that } R = \mathcal{O}_s(k^{c+2}).$$

Recall that, in the definition of R, the constant c is the palette-variety of $\mathcal{F}$. 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 where $k' \leq k$, and S is a subset of $V(G)$ such that $|S| = k$ and $\mathcal{F} \not\leq_{\mathsf{m}} G \setminus S$. 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 `Find-Wall` of Lemma 11 with input $(G \setminus S, R, 0)$. This outputs, in $2^{\mathcal{O}_s(k^{2(c+2)})} \cdot n$ time, either a report that $\mathsf{tw}(G \setminus S) \leq f_2(s) \cdot R$, or an R-wall W_0 of $G \setminus S$. Notice that `Find-Wall`$(G \setminus S, R, s)$ never outputs the third case, since $(G \setminus S, 0)$ is a yes-instance of $\mathcal{F}$-M-Deletion. In the first possible output, we know that $\mathsf{tw}(G) \leq f_2(s) \cdot R + k = \mathcal{O}_s(k^{c+2})$, and we call the algorithm of Proposition 3 with input $(G, f_2(s) \cdot R + k, k')$ and return a correct answer in $2^{\mathcal{O}_s(k^{c+2} \log k)} \cdot n$ steps. In the second possible output, the algorithm moves to the second step.

Step 2. Call `Clique-Or-Flat-Wall` of Proposition 12 on $(G \setminus S, r, s, W_0)$. Since $\mathcal{F} \not\leq_{\mathsf{m}} G \setminus S$ and $\mathcal{F} \leq_{\mathsf{m}} K_s$, the algorithm outputs, in time $2^{\mathcal{O}_s(r)} \cdot (m + n) = 2^{\mathcal{O}_s(k^{c+2})} \cdot n$, the following:

- A set $A \subseteq V(G) \setminus S$ of size at most $12288 s^{24}$,
- a flat r-wall W of $G \setminus (S \cup A)$ such that $V(W) \cap A = \emptyset$, and
- a separation (X, Y) of $G \setminus A$ that certifies that W is a flat wall, and an Ω-rendition of $G[Y]$ with flaps of treewidth at most $q = f_4(s) \cdot r$, where Ω is a cyclic ordering of $X \cap Y$ determined by the order on the outer cycle of W.

Let $\tilde{W}$ be the leveling of W and let W^R be the representation of W in the Δ-embedded graph $\tilde{W}$ ($\tilde{W}$ and W^R can straighforwardly be constructed in $\mathcal{O}(n)$ steps using the Ω-rendition of $G[Y]$). Let also $\bar{W} = (W^R)^{(r-p)}$, i.e., $\bar{W}$ is the central $(r-p)$-subwall of W^R.

[5] If $\mathcal{S}$ is a collection of objects where the operation $\cup$ is defined, then we denote $\bigcup \mathcal{S} = \bigcup_{X \in \mathcal{S}} X$.

Consider a family $\bar{\mathcal{W}} = \{\bar{W}_1, \ldots, \bar{W}_{l+1}\}$ of $l+1$ internal b-subwalls of $\bar{W}$, such that if D_i is the closed disk in Δ bounded by the perimeter, denoted by $\bar{P}_i$, of $\bar{W}_i$, then $D_i \cap D_j = \emptyset$, for $i \neq j$. We are allowed to do this since $r - p - 2 \geq \lceil \sqrt{l+1} \cdot b \rceil$. By $\mathsf{flaps}(D_i)$ we denote all the flaps corresponding to flap-vertices of $\tilde{W}$ that are inside D_i in the embedding of $\tilde{W}$ in Δ. For every $i \in [l+1]$, we compute, in $\mathcal{O}(n)$ time, the set $A_i = \{v \in S \cup A \mid v \text{ is adjacent in } G \text{ to a vertex of } \bigcup \mathsf{flaps}(D_i)\}$ and we proceed to the last step.

Step 3. The algorithm examines two cases:

Case A: For every $i \in [l+1]$, it holds that $|A_i| > a$. In this case the algorithm recursively calls `Solve-Compression` with input $(G \setminus x, k'-1, |S \setminus x|, S \setminus x)$, for every $x \in S \cup A$, and if one of these new instances is a yes-instance, certified by a set $\bar{S}$, then return $\bar{S} \cup \{x\}$, otherwise return that (G, k') is a no-instance.

We now prove that the above branching step of the algorithm is correct. Let $\tilde{W}$ be the leveling of W in $G \setminus (S \cup A)$. We define $\tilde{G}$ as the graph obtained from $G \setminus (S \cup A)$ if we remove all the vertices of the compass of W and take the union of the resulting graph with $\tilde{W}$. Notice that $\tilde{G}$ is *partially* Δ*-embedded* in the sense that the part of G that is embedded in Δ is $\tilde{W}$. Notice that $\tilde{G}$ is not necessarily a contraction of $G \setminus (S \cup A)$, and this is because the trivial flaps appear in $\tilde{W}$ as induced paths of length two instead of edges. Therefore, if $\breve{G}$ is the graph obtained from $\tilde{G}$ after dissolving each flap-vertex corresponding to a trivial flap, then

- $\breve{G}$ is a contraction of $G \setminus (S \cup A)$,
- $\breve{G}$ is a partially Δ-embedded graph whose compass is a dissolution of $\tilde{W}$, and
- $\breve{G}$ contains an r-wall $\breve{W}$ that is a dissolution of W^R and $\breve{W}$ is embedded in Δ so that its perimeter is a dissolution of the perimeter of W^R.

Consider a canonical partition $\tilde{\mathcal{Q}}$ of $\tilde{W}$. Let $\breve{\mathcal{Q}}$ be the collection of connected subgraphs of $\breve{G}$ that are obtained if we apply to the graphs in $\tilde{\mathcal{Q}}$ the same dissolutions that we used to transform $\tilde{G}$ to $\breve{G}$ (we just take care that the edge contracted during each dissolution has both endpoints in some bag). Moreover, we enhance $\breve{\mathcal{Q}}$ by adding in its external bag all the vertices of $\breve{G}$ that are not points of Δ. Notice that the vertex sets of the graphs in this new $\breve{\mathcal{Q}}$ define a partition of $V(\breve{G})$. Let now $\breve{G}^+$ be the graph obtained if we apply in G the same contractions that transform $G \setminus (S \cup A)$ to its contraction $\breve{G}$. Let $a^* = |S \cup A| \leq 12288 s^{24} + k + 1$. We now construct a minor of $\breve{G}^+$ by contracting all edges of each member of $\breve{\mathcal{Q}}$ to a single vertex and removing the vertex to which the external bag was contracted. We denote the resulting graph by $\bar{G}$ and we observe that $\bar{G}$ contains as a spanning subgraph an $S \cup A$-apex $(r-2)$-grid Γ. Recall that Γ is a $S \cup A$-fixed minor of G. Let $\bar{\Gamma}$ be the underlying grid of Γ and let $\bar{\Gamma}_1, \ldots, \bar{\Gamma}_{l+1}$ be the "packing" of the h-central grid $\bar{\Gamma}'$ of $\bar{\Gamma}$, corresponding to the walls in $\bar{\mathcal{W}}$, where each $\bar{\Gamma}_i$ is a b-grid. We can assume the existence of this packing because $h \geq \lceil \sqrt{l+1} \cdot b \rceil$. The initial assumption that $|A_i| > a$, for $i \in [l+1]$, implies that $\forall i \in [l+1]$, there are more than a apices of Γ that are adjacent to vertices of $\bar{\Gamma}_i$.

For every $v \in S \cup A$, let N_v be the set of neighbors of v in $\bar{\Gamma}'$ and let $\bar{N} = \bigcup_{v \in S \cup A} N_v$. Let A^* be the set of vertices of $S \cup A$ with $|N_v| \geq x$. We claim that $|A^*| \geq a$. Suppose to the contrary that $|A^*| < a$. This implies that the vertices in $(S \cup A) \setminus A^*$ are adjacent to at most $x \cdot |(S \cup A) \setminus A^*| \leq l$ vertices in $\bar{N}$. This, in turn implies that there is an $i \in [l+1]$ such that there are no vertices in $(S \cup A) \setminus A^*$ adjacent to vertices of $\bar{\Gamma}_i$. Thus, for this i, there are at most a apex vertices of $\bar{G}$ that are adjacent to vertices of $\bar{\Gamma}_i$, a contradiction to the conclusion of the previous paragraph. We arbitrarily remove vertices from A^* so that $|A^*| = a$.

Consider now the A^*-apex $(r-2)$-grid $H = \Gamma \setminus ((S \cup A) \setminus A^*)$, and as each vertex in A^* has at least x neighbors in $V(\bar{\Gamma}')$ and $r - 2 \geq m$, Lemma 15 can be applied for k', A^*, H. This implies that (G, k') is a yes-instance of $\mathcal{F}$-M-Deletion if and only if there is some

$v \in A^*$ such that $(G \setminus v, k'-1)$ is a yes-instance of $\mathcal{F}$-M-Deletion. This completes the correctness of the branching step in Case A.

Case B: there is an $i \in [l+1]$ such that $|A_i| \leq a$. Since $\bar{W}_i$ is an internal b-subwall of W^R, there is a subgraph of $\mathsf{compass}(W)$ that is a flat b-wall W_i'' of $G \setminus (S \cup A)$ such that the set of vertices of the compass of W_i'' is a subset of $\bigcup \mathsf{flaps}(D_i)$ (as we argued in Subsection 3.2, W_i'' is a tilt of the subwall of W represented by $\bar{W}_i$ and can be found in $\mathcal{O}(n)$ time). This implies that if A_i'' is the set of vertices from $S \cup A$ that are adjacent with vertices of the compass of W_i'' in $G \setminus (S \cup A)$, then $A_i'' \subseteq A_i$. Thus (A'', W_i'') is an $(|A_i''|, b)$-apex wall pair in G.

We now apply `Find-Irrelevant-Wall` of Lemma 14 for $(G, k, q, 3, A_i'', W_i'')$ and pick a vertex v in the center of the obtained 3-wall. According to Lemma 14, it holds that (G, k) and $(G \setminus v, k)$ are equivalent instances of $\mathcal{F}$-M-Deletion, v can be detected in $2^{\mathcal{O}_{a,\ell}(k \log k)} \cdot n$ time, and the algorithm correctly calls recursively `Solve-Compression` with input $(G \setminus v, k', k, S)$.

Recall that $|S \cup A| \leq k + 1 + 12288 s^{24} = \mathcal{O}_s(k)$. Therefore, if $T(n, k', k)$ is the running time of the above algorithm, then $T(n, k', k) \leq 2^{\mathcal{O}_s(k^{2(c+2)})} n + \max\{T(n-1, k', k), \mathcal{O}_s(k) \cdot T(n, k-1, k)\}$ which, given that $k' \leq k$, implies that $T(n, k', k) = 2^{\mathcal{O}_s(k^{2(c+2)})} n^2$.

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

4.2 The apex-minor free case

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 $2^{\mathcal{O}_{s_{\mathcal{F}}}(k^{2(c+1)})} \cdot n^2$, where $c = c_{a,\ell_{\mathcal{F}}}$ and $a = 12288(s_{\mathcal{F}})^{24}$. The existence of such an algorithm implies Theorem 2.

Let G be graph and let W be an r-wall in G. The *drop*, denoted by $D_{W'}$, of a q-subwall W' of W, where $q \leq r$, is defined as follows: Contract in G the perimeter of W to a single vertex v. $D_{W'}$ is the unique biconnected component of the resulting graph that contains the interior of W'. We call the vertex v the *pole* of the drop $D_{W'}$.

The algorithm. Our algorithm avoids iterative compression in the fashion that this is done by Marx and Schlotter in [41] for the Planarization problem. The algorithm has three main steps. We first set $s = s_{\mathcal{F}}$, $a = 12288 s^{24}$, $b' = f_7(a, k, \ell_{\mathcal{F}}) = \mathcal{O}_s(k^{c_{a,\ell_{\mathcal{F}}}})$, and we define

$$b = f_3(s) \cdot 2b' + 2 \qquad l = f_9(s,k) \cdot k \qquad p = f_{10}(s,k)$$
$$h = \max\{f_8(s,k) - p, b \cdot \sqrt{l+1}\} \qquad r = h + p + 2 \qquad R = f_3{}'(s) \cdot r + k = \mathcal{O}_s(k^{c+1}).$$

Step 1. Run the algorithm `Find-Wall` of Lemma 11 with input (G, R, k) and, in $2^{\mathcal{O}_s(k^{2(c+1)})} \cdot n$ time, either report a no-answer, or conclude that $\mathsf{tw}(G) \leq f_2(s) \cdot R + k$ and solve $\mathcal{F}$-M-Deletion in $\mathcal{O}(2^{\mathcal{O}_s(k^{c+1}) \log k} \cdot n)$ time using the algorithm of Proposition 3, or obtain an R-wall $W^\bullet$ of G. In the third case, consider all the $\binom{R}{b} = 2^{\mathcal{O}_s(k^c \log k)}$ b-subwalls of W and for each one of them, say W', construct its drop $D_{W'}$, consider in $D_{W'}$ the central $(b-2)$-subwall $\bar{W}$ of W', and run the algorithm `Clique-Or-Flat-Wall` of Proposition 12 with input $D_{W'}$, $2b'$, s, and $\bar{W}$. This takes time $2^{\mathcal{O}_s(k^c)} \cdot n$. If for some of these drops the result is an $(|A'|, 2b')$-apex wall pair (W'', A') where $|A'| \leq a$ and its flaps have treewidth at most $q = f_4(2b') \cdot r$, then apply Step 2, otherwise apply Step 3.

Step 2. Consider the leveling $\tilde{W}$ of W'' and, in the representation W^R of W'' in $\tilde{W}$, pick a b'-wall $\bar{W}$ whose flap vertices do not correspond to a flap containing the pole of $D_{W'}$. Then use $\bar{W}$ in order to find an $(|A'|, b')$-apex wall pair (W''', A') of $D_{W'}$ whose compass does not contain the pole of $D_{W'}$. Notice that (A', W''') is also an $(|A'|, b')$-apex wall pair of G, therefore the algorithm can apply `Find-Irrelevant-Vertex` of Lemma 14 for $(G, k, q, 3, A', W''')$ and obtain, in $2^{\mathcal{O}_s(k \log k)} \cdot n$ time, an "irrelevant" flat 3-wall and a vertex v in its center 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 $(G \setminus v, k)$.

Step 3. Consider all the $\binom{R}{r} = 2^{\mathcal{O}_s(k^c \log k)}$ r-subwalls of $W^\bullet$, and for each one W' of them, consider its central h-subwall $\bar{W}$ and compute the canonical partition $\mathcal{Q}$ of $\bar{W}$. Then for each internal bag Q of $\mathcal{Q}$ add a new vertex v_Q and make it adjacent with all vertices in Q, then add a new vertex x_{all} and make it adjacent with 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 $\mathcal{O}(k \cdot m) = \mathcal{O}_s(k \cdot n)$ (using standard flow techniques), if there there are $f_9(s, k)$ 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 y exists, then report that (G, k) is a no-instance of $\mathcal{F}$-M-Deletion.

Notice that the third step of the algorithm, when applied takes time $2^{\mathcal{O}_s(k^c \log k)} \cdot n^2$. However, it cannot be applied more than k times during the course of the algorithm. As the first step runs in time $2^{\mathcal{O}_s(k^{2(c+1)}) \log k} \cdot n$, and the second step runs in time $2^{\mathcal{O}_s(k \log k)} \cdot n$, they may be applied at most n times, and the claimed time complexity follows.

5 Concluding remarks

Limitations of the irrelevant vertex technique. An intriguing open question is whether Vertex Deletion to $\mathcal{G}$ admits an algorithm in time $2^{\mathcal{O}_{s_\mathcal{F}}(k^c)} \cdot n^{\mathcal{O}(1)}$ for some universal constant c (i.e., not depending on the class $\mathcal{G}$). Clearly, this is not the case of the algorithms of Theorem 1 and Theorem 2, running in time $2^{\mathcal{O}_{s_\mathcal{F}}(k^{2(c+2)} \log k)} \cdot n^3$ and $2^{\mathcal{O}_{s_\mathcal{F}}(k^{2(c+1)} \log k)} \cdot n^2$, respectively, where c is the *the palette-variety* of the minor-obstruction set $\mathcal{F}$ of $\mathcal{G}$ which, from the corresponding proofs, is estimated to be $c = 2^{\mathcal{O}({s_\mathcal{F}}^2 \cdot \log s_\mathcal{F})}$ and $c = 2^{\mathcal{O}({s_\mathcal{F}}^{24} \cdot \log s_\mathcal{F})}$, respectively (recall that $s_\mathcal{F}$ 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*, as we mentioned in the end of Subsection 2.2. Having homogeneous walls is critical for the application of this technique 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. [26] who proved that, even for minor-closed families $\mathcal{G}$ that exclude a planar graph, the dependence on $\mathcal{G}$ on the degree of the polynomial kernel, which exists because of [22], is unavoidable subject to reasonable complexity assumptions.

Variants. Our approach can be easily modified so to deal with several variants of the Vertex Deletion to $\mathcal{G}$ problem such as the *annotated version* of the problem where the input comes with a set of vertices that we are permitted to remove, the *weighted version* where the vertices carry positive weights, the *counting version* where we are asked to count the number of (minimal) solutions, or the *colored version* where the vertices of the input graph are partitioned into k parts and we are requested to pick one vertex from each part. The details can be found in the full version of the paper [47].

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 additionally consider edge additions. We leave it as an open research challenge (a first step was done for the case of planar graphs [21]).

Lower bounds. Concerning lower bounds for Vertex Deletion to $\mathcal{G}$ under the Exponential Time Hypothesis [27], 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 [7, 27].

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.

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. URL: http://dl.acm.org/citation.cfm?id=1347082.1347153.

3 Ernst Althaus and Sarah Ziegler. Optimal Tree Decompositions Revisited: A Simpler Linear-Time FPT Algorithm. *CoRR*, abs/1912.09144, 2019. arXiv:1912.09144.

4 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Optimal algorithms for hitting (topological) minors on graphs of bounded treewidth. In *Proc. of the 12th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 89 of *LIPIcs*, pages 4:1–4:12, 2017. doi:10.4230/LIPIcs.IPEC.2017.4.

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:10.1137/130947374.

6 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.

7 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.

8 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.

9 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.

10 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.

11 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.

12 Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theoretical Computer Science*, 411:3736–3756, September 2010. doi:10.1016/j.tcs.2010.06.026.

13 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`.

14 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`.

15 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`.

16 Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Bidimensional parameters and local treewidth. *SIAM Journal on Discrete Mathematics*, 18(3):501–511, 2004. `doi:10.1137/S0895480103433410`.

17 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`.

18 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`.

19 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`.

20 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`.

21 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`.

22 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`.

23 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Hitting Topological Minors is FPT. *CoRR*, abs/1904.02944, 2019. To appear in *Proc. of the 52nd Annual ACM Symposium on Theory of Computing (STOC 2020)*. `arXiv:1904.02944`.

24 Fedor V. Fomin, Saket Saurabh, and Neeldhara Misra. Graph modification problems: A modern perspective. In *Proc. of the 9th International Frontiers in Algorithmics Workshop (FAW)*, volume 9130 of *LNCS*, pages 3–6, 2015. `doi:10.1007/978-3-319-19647-3_1`.

25 Michael R. Garey and David S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness*. W. H. Freeman and Co., San Francisco, 1979.

26 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`.

27 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`.

28 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, 2014. `doi:10.1137/1.9781611973402.130`.

29 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`.

30 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`.

31 Ken-ichi Kawarabayashi and Yusuke Kobayashi. Linear min-max relation between the treewidth of H-minor-free graphs and its largest grid. In *Proc. of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 14 of *LIPIcs*, pages 278–289, 2012. doi:10.4230/LIPIcs.STACS.2012.278.

32 Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 2011. doi:10.1016/j.jctb.2011.07.004.

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.

34 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.

35 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.

36 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.

37 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.

38 Alexandr V. Kostochka. Lower bound of the Hadwiger number of graphs by their average degree. *Combinatorica*, 4:307–316, 1984. doi:10.1007/BF02579141.

39 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.

40 Daniel Lokshtanov. 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.

41 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.

42 Rolf Niedermeier. *Invitation to fixed parameter algorithms*, volume 31. Oxford University Press, 2006. doi:10.1093/ACPROF:OSO/9780198566076.001.0001.

43 Ljubomir Perkovic and Bruce Reed. An Improved Algorithm for Finding Tree Decompositions of Small Width. *International Journal of Foundations of Computer Science*, 11:365–371, January 2000. doi:10.1142/S0129054100000247.

44 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.

45 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.

46 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.

47 Ignasi Sau, Gianos Stamoulis, and Dimitrios M. Thilikos. An FPT-algorithm for recognizing k-apices of minor-closed graph classes. *CoRR*, abs/2004.12692, 2020. arXiv:2004.12692.

48 Roded Sharan. *Graph Modification Problems and their Applications to Genomic Research.* PhD thesis, Sackler Faculty of Exact Sciences, School of Computer Science, 2002.

49 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*, pages 228–256, 2012.

50 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.