Optimal algorithms for hitting (topological) minors on graphs of bounded treewidth*

Julien Baste[1], Ignasi Sau[2,3], and Dimitrios M. Thilikos[2,4]

1 Université de Montpellier, LIRMM, Montpellier, France
2 AlGCo project-team, CNRS, LIRMM, France
3 Departamento de Matemática, Universidade Federal do Ceará, Fortaleza, Brazil
4 Department of Mathematics, National and Kapodistrian University of Athens, Greece

Abstract

For a fixed collection of graphs $\mathcal{F}$, the $\mathcal{F}$-M-Deletion problem consists in, given a graph G and an integer k, decide whether there exists $S \subseteq V(G)$ with $|S| \leq k$ such that $G \setminus S$ does not contain any of the graphs in $\mathcal{F}$ as a minor. We are interested in the parameterized complexity of $\mathcal{F}$-M-Deletion when the parameter is the treewidth of G, denoted by $\mathbf{tw}$. Our objective is to determine, for a fixed $\mathcal{F}$, the smallest function $f_{\mathcal{F}}$ such that $\mathcal{F}$-M-Deletion can be solved in time $f_{\mathcal{F}}(\mathbf{tw}) \cdot n^{\mathcal{O}(1)}$ on n-vertex graphs. Using and enhancing the machinery of boundaried graphs and small sets of representatives introduced by Bodlaender *et al.* [J ACM, 2016], we prove that when all the graphs in $\mathcal{F}$ are connected and at least one of them is planar, then $f_{\mathcal{F}}(w) = 2^{\mathcal{O}(w \cdot \log w)}$. When $\mathcal{F}$ is a singleton containing a clique, a cycle, or a path on i vertices, we prove the following asymptotically tight bounds:

- $f_{\{K_4\}}(w) = 2^{\Theta(w \cdot \log w)}$.
- $f_{\{C_i\}}(w) = 2^{\Theta(w)}$ for every $i \leq 4$, and $f_{\{C_i\}}(w) = 2^{\Theta(w \cdot \log w)}$ for every $i \geq 5$.
- $f_{\{P_i\}}(w) = 2^{\Theta(w)}$ for every $i \leq 4$, and $f_{\{P_i\}}(w) = 2^{\Theta(w \cdot \log w)}$ for every $i \geq 6$.

The lower bounds hold unless the Exponential Time Hypothesis fails, and the superexponential ones are inspired by a reduction of Marcin Pilipczuk [Discrete Appl Math, 2016]. The single-exponential algorithms use, in particular, the rank-based approach introduced by Bodlaender *et al.* [Inform Comput, 2015]. We also consider the version of the problem where the graphs in $\mathcal{F}$ are forbidden as *topological* minors, and prove essentially the same set of results holds.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory.

Keywords and phrases parameterized complexity; graph minors; treewidth; hitting minors; topological minors; dynamic programming; Exponential Time Hypothesis.

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

Let $\mathcal{F}$ be a finite non-empty collection of non-empty graphs. In the $\mathcal{F}$-M-Deletion (resp. $\mathcal{F}$-TM-Deletion) problem, we are given a graph G and an integer k, and the objective is to decide whether there exists a set $S \subseteq V(G)$ with $|S| \leq k$ such that $G \setminus S$ does not contain any of the graphs in $\mathcal{F}$ as a minor (resp. topological minor). These problems have a big expressive power, as instantiations of them correspond to several notorious problems. For instance, the cases $\mathcal{F} = \{K_2\}$, $\mathcal{F} = \{K_3\}$, and $\mathcal{F} = \{K_5, K_{3,3}\}$ of $\mathcal{F}$-M-Deletion (or

* Emails of authors: baste@lirmm.fr, ignasi.sau@lirmm.fr, sedthilk@thilikos.info.
This work has been supported by project DEMOGRAPH (ANR-16-CE40-0028).

Editors: John Q. Open and Joan R. Acces; Article No. ; pp. :1–:37
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$\mathcal{F}$-TM-Deletion) correspond to Vertex Cover, Feedback Vertex Set, and Vertex Planarization, respectively.

For the sake of readability, we use the notation $\mathcal{F}$-Deletion in statements that apply to *both* $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion. Note that if $\mathcal{F}$ contains a graph with at least one edge, then $\mathcal{F}$-Deletion is NP-hard by the classical result of Lewis and Yannakakis [22].

In this article we are interested in the parameterized complexity of $\mathcal{F}$-Deletion when the parameter is the treewidth of the input graph. Since the property of containing a graph as a (topological) minor can be expressed in Monadic Second Order logic (see [19] for explicit formulas), by Courcelle's theorem [7], $\mathcal{F}$-Deletion can be solved in time $\mathcal{O}^*(f(\mathbf{tw}))$ on graphs with treewidth at most $\mathbf{tw}$, where f is some computable function[1]. Our objective is to determine, for a fixed collection $\mathcal{F}$, which is the *smallest* such function f that one can (asymptotically) hope for, subject to reasonable complexity assumptions.

This line of research has attracted some interest during the last years in the parameterized complexity community. For instance, Vertex Cover is easily solvable in time $\mathcal{O}^*(2^{\mathcal{O}(\mathbf{tw})})$, called *single-exponential*, by standard dynamic-programming techniques, and no algorithm with running time $\mathcal{O}^*(2^{o(\mathbf{tw})})$ exists unless the Exponential Time Hypothesis (ETH)[2] fails [17].

For Feedback Vertex Set, standard dynamic programming techniques give a running time of $\mathcal{O}^*(2^{\mathcal{O}(\mathbf{tw}\cdot\log\mathbf{tw})})$, while the lower bound under the ETH [17] is again $\mathcal{O}^*(2^{o(\mathbf{tw})})$. This gap remained open for a while, until Cygan *et al.* [9] presented an optimal algorithm running in time $\mathcal{O}^*(2^{\mathcal{O}(\mathbf{tw})})$, using the celebrated *Cut&Count* technique. This article triggered several other techniques to obtain single-exponential algorithms for so-called *connectivity problems* on graph of bounded treewidth, mostly based on algebraic tools [3, 13].

Concerning Vertex Planarization, Jansen *et al.* [18] presented an algorithm of time $\mathcal{O}^*(2^{\mathcal{O}(\mathbf{tw}\cdot\log\mathbf{tw})})$ as a crucial subroutine in an FPT algorithm parameterized by k. Marcin Pilipczuk [26] proved that this running time is *optimal* under the ETH, by using the framework introduced by Lokshtanov *et al.* [24] for proving superexponential lower bounds.

Our results. We present a number of upper and lower bounds for $\mathcal{F}$-Deletion parameterized by treewidth, several of them being tight. Namely, we prove the following results, all the lower bounds holding under the ETH:

1. For every $\mathcal{F}$, $\mathcal{F}$-Deletion can be solved in time $\mathcal{O}^*\left(2^{2^{\mathcal{O}(\mathbf{tw}\cdot\log\mathbf{tw})}}\right)$.
2. For every connected[3] $\mathcal{F}$ containing at least one planar graph (resp. subcubic planar graph), $\mathcal{F}$-M-Deletion (resp. $\mathcal{F}$-TM-Deletion) can be solved in time $\mathcal{O}^*\left(2^{\mathcal{O}(\mathbf{tw}\cdot\log\mathbf{tw})}\right)$.
3. For any connected $\mathcal{F}$, $\mathcal{F}$-Deletion cannot be solved in time $\mathcal{O}^*(2^{o(\mathbf{tw})})$.
4. When $\mathcal{F}=\{K_i\}$, the clique on i vertices, $\{K_i\}$-Deletion cannot be solved in time $\mathcal{O}^*(2^{o(\mathbf{tw}\cdot\log\mathbf{tw})})$ for $i\geq 4$. Note that $\{K_i\}$-Deletion can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(\mathbf{tw})})$ for $i\leq 3$ [9], and that the case $i=4$ is tight by item **2** above (as K_4 is planar).
5. When $\mathcal{F}=\{C_i\}$, the cycle on i vertices, $\{C_i\}$-Deletion can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(\mathbf{tw})})$ for $i\leq 4$, and cannot be solved in time $\mathcal{O}^*(2^{o(\mathbf{tw}\cdot\log\mathbf{tw})})$ for $i\geq 5$. Note that, by items **2** and **3** above, this settles completely the complexity of $\{C_i\}$-Deletion for every $i\geq 3$.
6. When $\mathcal{F}=\{P_i\}$, the path on i vertices, $\{P_i\}$-Deletion can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(\mathbf{tw})})$ for $i\leq 4$, and cannot be solved in time $\mathcal{O}^*(2^{o(\mathbf{tw}\cdot\log\mathbf{tw})})$ for $i\geq 6$. Note that, by items **2** and **3** above, this settles completely the complexity of $\{P_i\}$-Deletion for every $i\geq 2$, except for $i=5$, where there is still a gap.

---

[1] We use the notation $\mathcal{O}^*(\cdot)$ that suppresses polynomial factors depending on the size of the input graph.
[2] The ETH states that 3-SAT on n variables cannot be solved in time $2^{o(n)}$; see [17] for more details.
[3] A *connected* collection $\mathcal{F}$ is a collection containing only connected graphs.

The results discussed in the last three items are summarized in Table 1. Note that the cases with $i \leq 3$ were already known [9, 17], except when $\mathcal{F} = \{P_3\}$.

$\mathcal{F}$ \ i	2	3	4	5	≥ 6
K_i	**tw**	**tw**	**tw** $\cdot \log$ **tw**	**tw** $\cdot \log$ **tw** ? $2^{\mathcal{O}(\mathbf{tw} \cdot \log \mathbf{tw})}$	**tw** $\cdot \log$ **tw** ? $2^{\mathcal{O}(\mathbf{tw} \cdot \log \mathbf{tw})}$
C_i	**tw**	**tw**	**tw**	**tw** $\cdot \log$ **tw**	**tw** $\cdot \log$ **tw**
P_i	**tw**	**tw**	**tw**	**tw** ? **tw** $\cdot \log$ **tw**	**tw** $\cdot \log$ **tw**

Table 1 Summary of our results when $\mathcal{F}$ equals $\{K_i\}$, $\{C_i\}$, or $\{P_i\}$. If only one value 'x' is written in the table (like '**tw**'), it means that the corresponding problem can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(x)})$, and that this bound is tight. An entry of the form 'x ? y' means that the corresponding problem cannot be solved in time $\mathcal{O}^*(2^{o(x)})$ and that it can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(y)})$. We interpret $\{C_2\}$-Deletion as Feedback Vertex Set. Grey cells correspond to known results.

Our techniques. The algorithm running in time $\mathcal{O}^*\left(2^{2^{\mathcal{O}(\mathbf{tw} \cdot \log \mathbf{tw})}}\right)$ uses and, in a sense, enhances, the machinery of boundaried graphs, equivalence relations, and representatives originating in the seminal work of Bodlaender *et al.* [5], and which has been subsequently used in [14, 15, 19]. For technical reasons, we use *branch* decompositions instead of tree decompositions, whose associated widths are equivalent from a parametric point of view [28].

In order to obtain the faster algorithm running in time $\mathcal{O}^*\left(2^{\mathcal{O}(\mathbf{tw} \cdot \log \mathbf{tw})}\right)$ when $\mathcal{F}$ is a connected collection containing at least a (subcubic) planar graph, we combine the above ingredients with additional arguments to bound the number and the size of the representatives of the equivalence relation defined by the encoding that we use to construct the partial solutions. Here, the connectivity of $\mathcal{F}$ guarantees that every connected component of a minimum-sized representative intersects its boundary set (cf. Lemma 23). The fact that $\mathcal{F}$ contains a (subcubic) planar graph is essential in order to bound the treewidth of the resulting graph after deleting a partial solution (cf. Lemma 9).

We present these algorithms for the topological minor version and then it is easy to adapt them to the minor version within the claimed running time (cf. Lemma 7).

The single-exponential algorithms when $\mathcal{F} \in \{\{P_3\}, \{P_4\}, \{C_4\}\}$ are ad hoc. Namely, the algorithms for $\{P_3\}$-Deletion and $\{P_4\}$-Deletion use standard (but nontrivial) dynamic programming techniques on graphs of bounded treewidth, exploiting the simple structure of graphs that do not contain P_3 or P_4 as a minor (or as a subgraph, which in the case of paths is equivalent). The algorithm for $\{C_4\}$-Deletion is more involved, and uses the rank-based approach introduced by Bodlaender *et al.* [3], exploiting again the structure of graphs that do not contain C_4 as a minor (cf. Lemma 12). It might seem counterintuitive that this technique works for C_4, and stops working for C_i with $i \geq 5$ (see Table 1). A possible reason for that is that the only cycles of a C_4-minor-free graph are triangles and each triangle is contained in a bag of a tree decomposition. This property, which is not true anymore for C_i-minor-free graphs with $i \geq 5$, permits to keep track of the structure of partial solutions with tables of small size.

As for the lower bounds, the general lower bound of $\mathcal{O}^*(2^{o(\mathbf{tw})})$ for connected collections is based on a simple reduction from Vertex Cover. The superexponential lower bounds, namely $\mathcal{O}^*(2^{o(\mathbf{tw} \cdot \log \mathbf{tw})})$, are strongly based on the ideas presented by Marcin Pilipczuk [26] for Vertex Planarization. We present a general hardness result (cf. Theorem 18) that applies to wide families of connected collections $\mathcal{F}$. Then, our superexponential lower bounds, as well as the result of Marcin Pilipczuk [26] itself, are corollaries of this general result. Combining Theorem 18 with **2**, it easily follows that the running time $\mathcal{O}^*(2^{\mathcal{O}(\mathbf{tw} \cdot \log \mathbf{tw})})$ is tight for a wide family of $\mathcal{F}$, for example, when all graphs in $\mathcal{F}$ are planar and 3-connected.

Further research. In order to complete the dichotomy for cliques and paths (see Table 1), it remains to settle the complexity when $\mathcal{F} = \{K_i\}$ with $i \geq 5$ and when $\mathcal{F} = \{P_5\}$. An ultimate goal is to establish the tight complexity of $\mathcal{F}$-Deletion for all collections $\mathcal{F}$, but we are still very far from it. In particular, we do not know whether there exists some $\mathcal{F}$ for which a double-exponential lower bound can be proved, or for which the complexities of $\mathcal{F}$-M-Deletion and $\mathcal{F}$-TM-Deletion differ.

Note that the connectivity of $\mathcal{F}$ was relevant in previous work on the $\mathcal{F}$-M-Deletion problem taking as the parameter the size of the solution [12, 19]. Getting rid of connectivity in both the lower and upper bounds we presented is an interesting avenue. We did not focus on optimizing either the degree of the polynomials involved or the constants involved in our algorithms. Concerning the latter, one could use the framework presented by Lokshtanov *et al.* [23] to prove lower bounds based on the *Strong* Exponential Time Hypothesis.

Finally, let us mention that Bonnet *et al.* [6] recently studied generalized feedback vertex set problems parameterized by treewidth, and obtained independently that excluding C_4 plays a fundamental role in the existence of single-exponential algorithms, similarly to our dichotomy for cycles summarized in Table 1.

Organization of the paper. In Section 2 we provide some preliminaries. The algorithms based on boundaried graphs are presented in Section 3, and the single-exponential algorithms for hitting paths and cycles are presented in Section 4. The general lower bound for connected collections is deferred to Appendix G, and the superexponential lower bounds are presented in Section 5. The proofs of the results marked with '($\star$)' have been moved to the appendices.

2 Preliminaries

In this section we provide some preliminaries to be used in the following sections. We include here only the "non-standard" definitions; the other ones can be found in Appendix A.

Block-cut trees. A connected graph G is *biconnected* if for any $v \in V(G)$, $G \setminus \{v\}$ is connected (notice that K_2 is the only biconnected graph that it is not 2-connected). A *block* of a graph G is a maximal biconnected subgraph of G. We name $\mathsf{block}(G)$ the set of all blocks of G and we name $\mathsf{cut}(G)$ the set of all cut vertices of G. If G is connected, we define the *block-cut tree* of G to be the tree $\mathsf{bct}(G) = (V, E)$ such that $V = \mathsf{block}(G) \cup \mathsf{cut}(G)$ and $E = \{\{B, v\} \mid B \in \mathsf{block}(G), v \in \mathsf{cut}(G) \cap V(B)\}$. Note that $L(\mathsf{bct}(G)) \subseteq \mathsf{block}(G)$. The block-cut tree of a graph can be computed in linear time using depth-first search [16]. Let $\mathcal{F}$ be a set of connected graphs such that for each $H \in \mathcal{F}$, $|V(H)| \geq 2$. Given $H \in \mathcal{F}$ and $B \in L(\mathsf{bct}(H))$, we say that (H, B) is an *essential pair* if for each $H' \in \mathcal{F}$ and each $B' \in L(\mathsf{bct}(H'))$, $|E(B)| \leq |E(B')|$. Given an essential pair (H, B) of $\mathcal{F}$, we define the *first vertex* of (H, B) to be, if it exists, the only cut vertex of H contained in $V(B)$, or an arbitrarily chosen vertex of $V(B)$ otherwise. We define the *second vertex* of (H, B) to be an arbitrarily chosen vertex of $V(B)$ that is a neighbor in $H[B]$ of the first vertex of (H, B). Note that, given an essential pair (H, B) of $\mathcal{F}$, the first vertex and the second vertex of (H, B) exist and, by definition, are fixed. Moreover, given an essential pair (H, B) of $\mathcal{F}$, we define the *core* of (H, B) to be the graph $H \setminus (V(B) \setminus \{a\})$ where a is the first vertex of (H, B). Note that a is a vertex of the core of (H, B).

Topological minors and graph separators. For the statement of our results, we need to consider the class $\mathcal{K}$ containing every connected graph G such that for each $B \in L(\mathsf{bct}(G))$ and for each $r \in \mathbb{N}$, $B \not\preceq_{\mathsf{tm}} K_{2,r}$ (or equivalently, $B \not\preceq_{\mathsf{m}} K_{2,r}$). Let H be a graph. We define the set of graphs $\mathsf{tpm}(H)$ as follows: among all the graphs containing H as a minor, we consider only those that are minimal with respect to the topological minor relation.

▶ **Observation 1.** *There is a function $f_1 : \mathbb{N} \to \mathbb{N}$ such that for every h-vertex graph H, every graph in* $\mathsf{tpm}(H)$ *has at most $f_1(h)$ vertices.*

▶ **Observation 2.** *Given two graphs H and G, H is a minor of G if and only if some of the graphs in* $\mathsf{tpm}(H)$ *is a topological minor of G.*

Let G be a graph and $S \subseteq V(G)$. Then for each connected component C of $G \setminus S$, we define the *cut-clique* of the triple (C, G, S) to be the graph whose vertex set is $V(C) \cup S$ and whose edge set is $E(G[V(C) \cup S]) \cup \binom{S}{2}$.

▶ **Lemma 1** ($\star$). *Let $i \geq 2$ be an integer, let H be an i-connected graph, let G be a graph, and let $S \subseteq V(G)$ such that $|S| \leq i - 1$. If H is a topological minor (resp. a minor) of G, then there exists a connected component G' of $G \setminus S$ such that H is a topological minor (resp. a minor) of the cut-clique of (G', G, S).*

▶ **Lemma 2** ($\star$). *Let G be a connected graph, let v be a cut vertex of G, and let V be the vertex set of a connected component of $G \setminus \{v\}$. If H is a connected graph such that $H \preceq_{\mathsf{tm}} G$ and for each leaf B of* $\mathsf{bct}(H)$*, $B \not\preceq_{\mathsf{tm}} G[V \cup \{v\}]$, then $H \preceq_{\mathsf{tm}} G \setminus V$.*

Graph collections. Let $\mathcal{F}$ be a collection of graphs. From now on instead of "collection of graphs" we use the shortcut "collection". If $\mathcal{F}$ is a collection that is finite, non-empty, and all its graphs are non-empty, then we say that $\mathcal{F}$ is a *regular collection*. For any regular collection $\mathcal{F}$, we define $\mathsf{size}(\mathcal{F}) = \max\{|V(H)| \mid H \in \mathcal{F}\} \cup \{|\mathcal{F}|\}$. Note that if the size of $\mathcal{F}$ is bounded, then the size of the graphs in $\mathcal{F}$ is also bounded. We say that $\mathcal{F}$ is a *planar collection* (resp. *planar subcubic collection*) if it is regular and *at least one* of the graphs in $\mathcal{F}$ is planar (resp. planar and subcubic). We say that $\mathcal{F}$ is a *connected collection* if it is regular and *all* the graphs in $\mathcal{F}$ are connected. We say that $\mathcal{F}$ is an *(topological) minor antichain* if no two of its elements are comparable via the (topological) minor relation.

Let $\mathcal{F}$ be a regular collection. We extend the (topological) minor relation to $\mathcal{F}$ such that, given a graph G, $\mathcal{F} \preceq_{\mathsf{tm}} G$ (resp. $\mathcal{F} \preceq_{\mathsf{m}} G$) if and only if there exists a graph $H \in \mathcal{F}$ such that $H \preceq_{\mathsf{tm}} G$ (resp. $H \preceq_{\mathsf{m}} G$). We also denote $\mathsf{ex}_{\mathsf{tm}}(\mathcal{F}) = \{G \mid \mathcal{F} \not\preceq_{\mathsf{tm}} G\}$, i.e., $\mathsf{ex}_{\mathsf{tm}}(\mathcal{F})$ is the class of graphs that do not contain any graph in $\mathcal{F}$ as a topological minor. The set $\mathsf{ex}_{\mathsf{m}}(\mathcal{F})$ is defined analogously.

Definition of the problems. Let $\mathcal{F}$ be a regular collection. We define the parameter $\mathbf{tm}_{\mathcal{F}}$ as the function that maps graphs to non-negative integers as follows:

$$\mathbf{tm}_{\mathcal{F}}(G) \quad = \quad \min\{|S| \mid S \subseteq V(G) \wedge G \setminus S \in \mathsf{ex}_{\mathsf{tm}}(\mathcal{F})\}. \tag{1}$$

The parameter $\mathbf{m}_{\mathcal{F}}$ is defined analogously. The main objective of this paper is to study the problem of computing the parameters $\mathbf{tm}_{\mathcal{F}}$ and $\mathbf{m}_{\mathcal{F}}$ for graphs of bounded treewidth under several instantiations of the collection $\mathcal{F}$. Note that in both problems, we can always assume that $\mathcal{F}$ is an antichain with respect to the considered relation. Indeed, this is the case because if $\mathcal{F}$ is contains two graphs H_1 and H_2 where $H_1 \preceq_{\mathsf{tm}} H_2$, then $\mathbf{tm}_{\mathcal{F}}(G) = \mathbf{tm}_{\mathcal{F}'}(G)$ where $\mathcal{F}' = \mathcal{F} \setminus \{H_2\}$ (similarly for the minor relation).

Throughout the article, we let n and $\mathbf{tw}$ be the number of vertices and the treewidth of the input graph of the considered problem, respectively. In some proofs, we will also use w to denote the width of a (nice) tree decomposition that is given together with the input graph (which will differ from $\mathbf{tw}$ by at most a factor 5).

3 Dynamic programming algorithms for computing $\mathbf{tm}_{\mathcal{F}}$

The purpose of this section is to prove the following results.

▶ **Theorem 3.** *If $\mathcal{F}$ is a regular collection, where $d = \mathsf{size}(\mathcal{F})$, then there exists an algorithm that solves $\mathcal{F}$-TM-Deletion in $2^{2^{\mathcal{O}_d(\mathbf{tw}\cdot\log\mathbf{tw})}} \cdot n$ steps.*

▶ **Theorem 4.** *If $\mathcal{F}$ is a connected and planar subcubic collection, where $d = \mathsf{size}(\mathcal{F})$, then there exists an algorithm that solves $\mathcal{F}$-TM-Deletion in $2^{\mathcal{O}_d(\mathbf{tw}\cdot\log\mathbf{tw})} \cdot n$ steps.*

▶ **Theorem 5.** *If $\mathcal{F}$ is a regular collection, where $d = \mathsf{size}(\mathcal{F})$, then there exists an algorithm that solves $\mathcal{F}$-M-Deletion in $2^{2^{\mathcal{O}_d(\mathbf{tw}\cdot\log\mathbf{tw})}} \cdot n$ steps.*

▶ **Theorem 6.** *If $\mathcal{F}$ is a connected and planar collection, where $d = \mathsf{size}(\mathcal{F})$, then there exists an algorithm that solves $\mathcal{F}$-M-Deletion in $2^{\mathcal{O}_d(\mathbf{tw}\cdot\log\mathbf{tw})} \cdot n$ steps.*

The following lemma is a direct consequence of Observation 2.

▶ **Lemma 7.** *Let $\mathcal{F}$ be a regular collection. Then, for every graph G, it holds that $\mathbf{m}_{\mathcal{F}}(G) = \mathbf{tm}_{\mathcal{F}'}(G)$ where $\mathcal{F}' = \bigcup_{F\in\mathcal{F}} \mathsf{tpm}(F)$.*

It is easy to see that for every (planar) graph F, the set $\mathsf{tpm}(F)$ contains a subcubic (planar) graph. Combining this observation with Lemma 7 and Observation 1, Theorems 5 and 6 follow directly from Theorems 3 and 4, respectively. The rest of this section is dedicated to the proofs of Theorems 3 and 4. For this, we need a number of definitions about boundaried graphs, their equivalence classes, and their branch decompositions. Many of these definitions were introduced in [5, 14] (see also [15, 19]), and can be found in Appendix B. We present here only the most fundamental definitions in order to be able to state our results.

Basic definitions about boundaried graphs. Let $t \in \mathbb{N}$. A *t-boundaried graph* is a triple $\mathbf{G} = (G, R, \lambda)$ where G is a graph, $R \subseteq V(G)$, $|R| = t$, and $\lambda : R \to \mathbb{N}^+$ is an injective function. We call R the *boundary* of $\mathbf{G}$ and we call the vertices of R *the boundary vertices* of $\mathbf{G}$. We also call G *the underlying graph* of $\mathbf{G}$. Moreover, we call $t = |R|$ the *boundary size* of $\mathbf{G}$ and we define the *label set* of $\mathbf{G}$ as $\Lambda(\mathbf{G}) = \lambda(R)$. We also say that $\mathbf{G}$ is a *boundaried graph* if there exists an integer t such that $\mathbf{G}$ is an t-boundaried graph. We say that a boundary graph $\mathbf{G}$ is *consecutive* if $\Lambda(\mathbf{G}) = [1, |R|]$. We define $\mathcal{B}^{(t)}$ as the set of all t-boundaried graphs.

Let $\mathbf{G}_1 = (G_1, R_1, \lambda_1)$ and $\mathbf{G}_2 = (G_2, R_2, \lambda_2)$ be two t-boundaried graphs. We define the *gluing operation* $\oplus$ such that $(G_1, R_1, \lambda_1) \oplus (G_2, R_2, \lambda_2)$ is the graph G obtained by taking the disjoint union of G_1 and G_2 and then, for each $i \in [1, t]$, identifying the vertex $\psi_{\mathbf{G}_1}^{-1}(i)$ and the vertex $\psi_{\mathbf{G}_2}^{-1}(i)$.

Let $\mathcal{F}$ be a regular collection and let t be a non-negative integer. We define an equivalence relation $\equiv^{(\mathcal{F},t)}$ on t-boundaried graphs as follows: Given two t-boundaried graphs $\mathbf{G}_1$ and $\mathbf{G}_2$, we write $\mathbf{G}_1 \equiv^{(\mathcal{F},t)} \mathbf{G}_2$ to denote that $\forall \mathbf{G} \in \mathcal{B}^{(t)}$, $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{G}_1 \iff \mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{G}_2$. We set up a *set of representatives* $\mathcal{R}^{(\mathcal{F},t)}$ as a set containing, for each equivalent class $\mathcal{C}$ of $\equiv^{(\mathcal{F},t)}$, some consecutive t-boundaried graph in $\mathcal{C}$ with minimum number of edges and no isolated vertices out of its boundary (if there are more than one such graphs, pick one arbitrarily). Given a t-boundaried graph $\mathbf{G}$ we denote by $\mathsf{rep}^{(\mathcal{F})}(\mathbf{G})$ the t-boundaried graph $\mathbf{B} \in \mathcal{R}^{(\mathcal{F},t)}$ where $\mathbf{B} \equiv^{(\mathcal{F},t)} \mathbf{G}$ and we call $\mathbf{B}$ the *$\mathcal{F}$-representative of* $\mathbf{G}$.

Given $t, r \in \mathbb{N}$, we define $\mathcal{A}^{(t)}_{\mathcal{F},r}$ as the set of all pairwise non-isomorphic boundaried graphs that contain at most r non-boundary vertices, whose label set is a subset of $[1, t]$, and whose underlying graph belongs in $\mathsf{ex}_{\mathsf{tm}}(\mathcal{F})$. Given a t-boundaried graph $\mathbf{B}$ and an integer $r \in \mathbb{N}$, we define the $(\mathcal{F}, r)$-folio of $\mathbf{B}$, denoted by $\mathsf{folio}(\mathbf{B}, \mathcal{F}, r)$ the set containing all boundaried graphs in $\mathcal{A}^{(t)}_{\mathcal{F},r}$ that are topological minors of $\mathbf{B}$.

▶ **Lemma 8** (⋆). *There exists a function $h_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that if $\mathcal{F}$ is a regular collection and $t \in \mathbb{N}$, then $|\mathcal{R}^{(\mathcal{F},t)}| \leq h_1(d,t)$ where $d = \mathsf{size}(\mathcal{F})$. Moreover $h_1(d,t) = 2^{2^{\mathcal{O}_d(t \cdot \log t)}}$.*

▶ **Lemma 9** (⋆). *There exists a function $\mu : \mathbb{N} \to \mathbb{N}$ such that for every planar subcubic collection $\mathcal{F}$, every graph in $\mathsf{ex}_{\mathsf{tm}}(\mathcal{F})$ has branchwidth at most $y = \mu(d)$ where $d = \mathsf{size}(\mathcal{F})$.*

We already have all the main ingredients to prove Theorem 3; the proof can be found in Appendix C. In order to prove Theorem 4, we need Lemma 11 below, which should be contrasted with Lemma 8. Its proof, which can be found in Appendix E, uses, among others, the following result of Baste *et al.* [2] on the number of labeled graphs of bounded treewidth.

▶ **Proposition 10** (Baste *et al.* [2]). *Let $n, y \in \mathbb{N}$. The number of labeled graphs with at most n vertices and branchwidth at most q is $2^{\mathcal{O}_q(n \cdot \log n)}$.*

▶ **Lemma 11** (⋆). *Let $t \in \mathbb{N}$ and $\mathcal{F}$ be a connected and planar collection, where $d = \mathsf{size}(\mathcal{F})$, and let $\mathcal{R}^{(\mathcal{F},t)}$ be a set of representatives. Then $|\mathcal{R}^{(\mathcal{F},t)}| = 2^{\mathcal{O}_d(t \cdot \log t)}$. Moreover, there exists an algorithm that given $\mathcal{F}$ and t, constructs a set of representatives $\mathcal{R}^{(\mathcal{F},t)}$ in $2^{\mathcal{O}_d(t \cdot \log t)}$ steps.*

The proof of Theorem 4 can be found in Appendix D. The main difference with respect to the proof of Theorem 3 is an improvement on the size of the tables of the dynamic programming algorithm, namely $|\mathcal{P}_e|$, where the fact that $\mathcal{F}$ is a connected and planar subcubic collection is exploited.

4 Single-exponential algorithms for hitting paths and cycles

In this section we show that if $\mathcal{F} \in \{\{P_3\}, \{P_4\}, \{C_4\}\}$, then $\mathcal{F}$-TM-Deletion can also be solved in single-exponential time. It is worth mentioning that the $\{C_i\}$-TM-Deletion problem has been studied in digraphs from a non-parameterized point of view [25].

The algorithms we present for $\{P_3\}$-TM-Deletion and $\{P_4\}$-TM-Deletion use standard dynamic programming techniques, and can be found in Appendix H. The definition of *nice tree decomposition* can also be found there.

We proceed to use the dynamic programming techniques introduced by Bodlaender *et al.* [3] to obtain a single-exponential algorithm for $\{C_4\}$-TM-Deletion. The algorithm we present solves the decision version of $\{C_4\}$-TM-Deletion: the input is a pair (G, k), where G is a graph and k is an integer, and the output is the boolean value $\mathbf{tm}_{\mathcal{F}}(G) \leq k$.

Given a graph G, we denote by $n(G) = |V(G)|$, $m(G) = |E(G)|$, $\mathsf{c}_3(G)$ the number of C_3's that are subgraphs of G, and $\mathsf{cc}(G)$ the number of connected components of G. We say that G satisfies the *C_4-condition* if G does not contain the diamond as a subgraph and $n(G) - m(G) + \mathsf{c}_3(G) = \mathsf{cc}(G)$. As in the case of P_3 and P_4, we state in Lemma 12 a structural characterization of the graphs that exclude C_4 as a (topological) minor.

▶ **Lemma 12** (⋆). *Let G be a graph. $C_4 \not\preceq_{\mathsf{tm}} G$ if and only if G satisfies the C_4-condition.*

▶ **Lemma 13** (⋆). *If G is a non-empty graph such that $C_4 \not\preceq_{\mathsf{tm}} G$, then $m(G) \leq \frac{3}{2}(n(G) - 1)$.*

We are now going to restate the tools introduced by Bodlaender *et al.* [3] that we need for our purposes. Let U be a set. We define $\Pi(U)$ to be the set of all partitions of U. Given two partitions p and q of U, we define the coarsening relation $\sqsubseteq$ such that $p \sqsubseteq q$ if for each $S \in q$, there exists $S' \in p$ such that $S \subseteq S'$. $(\Pi(U), \sqsubseteq)$ defines a lattice with minimum element $\{\{U\}\}$ and maximum element $\{\{x\} \mid x \in U\}$. On this lattice, we denote by $\sqcap$ the meet operation and by $\sqcup$ the join operation. Let $p \in \Pi(U)$. For $X \subseteq U$ we denote by $p_{\downarrow X} = \{S \cap X \mid S \in p, S \cap X \neq \emptyset\} \in \Pi(X)$ the partition obtained by removing all elements not in X from p, and analogously for $U \subseteq X$ we denote $p_{\uparrow X} = p \cup \{\{x\} \mid x \in X \setminus U\} \in \Pi(X)$ the partition obtained by adding to p a singleton for each element in $X \setminus U$. Given a subset S of U, we define the partition $U[S] = \{\{x\} \mid x \in U \setminus S\} \cup \{S\}$. A set of *weighted partitions* is a set $\mathcal{A} \subseteq \Pi(U) \times \mathbb{N}$. We also define $\mathsf{rmc}(\mathcal{A}) = \{(p, w) \in \mathcal{A} \mid \forall (p', w') \in \mathcal{A} : p' = p \Rightarrow w \leq w'\}$.

We now define some operations on weighted partitions. Let U be a set and $\mathcal{A} \subseteq \Pi(U) \times \mathbb{N}$.

Union. Given $\mathcal{B} \subseteq \Pi(U) \times \mathbb{N}$, we define $\mathcal{A} \uplus \mathcal{B} = \mathsf{rmc}(\mathcal{A} \cup \mathcal{B})$.
Insert. Given a set X such that $X \cap U = \emptyset$, we define $\mathsf{ins}(X, \mathcal{A}) = \{(p_{\uparrow U \cup X}, w) \mid (p, w) \in \mathcal{A}\}$.
Shift. Given $w' \in \mathbb{N}$, we define $\mathsf{shft}(w', \mathcal{A}) = \{(p, w + w') \mid (p, w) \in \mathcal{A}\}$.
Glue. Given a set S, we define $\hat{U} = U \cup S$ and $\mathsf{glue}(S, \mathcal{A}) \subseteq \Pi(\hat{U}) \times \mathbb{N}$ as
$\mathsf{glue}(S, \mathcal{A}) = \mathsf{rmc}(\{(\hat{U}[S] \sqcap p_{\uparrow \hat{U}}, w \mid (p, w) \in \mathcal{A}\})$.
Given $w : \hat{U} \times \hat{U} \to \mathcal{N}$, we define $\mathsf{glue}_w(\{u, v\}, \mathcal{A}) = \mathsf{shft}(w(u, v), \mathsf{glue}(\{u, v\}, \mathcal{A}))$.
Project. Given $X \subseteq U$, we define $\overline{X} = U \setminus X$ and $\mathsf{proj}(X, \mathcal{A}) \subseteq \Pi(\overline{X}) \times \mathbb{N}$ as
$\mathsf{proj}(X, \mathcal{A}) = \mathsf{rmc}(\{(p_{\downarrow \overline{X}}, w) \mid (p, w) \in \mathcal{A}, \forall e \in X : \forall e' \in \overline{X} : p \sqsubseteq U[ee']\})$.
Join. Given a set U', $\mathcal{B} \subseteq \Pi(U) \times \mathbb{N}$, and $\hat{U} = U \cup U'$, we define $\mathsf{join}(\mathcal{A}, \mathcal{B}) \subseteq \Pi(\hat{U}) \times \mathbb{N}$ as
$\mathsf{join}(\mathcal{A}, \mathcal{B}) = \mathsf{rmc}(\{(p_{\uparrow \hat{U}} \sqcap q_{\uparrow \hat{U}}, w_1 + w_2) \mid (p, w_1) \in \mathcal{A}, (q, w_2) \in \mathcal{B}\})$.

▶ **Proposition 14** (Bodlaender *et al.* [3]). *Each of the operations union, insert, shift, glue, and project can be carried out in time $s \cdot |U|^{\mathcal{O}(1)}$, where s is the size of the input of the operation. Given two weighted partitions $\mathcal{A}$ and $\mathcal{B}$, $\mathsf{join}(\mathcal{A}, \mathcal{B})$ can be computed in time $|\mathcal{A}| \cdot |\mathcal{B}| \cdot |U|^{\mathcal{O}(1)}$.*

Given a weighted partition $\mathcal{A} \subseteq \Pi(U) \times \mathbb{N}$ and a partition $q \in \Pi(U)$, we define $\mathsf{opt}(q, \mathcal{A}) = \min\{w \mid (p, w) \in \mathcal{A}, p \sqcap q = \{U\}\}$. Given two weighted partitions $\mathcal{A}, \mathcal{A}' \subseteq \Pi(U) \times \mathbb{N}$, we say that $\mathcal{A}$ *represents* $\mathcal{A}'$ if for each $q \in \Pi(U)$, $\mathsf{opt}(q, \mathcal{A}) = \mathsf{opt}(q, \mathcal{A}')$. Given a set Z and a function $f : 2^{\Pi(U) \times \mathbb{N}} \times Z \to 2^{\Pi(U) \times \mathbb{N}}$, we say that f *preserves representation* if for each two weighted partitions $\mathcal{A}, \mathcal{A}' \subseteq \Pi(U) \times \mathbb{N}$ and each $z \in Z$, it holds that if $\mathcal{A}'$ represents $\mathcal{A}$ then $f(\mathcal{A}', z)$ represents $f(\mathcal{A}, z)$.

▶ **Proposition 15** (Bodlaender *et al.* [3]). *The union, insert, shift, glue, project, and join operations preserve representation.*

▶ **Theorem 16** (Bodlaender *et al.* [3]). *There exists an algorithm reduce that, given a set of weighted partitions $\mathcal{A} \subseteq \Pi(U) \times \mathbb{N}$, outputs in time $|\mathcal{A}| \cdot 2^{(\omega-1)|U|} \cdot |U|^{\mathcal{O}(1)}$ a set of weighted partitions $\mathcal{A}' \subseteq \mathcal{A}$ such that $\mathcal{A}'$ represents $\mathcal{A}$ and $|\mathcal{A}'| \leq 2^{|U|}$, where ω denotes the matrix multiplication exponent.*

We now have all the tools needed to describe our algorithm. This algorithm is based on the one given in [3, Section 3.5] for Feedback Vertex Set. We define a new graph $G_0 = (V(G) \cup \{v_0\}, E(G) \cup E_0)$, where v_0 is a new vertex and $E_0 = \{\{v_0, v\} \mid v \in V(G)\}$. The role of v_0 is to artificially guarantee the connectivity of the solution graph, so that the machinery of Bodlaender *et al.* [3] can be applied. In the following, for each subgraph H of G, for each $Z \subseteq V(H)$, and for each $Z_0 \subseteq E_0 \cap E(H)$, we denote by $H\langle Z, Z_0\rangle$ the graph $\left(Z, Z_0 \cup \left(E(H) \cap \binom{Z \setminus \{v_0\}}{2}\right)\right)$.

Given a nice tree decomposition of G of width w, we define a nice tree decomposition $((T, \mathcal{X}), r, \mathcal{G})$ of G_0 of width $w + 1$ such that the only empty bags are the root and the leaves and for each $t \in T$, if $X_t \neq \emptyset$ then $v_0 \in X_t$. Note that this can be done in linear time. For each bag t, each integers i, j, and ℓ, each function $\mathbf{s} : X_t \to \{0, 1\}$, each function $\mathbf{s}_0 : \{v_0\} \times \mathbf{s}^{-1}(1) \to \{0, 1\}$, and each function $\mathbf{r} : E(G_t\langle \mathbf{s}^{-1}(1), \mathbf{s}_0^{-1}(1)\rangle) \to \{0, 1\}$, if $C_4 \not\preceq_{\mathsf{tm}} G_t\langle \mathbf{s}^{-1}(1), \mathbf{s}_0^{-1}(1)\rangle$, we define:

$$
\begin{aligned}
\mathcal{E}_t(p,\mathbf{s},\mathbf{s}_0,\mathbf{r},i,j,\ell) \quad = \quad & \{(Z,Z_0) \mid (Z,Z_0) \in 2^{V_t} \times 2^{E_0 \cap E(G_t)} \\
& |Z| = i,\ |E(G_t\langle Z,Z_0\rangle)| = j,\ \mathsf{c}_3(G_t\langle Z,Z_0\rangle) = \ell, \\
& G_t\langle Z,Z_0\rangle \text{ does not contain the diamond as a subgraph}, \\
& Z \cap X_t = \mathbf{s}^{-1}(1),\ Z_0 \cap (X_t \times X_t) = \mathbf{s}_0^{-1}(1),\ v_0 \in X_t \Rightarrow \mathbf{s}(v_0) = 1, \\
& \quad \forall u \in Z \setminus X_t : \text{ either } t \text{ is the root or} \\
& \qquad\qquad \exists u' \in \mathbf{s}^{-1}(1) : u \text{ and } u' \text{ are connected in } G_t\langle Z,Z_0\rangle, \\
& \forall v_1,v_2 \in \mathbf{s}^{-1}(1) : p \sqsubseteq V_t[\{v_1,v_2\}] \Leftrightarrow v_1 \text{ and } v_2 \text{ are} \\
& \qquad\qquad\qquad \text{connected in } G_t\langle Z,Z_0\rangle, \\
& \forall e \in E(G_t\langle Z,Z_0\rangle) \cap \binom{\mathbf{s}^{-1}(1)}{2} : \mathbf{r}(e) = 1 \Leftrightarrow\ e \text{ is an} \\
& \qquad\qquad\qquad \text{edge of a } C_3 \text{ in } G_t\langle Z,Z_0\rangle\} \\
\mathcal{A}_t(\mathbf{s},\mathbf{s}_0,\mathbf{r},i,j,\ell) \quad = \quad & \{p \mid p \in \Pi(\mathbf{s}^{-1}(1)),\ \mathcal{E}_t(p,\mathbf{s},\mathbf{s}_0,\mathbf{r},i,j,\ell) \neq \emptyset\}.
\end{aligned}
$$

Otherwise, i.e., if $C_4 \preceq_{\mathsf{tm}} G_t\langle \mathbf{s}^{-1}(1), \mathbf{s}_0^{-1}(1)\rangle$, we define $\mathcal{A}_t(\mathbf{s},\mathbf{s}_0,\mathbf{r},i,j,\ell) = \emptyset$.

Note that we do not need to keep track of partial solutions if $C_4 \preceq_{\mathsf{tm}} G_t\langle \mathbf{s}^{-1}(1), \mathbf{s}_0^{-1}(1)\rangle$, as we already know they will not lead to a global solution. Moreover, if $C_4 \not\preceq_{\mathsf{tm}} G_t\langle \mathbf{s}^{-1}(1), \mathbf{s}_0^{-1}(1)\rangle$, then by Lemma 13 it follows that $m(G_t\langle \mathbf{s}^{-1}(1), \mathbf{s}_0^{-1}(1)\rangle) \leq \frac{3}{2}(n(G_t\langle \mathbf{s}^{-1}(1), \mathbf{s}_0^{-1}(1)\rangle) - 1)$.

Using the definition of $\mathcal{A}_r$, Lemma 12, and Lemma 13 we have that $\mathbf{tm}_{\{C_4\}}(G) \leq k$ if and only if for some $i \geq |V(G) \cup \{v_0\}| - k$ and some $j \leq \frac{2}{3}(i-1)$, we have $\mathcal{A}_r(\varnothing,\varnothing,\varnothing,i,j,1+j-i) \neq \emptyset$. For each $t \in V(T)$, we assume that we have already computed $\mathcal{A}_{t'}$ for each children t' of t, and in Appendix F we show how to compute $\mathcal{A}_t$, distinguishing several cases depending on the type of node t. The proof of the following theorem can also be found in Appendix F.

▶ **Theorem 17** ($\star$). $\{C_4\}$-TM-Deletion *can be solved in time* $2^{\mathcal{O}(\mathbf{tw})} \cdot n^7$.

5 Superexponential lower bound for specific cases

In this section, we focus on the graph classes $\mathcal{P} = \{P_i \mid i \geq 6\}$ and $\mathcal{K}$, and we show the following theorem. Let us recall that $\mathcal{K}$ is the set containing every connected graph G such that for each leaf $B \in L(\mathsf{bct}(G))$ and $r \in \mathbb{N}$, $B \not\preceq_{\mathsf{tm}} K_{2,r}$ (or $B \not\preceq_{\mathsf{m}} K_{2,r}$, which is equivalent).

▶ **Theorem 18.** *Let $\mathcal{F}$ be a regular collection such that $\mathcal{F} \subseteq \mathcal{P}$ or $\mathcal{F} \subseteq \mathcal{K}$. Unless the ETH fails, neither $\mathcal{F}$-TM-Deletion nor $\mathcal{F}$-M-Deletion can be solved in time $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$.*

In particular, this theorem implies the result of Pilipczuk [26] as a corollary. Indeed, Vertex Planarization corresponds to $\mathcal{F}$-Deletion where $\mathcal{F} = \{K_5, K_{3,3}\}$, and note that $\{K_5, K_{3,3}\} \subseteq \mathcal{K}$. Note also that Theorem 18 also implies the results stated in items **4** and **5** of the introduction, as all these graphs are easily seen to belong in $\mathcal{K}$.

▶ **Corollary 19.** *Unless the ETH fails, for each $\mathcal{F} \in \{\{C_i\} \mid i \geq 5\} \cup \{\{K_i\} \mid i \geq 4\}$, neither $\mathcal{F}$-TM-Deletion nor $\mathcal{F}$-M-Deletion can be solved in time $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$.*

In the following we prove Theorem 18 for $\mathcal{F}$-TM-Deletion, and we explain in Appendix I how to modify the proof to obtain the result for $\mathcal{F}$-M-Deletion. To prove Theorem 18, we reduce from $k \times k$ Permutation Clique ($k \times k$ P. Clique for short), defined by Lokshtanov *et al.* [24]. In this problem, we are given an integer k and a graph G with vertex set $[1,k] \times [1,k]$. The question is whether there is a k-clique in G with exactly one element from each row and exactly one element from each column. Lokshtanov *et al.* [24] proved that $k \times k$ P. Clique cannot be solved in time $2^{o(k \log k)}$ unless the ETH fails.

We now present the common part of the construction for both $\mathcal{P}$ and $\mathcal{K}$. Let $\mathcal{F}$ be a regular collection such that $\mathcal{F} \subseteq \mathcal{P}$ or $\mathcal{F} \subseteq \mathcal{K}$. Note that if $\mathcal{F} \subseteq \mathcal{P}$, then $|\mathcal{F}| = 1$. Let us fix (H, B) to be an essential pair of $\mathcal{F}$. We first define some gadgets that generalize the *K_5-edge gadget* and the *s-choice gadget* introduced in [26]. Given a graph G and two vertices x and y of G, by *introducing an H-edge gadget* between x and y we mean that we add a copy of H where we identify the first vertex of (H, B) with y and the second vertex of (H, B) with x. Using the fact that an H-edge gadget between two vertices x and y is a copy of H and that $\{x, y\}$ is a cut set, we have that the H-edge gadgets clearly satisfy the following.

▶ **Proposition 20.** *If $\mathcal{F}$-TM-Deletion has a solution on (G, k) then this solution intersects every H-edge gadget, and there exists a solution S such that for each H-edge gadget A between two vertices x and y, $V(A) \cap S \subseteq \{x, y\}$ and $\{x, y\} \cap S \neq \emptyset$.*

In the following, we will always assume that the solution that we take into consideration is a solution satisfying the properties given by Proposition 20. Moreover, we will restrict the solution to contain only vertices of H-edge gadgets by setting an appropriate budget to the number of vertices we can remove from the input graph G.

Given a graph G and two vertices x and y of G, by *introducing a B-edge gadget* between x and y we mean that we add a copy of B where we identify the first vertex of (H, B) with y and the second vertex of (H, B) with x. Given a graph G and three vertices x, y, and z of G, by *introducing a double H-edge gadget* between x and z through y we mean that we introduce an H-edge gadget between z and y, and a B-edge gadget between x and y.

Given a set of s vertices $\{x_i \mid i \in [1, s]\}$, by *introducing an H-choice gadget connecting* $\{x_i \mid i \in [1, s]\}$, we mean that we add $2s + 2$ vertices z_i, $i \in [0, 2s + 1]$, for each $i \in [0, 2s]$, we introduce an H-edge gadget between z_i and z_{i+1}, and for each $i \in [1, s]$, we introduce a B-edge gadget between x_i and z_{2i-1} and another one between x_i and z_{2i}. We see the H-choice gadget as a graph induced by $\{x_i \mid i \in [1, s]\} \cup \{z_i \mid i \in [0, 2s]\}$, the B-edge gadgets, and the H-edge gadgets. The following proposition is similar to [26, Lemma 5].

▶ **Proposition 21** ($\star$). *For every H-choice gadget C connecting $\{x_i \mid i \in [1, s]\}$, any solution S of $\mathcal{F}$-TM-Deletion satisfies $|S \cap V(C)| \geq 2s$, for every $i \in [1, s]$ there exists a solution S such that $x_i \notin S$, and for every solution S with $|S \cap V(C)| = 2s$, $\exists i \in [1, s]$ such that $x_i \notin S$.*

We now start the description of the general construction. Given an instance (G, k) of $k \times k$ P. Clique, we construct an instance (G', ℓ) of $\mathcal{F}$-TM-Deletion, which we call the *general H-construction* of (G, k). We first introduce $k^2 + 2k$ vertices, namely $\{c_i \mid i \in [1, k]\}$, $\{r_i \mid i \in [1, k]\}$, and $\{t_{i,j} \mid i, j \in [1, k]\}$. For each $i, j \in [1, k]$, we add the edges $\{r_j, t_{i,j}\}$ and $\{t_{i,j}, c_i\}$. For each $j \in [1, k]$, we introduce an H-choice gadget connecting $\{t_{i,j} \mid i \in [1, k]\}$. This part of the construction is depicted in Figure 1 in Appendix I.

We now describe how we encode the edges of G in G'. For each $e \in E(G)$, we define the integers $p(e)$, $\gamma(e)$, $q(e)$, and $\delta(e)$ in $[1, k]$, such that $e = \{(p(e), \gamma(e)), (q(e), \delta(e))\}$ with $p(e) \leq q(e)$. Note that the edges e with $p(e) = q(e)$ are not relevant to our construction and hence we safely forget them. For each $e \in E(G)$, we add to G' three new vertices, d_e^ℓ, d_e^m, and d_e^r, and four edges $\{d_e^\ell, c_{p(e)}\}$, $\{d_e^\ell, r_{\gamma(e)}\}$, $\{d_e^r, c_{q(e)}\}$, and $\{d_e^r, r_{\delta(e)}\}$. We introduce a double H-edge gadget between d_e^ℓ and d_e^r through d_e^m. The encoding of an edge $e \in E(G)$ is depicted in Figure 2 in Appendix I. For each $1 \leq p < q \leq k$, we define $E(p, q) = \{e \in E(G) \mid (p(e), q(e)) = (p, q)\}$ and we introduce an H-choice gadget connecting $\{d_e^\ell \mid e \in E(p, q)\}$.

For each $e \in E(G)$, we increase the size of the requested solution in G' by one, the initial budget being the sum of the budget given by Proposition 21 over all the H-choice gadgets introduced in the construction. Because of the double H-edge gadget, we need to take in the solution either d_e^m or both d_e^ℓ and d_e^r. The extra budget given for each edge permits to include d_e^m in the solution. If the H-choice gadget connected to d_e^ℓ already chooses d_e^ℓ to be in the solution, then we can use the extra budget given for the edge e to choose d_e^r instead of d_e^m. In the case d_e^m is chosen, in the resulting graph $c_{p(e)}$ remains connected to $r_{\gamma(e)}$ and $c_{q(e)}$ remains connected to $r_{\delta(e)}$. In the following, we consider only a solution S such that either $\{d_e^\ell, d_e^m, d_e^r\} \cap S = \{d_e^\ell, d_e^r\}$ or $\{d_e^\ell, d_e^m, d_e^r\} \cap S = \{d_e^m\}$ for each $e \in E(G)$.

We set $\ell = 3|E(G)| + 2k^2$. By construction, this budget is tight and permits to take only a minimum-size solution in every H-choice gadget and one endpoint of each H-edge gadget between d_e^r and d_e^m, $e \in E(G)$. This concludes the general H-construction (G', ℓ) of (G, k).

Let us now discuss about the treewidth of G'. By deleting $2k$ vertices, namely the vertices $\{c_i \mid i \in [1,k]\}$ and the vertices $\{r_j \mid j \in [1,k]\}$, we obtain a graph where each connected component is an H-choice gadget, with eventually some pendant H-edge gadgets or double H-edge gadgets. As the treewidth of the H-choice gadget, the H-edge gadget, and the double H-choice gadget is linear in $|V(H)|$, we obtain that $\mathbf{tw}(G) = \mathcal{O}_d(k)$ (recall that $d = \mathsf{size}(\mathcal{F})$).

We explain in Appendix I that, given a permutation $\sigma : [1,k] \to [1,k]$ defining a solution of $k \times k$ P. Clique on (G,k), we can define a so-called *σ-general H-solution S* having *nice* properties. Conversely, given a set $S \subseteq V(G')$ of size at most $3|E(G)| + 2k^2$ satisfying the so-called *permutation property*, we can define (cf. Lemma 39) a unique permutation σ that defines a k-clique in G; we call σ the *associated permutation* of S.

To conclude the reduction, we deal separately with the cases $\mathcal{F} \subseteq \mathcal{P}$ and $\mathcal{F} \subseteq \mathcal{K}$. For each such $\mathcal{F}$, we assume w.l.o.g. that $\mathcal{F}$ is a topological minor antichain, we fix (H, B) to be an essential pair of $\mathcal{F}$, and given an instance (G,k) of $k \times k$ P. Clique, we start from the general H-construction (G', ℓ) and add some edges and vertices in order to build an instance (G'', ℓ) of $\mathcal{F}$-TM-Deletion. We show that if $k \times k$ P. Clique on (G,k) has a solution σ, then the σ-general H-solution is a solution of $\mathcal{F}$-TM-Deletion on (G'', ℓ). Conversely, we show that if $\mathcal{F}$-TM-Deletion on (G'', ℓ) has a solution S, then this solution satisfies the permutation property. This gives, by Lemma 39, that the associated permutation σ of S is a solution of $k \times k$ P. Clique on (G,k). The details can be found in Appendix I.

References

1. Z. Baia, J. Tu, and Y. Shi. An improved algorithm for the vertex cover P_3 problem on graphs of bounded treewidth. *CoRR*, abs/1603.09448, 2017.
2. J. Baste, M. Noy, and I. Sau. The number of labeled graphs of bounded treewidth. *CoRR*, abs/1604.07273, 2016. To appear in *Proc. of WG 2017*.
3. H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86–111, 2015.
4. H. L. Bodlaender, P. G. Drange, M. S. Dregi, F. V. Fomin, D. Lokshtanov, and M. Pilipczuk. A $c^k n$ 5-Approximation Algorithm for Treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016.
5. H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos. (meta) kernelization. *Journal of the ACM*, 63(5):44:1–44:69, 2016.
6. É. Bonnet, N. Brettell, O. Kwon, and D. Marx. Generalized feedback vertex set problems on bounded-treewidth graphs: chordality is the key to single-exponential parameterized algorithms. *CoRR*, abs/1704.06757, 2017.
7. B. Courcelle. The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs. *Information and Computation*, 85(1):12–75, 1990.
8. M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

9 M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving Connectivity Problems Parameterized by Treewidth in Single Exponential Time. In *Proc. of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 150–159, 2011.

10 R. Diestel. *Graph Theory*. Springer-Verlag, 3rd edition, 2005.

11 R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

12 F. V. Fomin, D. Lokshtanov, N. Misra, and S. 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.

13 F. V. Fomin, D. Lokshtanov, F. Panolan, and S. Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *Journal of the ACM*, 63(4):29:1–29:60, 2016.

14 F. V. Fomin, D. Lokshtanov, S. Saurabh, and D. M. Thilikos. Bidimensionality and kernels. In *Proc. of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 503–510, 2010. Full version available at *CoRR*, abs/1606.05689, 2016.

15 V. Garnero, C. Paul, I. Sau, and D. M. Thilikos. Explicit linear kernels via dynamic programming. *SIAM Journal on Discrete Mathematics*, 29(4):1864–1894, 2015.

16 J. E. Hopcroft and R. E. Tarjan. Efficient algorithms for graph manipulation. *Communications of ACM*, 16(6):372–378, 1973.

17 R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

18 B. M. P. Jansen, D. Lokshtanov, and S. Saurabh. A near-optimal planarization algorithm. In *Proc. of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1802–1811, 2014.

19 E. J. Kim, A. Langer, C. Paul, F. Reidl, P. Rossmanith, I. Sau, and S. Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. *ACM Transactions on Algorithms*, 12(2):21:1–21:41, 2016.

20 T. Kloks. *Treewidth. Computations and Approximations.* Springer-Verlag LNCS, 1994.

21 J. Komlós and E. Szemerédi. Topological cliques in graphs 2. *Combinatorics, Probability & Computing*, 5:79–90, 1996.

22 J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

23 D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.

24 D. Lokshtanov, D. Marx, and S. Saurabh. Slightly superexponential parameterized problems. In *Proc. of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 760–776, 2011.

25 D. Paik, S. M. Reddy, and S. Sahni. Deleting vertices to bound path length. *IEEE Transactions on Computers*, 43(9):1091–1096, 1994.

26 M. Pilipczuk. A tight lower bound for Vertex Planarization on graphs of bounded treewidth. *Discrete Applied Mathematics*, DOI: `http://dx.doi.org/10.1016/j.dam.2016.05.019`, 2016, to appear.

27 N. Robertson and P. D. Seymour. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, 1986.

28 N. Robertson and P. D. Seymour. Graph minors. X. Obstructions to tree decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, 1991.

29 J. Tu, L. Wu, J. Yuan, and L. Cui. On the vertex cover P_3 problem parameterized by treewidth. *Journal of Combinatorial Optimization*, DOI: `10.1007/s10878-016-9999-6`, 2016, to appear.

A Extended preliminaries

Sets, integers, and functions. We denote by $\mathbb{N}$ the set of every non-negative integer and we set $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. Given two integers p and q, the set $[p, q]$ refers to the set of every integer r such that $p \leq r \leq q$. Moreover, for each integer $p \geq 1$, we set $\mathbb{N}_{\geq p} = \mathbb{N} \setminus [0, p-1]$. In the set $[1, k] \times [1, k]$, a *row* is a set $\{i\} \times [1, k]$ and a *column* is a set $[1, k] \times \{i\}$ for some $i \in [1, k]$.

We use $\emptyset$ to denote the empty set and $\varnothing$ to denote the empty function, i.e., the unique subset of $\emptyset \times \emptyset$. Given a function $f : A \to B$ and a set S, we define $f|_S = \{(x, f(x)) \mid x \in S \cap A\}$. Moreover if $S \subseteq A$, we set $f(S) = \bigcup_{s \in S}\{f(s)\}$. Given a set S, we denote by $\binom{S}{2}$ the set containing every subset of S that has cardinality 2. We also denote by 2^S the set of all the subsets of S. If $\mathcal{S}$ is a collection of objects where the operation $\cup$ is defined, then we then denote $\bigcup \mathcal{S} = \bigcup_{X \in \mathcal{S}} X$.

Let $p \in \mathbb{N}$ with $p \geq 2$, let $f : \mathbb{N}^p \to \mathbb{N}$, and let $g : \mathbb{N}^{p-1} \to \mathbb{N}$. We say that $f(x_1, \ldots, x_p) = \mathcal{O}_{x_p}(g(x_1, \ldots, x_{p-1}))$ if there is a function $h : \mathbb{N} \to \mathbb{N}$ such that $f(x_1, \ldots, x_p) = \mathcal{O}(h(x_p) \cdot g(x_1, \ldots, x_{p-1}))$.

Graphs. All the graphs that we consider in this paper are undirected, finite, and without loops or multiple edges. We use standard graph-theoretic notation, and we refer the reader to [10] for any undefined terminology. Given a graph G, we denote by $V(G)$ the set of vertices of G and by $E(G)$ the set of the edges of G. We call $|V(G)|$ *the size* of G. A graph is *the empty* graph if its size is 0. We also denote by $L(G)$ the set of the vertices of G that have degree exactly 1. If G is a tree (i.e., a connected acyclic graph) then $L(G)$ is the set of the *leaves* of G. A *vertex labeling* of G is some injection $\rho : V(G) \to \mathbb{N}^+$. Given a vertex $v \in V(G)$, we define the *neighborhood* of v as $N_G(v) = \{u \mid u \in V(G), \{u, v\} \in E(G)\}$ and the *closed neighborhood* of v as $N_G[v] = N_G(v) \cup \{v\}$. If $X \subseteq V(G)$, then we write $N_G(X) = (\bigcup_{v \in X} N_G(v)) \setminus X$. The *degree* of a vertex v in G is defined as $\mathsf{deg}_G(v) = |N_G(v)|$. A graph is called *subcubic* if all its vertices have degree at most 3.

A *subgraph* $H = (V_H, E_H)$ of a graph $G = (V, E)$ is a graph such that $V_H \subseteq V(G)$ and $E_H \subseteq E(G) \cap \binom{V(H)}{2}$. If $S \subseteq V(G)$, the subgraph of G *induced by* S, denoted $G[S]$, is the graph $(S, E(G) \cap \binom{S}{2})$. We also define $G \setminus S$ to be the subgraph of G induced by $V(G) \setminus S$. If $S \subseteq E(G)$, we denote by $G \setminus S$ the graph $(V(G), E(G) \setminus S)$.

If $s, t \in V(G)$, an (s, t)*-path* of G is any connected subgraph P of G with maximum degree 2 and where $s, t \in L(P)$. We finally denote by $\mathcal{P}(G)$ the set of all paths of G. Given $P \in \mathcal{P}(G)$, we say that $v \in V(P)$ is an *internal vertex* of P if $\mathsf{deg}_P(v) = 2$. Given an integer i and a graph G, we say that G is i-connected if for each $\{u, v\} \in \binom{V(G)}{2}$, there exists a set $\mathcal{Q} \subseteq \mathcal{P}(G)$ of (u, v)-paths of G such that $|\mathcal{Q}| = i$ and for each $P_1, P_2 \in \mathcal{Q}$ such that $P_1 \neq P_2$, $V(P_1) \cap V(P_2) = \{u, v\}$.

We denote by K_r the complete graph on r vertices, by K_{r_1,r_2} the complete bipartite graph where the one part has r_1 vertices and the other r_2, and by $K_{2,r}^+$ the graph obtained if we take $K_{2,r}$ and add an edge between the two vertices of the part of size 2. Finally we denote by P_k and C_k the path and cycles of k vertices respectively. We define the *diamond* to be the graph $K_{2,2}^+$.

Minors and topological minors. Given two graphs H and G and two functions $\phi : V(H) \to V(G)$ and $\sigma : E(H) \to \mathcal{P}(G)$, we say that (ϕ, σ) is *a topological minor model of H in G* if

- for every $\{x, y\} \in E(H)$, $\sigma(\{x, y\})$ is an $(\phi(x), \phi(y))$-path in G and
- if P_1, P_2 are two distinct paths in $\sigma(E(H))$, then none of the internal vertices of P_1 is a vertex of P_2.

The *branch* vertices of (ϕ, σ) are the vertices in $\phi(V(E))$, while the *subdivision* vertices of (ϕ, σ) are the internal vertices of the paths in $\sigma(E(H))$.

We say that G contains H as a *topological minor*, denoted by $H \preceq_{\mathsf{tm}} G$, if there is a topological minor model (ϕ, σ) of H in G.

Given two graphs H and G and a function $\phi : V(H) \to 2^{V(G)}$, we say that ϕ is *a minor model of H in G* if

- for every $x \in V(H)$, $G[\phi(x)]$ is a connected non-empty graph and
- for every $\{x, y\} \in E(H)$, there exist $x' \in \phi(x)$ and $y' \in \phi(y)$ such that $\{x', y'\} \in E(G)$.

We say that G contains H as a *minor*, denoted by $H \preceq_{\mathsf{m}} G$, if there is a minor model ϕ of H in G.

Proof of Lemma 1. We prove the lemma for the topological minor version, and the minor version can be proved with the same kind of arguments. Let i, H, G, and S be defined as in the statement of the lemma. Assume that $H \preceq_{\mathsf{tm}} G$ and let (ϕ, σ) be a topological minor model of H in G. If S is not a separator of G, then the statement is trivial, as in that case the cut-clique of $(G \setminus S, G, S)$ is a supergraph of G. Suppose henceforth that S is a separator of G, and assume for contradiction that there exist two connected components G_1 and G_2 of $G \setminus S$ and two distinct vertices x_1 and x_2 of H such that $\phi(x_1) \in V(G_1)$ and $\phi(x_2) \in V(G_2)$. Then, as H is i-connected, there should be i internally vertex-disjoint paths from $\phi(x_1)$ to $\phi(x_2)$ in G. As S is a separator of size at most $i - 1$, this is not possible. Thus, there exists a connected component G' of $G \setminus S$ such that for each $x \in V(H)$, $\phi(x) \in V(G') \cup S$. This implies that H is a topological minor of the cut-clique of (G', G, S). ◀

Proof of Lemma 2. Let G, v, V, and H be defined as in the statement of the lemma. Let $B \in L(\mathsf{bct}(H))$. If B is a single edge, then the condition $B \not\preceq_{\mathsf{tm}} G[V \cup \{v\}]$ implies that $V = \emptyset$. But V is the vertex set of a connected component of $G \setminus \{v\}$ and so $V \neq \emptyset$. This implies that the case B is a single edge cannot occur. If B is not a simple edge, then by definition B is 2-connected and then, by Lemma 1, $B \preceq_{\mathsf{tm}} G \setminus V$. This implies that there is a topological minor model (ϕ, σ) of H in G such that for each $B \in L(\mathsf{bct}(H))$ and for each $b \in B$, $\phi(b) \notin V$.

We show now that for each $x \in V(H)$, $\phi(x) \notin V$. If $V(H) \setminus (\bigcup_{B \in L(\mathsf{bct}(H))} V(B)) = \emptyset$ then the result is already proved. Otherwise, let $x \in V(H) \setminus (\bigcup_{B \in L(\mathsf{bct}(H))} V(B))$. By definition of the block-cut tree, there exist b_1 and b_2 in $\bigcup_{B \in L(\mathsf{bct}(H))} V(B)$ such that x lies on a (b_1, b_2)-path P of $\mathcal{P}(H)$. Let P_i be the (b_i, x)-subpath of P for each $i \in \{1, 2\}$. By definition of P, we have that $V(P_1) \cap V(P_2) = \{x\}$. This implies that there exists a $(\phi(b_1), \phi(x))$-path P_1' and a $(\phi(b_2), \phi(x))$-path P_2' in $\mathcal{P}(G)$ such that $V(P_1') \cap V(P_2') = \{\phi(x)\}$. Then, as v is a cut vertex of G, it follows that $\phi(x) \notin V$. Thus, for each $x \in V(H)$, $\phi(x) \notin V$. Let $\{x, y\}$ be an edge of $E(H)$. As $\sigma(\{x, y\})$ is a simple $(\phi(x), \phi(y))$-path, both $\phi(x)$ and $\phi(y)$ are not in V and v is a cut vertex of G, we have, with the same argumentation that before that, for each $z \in V(\sigma(\{x, y\})$, $z \notin V$. This concludes the proof. ◀

In the above lemma, we have required graph H to be connected so that $\mathsf{bct}(H)$ is well-defined, but we could relax this requirement, and replace in both statements "for each leaf B of $\mathsf{bct}(H)$" with "for each connected component H' of H and each leaf B of $\mathsf{bct}(H')$".

Tree decompositions. A *tree decomposition* of a graph G is a pair $\mathcal{D} = (T, \mathcal{X})$, where T is a tree and $\mathcal{X} = \{X_t \mid t \in V(T)\}$ is a collection of subsets of $V(G)$ such that:

- $\bigcup_{t \in V(T)} X_t = V(G)$,
- for every edge $\{u, v\} \in E$, there is a $t \in V(T)$ such that $\{u, v\} \subseteq X_t$, and

- for each $\{x, y, z\} \subseteq V(T)$ such that z lies on the unique path between x and y in T, $X_x \cap X_y \subseteq X_z$.

We call the vertices of T *nodes* of $\mathcal{D}$ and the sets in $\mathcal{X}$ *bags* of $\mathcal{D}$. The *width* of a tree decomposition $\mathcal{D} = (T, \mathcal{X})$ is $\max_{t \in V(T)} |X_t| - 1$. The *treewidth* of a graph G, denoted by $\mathbf{tw}(G)$, is the smallest integer w such that there exists a tree decomposition of G of width at most w. For each $t \in V(T)$, we denote by E_t the set $E(G[X_t])$.

Parameterized complexity. We refer the reader to [8, 11] for basic background on parameterized complexity, and we recall here only some very basic definitions. A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$. For an instance $I = (x, k) \in \Sigma^* \times \mathbb{N}$, k is called the *parameter*. A parameterized problem is *fixed-parameter tractable* (FPT) if there exists an algorithm $\mathcal{A}$, a computable function f, and a constant c such that given an instance $I = (x, k)$, $\mathcal{A}$ (called an FPT *algorithm*) correctly decides whether $I \in L$ in time bounded by $f(k) \cdot |I|^c$.

B Boundaried graphs

Boundaried graphs. Let $t \in \mathbb{N}$. A *t-boundaried graph* is a triple $\mathbf{G} = (G, R, \lambda)$ where G is a graph, $R \subseteq V(G)$, $|R| = t$, and $\lambda : R \to \mathbb{N}^+$ is an injective function. We call R the *boundary* of $\mathbf{G}$ and we call the vertices of R *the boundary vertices* of $\mathbf{G}$. We also call G *the underlying graph* of $\mathbf{G}$. Moreover, we call $t = |R|$ the *boundary size* of $\mathbf{G}$ and we define the *label set* of $\mathbf{G}$ as $\Lambda(\mathbf{G}) = \lambda(R)$. We also say that $\mathbf{G}$ is a *boundaried graph* if there exists an integer t such that $\mathbf{G}$ is an t-boundaried graph. We say that a boundary graph $\mathbf{G}$ is *consecutive* if $\Lambda(\mathbf{G}) = [1, |R|]$. We define the *size* of $\mathbf{G} = (G, R, \lambda)$, as $|V(G)|$ and we use the notation $V(\mathbf{G})$ and $E(\mathbf{G})$ for $V(G)$ and $E(G)$, respectively. If $S \subseteq V(G)$, we define $\mathbf{G}' = \mathbf{G} \setminus S$ such that $\mathbf{G}' = (G', R', \lambda')$, $G' = G \setminus S$, $R' = R \setminus S$, and $\lambda' = \lambda|_{R'}$. We define $\mathcal{B}^{(t)}$ as the set of all t-boundaried graphs. We also use the notation $\mathbf{B}_\varnothing = ((\emptyset, \{\emptyset\}), \emptyset, \varnothing)$ to denote the (unique) 0-boundaried *empty boundaried graph*.

Given a t-boundaried graph $\mathbf{G} = (G, R, \lambda)$, we define $\psi_\mathbf{G} : R \to [1, t]$ such that for each $v \in R$, $\psi_\mathbf{G}(v) = |\{u \in R \mid \lambda(u) \leq \lambda(v)\}|$. Note that, as λ is an injective function, $\psi_\mathbf{G}$ is a bijection and, given a boundary vertex v of $\mathbf{G}$, we call $\psi_\mathbf{G}(v)$ the *index* of v.

Let $t \in \mathbb{N}$. We say that two t-boundaried graphs $\mathbf{G}_1 = (G_1, R_1, \lambda_1)$ and $\mathbf{G}_2 = (G_2, R_2, \lambda_2)$ are *isomorphic* if there is a bijection $\sigma : V(\mathbf{G}_1) \to V(\mathbf{G}_2)$ that is an isomorphism $\sigma : V(G_1) \to V(G_2)$ from G_1 to G_2 and additionally $\psi_{\mathbf{G}_1}^{-1} \circ \psi_{\mathbf{G}_2} \subseteq \sigma$, i.e., σ sends the boundary vertices of $\mathbf{G}_1$ to equally-indexed boundary vertices of $\mathbf{G}_2$. We say that $\mathbf{G}_1$ and $\mathbf{G}_2$ are *boundary-isomorphic* if $\psi_{\mathbf{G}_1}^{-1} \circ \psi_{\mathbf{G}_2}$ is an isomorphism from $G_1[R_1]$ to $G_2[R_2]$ and we denote this fact by $\mathbf{G}_1 \sim \mathbf{G}_2$. It is easy to make the following observation.

▶ **Observation 3.** *For every $t \in \mathbb{N}$, if $\mathcal{S}$ is a collection of t-boundaried graphs where $|\mathcal{S}| > 2^{\binom{t}{2}}$, then $\mathcal{S}$ contains at least two boundary-isomorphic graphs.*

Topological minors of boundaried graphs. Let $\mathbf{G}_1 = (G_1, R_1, \lambda_1)$ and $\mathbf{G}_2 = (G_2, R_2, \lambda_2)$ be two boundaried graphs. We say that $\mathbf{G}_1$ is a *topological minor* of $\mathbf{G}_2$ if there is a topological minor model (ϕ, σ) of G_1 in G_2 such that

- $\psi_{\mathbf{G}_1} = \psi_{\mathbf{G}_2} \circ \phi|_{R_1}$, i.e., the vertices of R_1 are mapped via ϕ to equally indexed vertices of R_2 and
- none of the vertices in $R_2 \setminus \phi(R_1)$ is a subdivision vertex of (ϕ, σ).

Operations on boundaried graphs. Let $\mathbf{G}_1 = (G_1, R_1, \lambda_1)$ and $\mathbf{G}_2 = (G_2, R_2, \lambda_2)$ be two t-boundaried graphs. We define the *gluing operation* $\oplus$ such that $(G_1, R_1, \lambda_1) \oplus (G_2, R_2, \lambda_2)$

is the graph G obtained by taking the disjoint union of G_1 and G_2 and then, for each $i \in [1,t]$, identifying the vertex $\psi^{-1}_{\mathbf{G}_1}(i)$ and the vertex $\psi^{-1}_{\mathbf{G}_2}(i)$. Keep in mind that $\mathbf{G}_1 \oplus \mathbf{G}_2$ is a graph and not a boundaried graph. Moreover, the operation $\oplus$ requires both boundaried graphs to have boundaries of the same size.

Let $\mathbf{G} = (G, R, \lambda)$ be a t-boundaried graph and let $I \subseteq \mathbb{N}$. We denote $\mathbf{G}|_I = (G, \lambda^{-1}(I), \lambda|_{\lambda^{-1}(I)})$, i.e., we do not include in the boundary anymore the vertices that are not indexed by numbers in I. Clearly, $\mathbf{G}|_I$ is a t'-boundaried graph where $t' = |I \cap \Lambda(\mathbf{G})|$.

Let $\mathbf{G}_1 = (G_1, R_1, \lambda_1)$ and $\mathbf{G}_2 = (G_2, R_2, \lambda_2)$ be two boundaried graphs. Let also $I = \lambda_1(R_1) \cap \lambda_2(R_2)$ and let $t = |R_1| + |R_2| - |I|$. We define the *merging operation* $\odot$ such that $(G_1, R_1, \lambda_1) \odot (G_2, R_2, \lambda_2)$ is the t-boundaried graph $G = (G, R, \lambda)$ where G is obtained by taking the disjoint union of G_1 and G_2 and then for each $i \in I$ identify the vertex $\lambda_1^{-1}(i)$ with the vertex $\lambda_2^{-1}(i)$. Similarly, R is the obtained by $R_1 \cup R_2$ after applying the same identifications to pairs of vertices in R_1 and R_2. Finally, $\lambda = \lambda'_1 \cup \lambda'_2$ where, for $j \in [1,2]$, λ'_j is obtained from λ_j after replacing each $(x,i) \in \lambda_j$ (for some $i \in I$) by (x_{new}, i), where x_{new} is the result of the identification of $\lambda_1^{-1}(i)$ and $\lambda_2^{-1}(i)$. Observe that $\mathbf{G}_1 \odot \mathbf{G}_2$ is a boundaried graph and that the operation $\odot$ does not require input boundaried graphs to have boundaries of the same size.

Let $\mathbf{G} = (G, R, \lambda)$ be a consecutive t-boundaried graph and let $I \subseteq \mathbb{N}$ be such that $|I| = t$. We define $\mathbf{G} = (G, R, \lambda) \diamond I$ as the unique t-boundaried graph $\mathbf{G}' = (G, R, \lambda')$ where $\lambda' : R \to I$ is a bijection and $\psi_{\mathbf{G}'} = \lambda$.

Equivalence relations. Let $\mathcal{F}$ be a regular collection and let t be a non-negative integer. We define an equivalence relation $\equiv^{(\mathcal{F},t)}$ on t-boundaried graphs as follows: Given two t-boundaried graphs $\mathbf{G}_1$ and $\mathbf{G}_2$, we write $\mathbf{G}_1 \equiv^{(\mathcal{F},t)} \mathbf{G}_2$ to denote that

$$\forall \mathbf{G} \in \mathcal{B}^{(t)} \quad \mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{G}_1 \iff \mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{G}_2.$$

It is easy to verify that $\equiv^{(\mathcal{F},t)}$ is an equivalent relation. We set up a *set of representatives* $\mathcal{R}^{(\mathcal{F},t)}$ as a set containing, for each equivalent class $\mathcal{C}$ of $\equiv^{(\mathcal{F},t)}$, some consecutive t-boundaried graph in $\mathcal{C}$ with minimum number of edges and no isolated vertices out of its boundary (if there are more than one such graphs, pick one arbitrarily). Given a t-boundaried graph $\mathbf{G}$ we denote by $\mathsf{rep}^{(\mathcal{F})}(\mathbf{G})$ the t-boundaried graph $\mathbf{B} \in \mathcal{R}^{(\mathcal{F},t)}$ where $\mathbf{B} \equiv^{(\mathcal{F},t)} \mathbf{G}$ and we call $\mathbf{B}$ the *$\mathcal{F}$-representative of* $\mathbf{G}$. Clearly, $\mathsf{rep}^{(\mathcal{F})}(\mathbf{B}) = \mathbf{B}$.

Note that if $\mathbf{B} = (B, R, \lambda)$ is a t-boundaried graph and $\mathcal{F} \preceq_{\mathsf{tm}} B$, then $\mathsf{rep}^{(\mathcal{F})}(\mathbf{B})$ is, by definition, a consecutive t-boundaried graph whose underlying graph is a graph $H \in \mathcal{F}$ with minimum number of edges, possibly completed with $t - |V(H)|$ isolated vertices in the case where $|V(H)| < t$. We denote this graph by $\mathbf{F}^{(\mathcal{F},t)}$ (if there are many possible choices, just pick one arbitrarily). Note also that the underlying graph of every boundaried graph in $\mathcal{R}^{(\mathcal{F},t)} \setminus \{\mathbf{F}^{(\mathcal{F},t)}\}$ belongs in $\mathsf{ex}_{\mathsf{tm}}(\mathcal{F})$.

We need the following three lemmata. The first one is a direct consequence of the definitions of the equivalence relation $\equiv^{(\mathcal{F},t)}$ and the set of representatives $\mathcal{R}^{(\mathcal{F},t)}$.

▶ **Lemma 22.** *Let $\mathcal{F}$ be a regular collection and let $t \in \mathbb{N}$. Let also $\mathbf{B}_1$ and $\mathbf{B}_2$ be t-boundaried graphs. Then $\mathbf{B}_1 \equiv^{(\mathcal{F},t)} \mathbf{B}_2$ if an only if $\forall \mathbf{G} \in \mathcal{R}^{(\mathcal{F},t)}$ $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_1 \iff \mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_2$.*

▶ **Lemma 23.** *Let $\mathcal{F}$ be a connected collection and let $t \in \mathbb{N}$. Let also $\mathbf{B} \in \mathcal{R}^{(\mathcal{F},t)}$. Then every connected component of the underlying graph of $\mathbf{B}$ intersects its boundary set.*

Proof. Let $\mathbf{B} = (B, R, \lambda) \in \mathcal{R}^{(\mathcal{F},t)}$. As the lemma follows directly in the case where $\mathbf{B} = \mathbf{F}^{(\mathcal{F},t)}$, we may assume that $\mathcal{F} \not\preceq_{\mathsf{tm}} B$. We assume, towards a contradiction, that B has a component C whose vertex set does not contain any of the vertices of R. This means that B can be seen as the disjoint union of C and $B' = B \setminus V(C)$. As $\mathcal{F} \not\preceq_{\mathsf{tm}} B$, we also have that $\mathcal{F} \not\preceq_{\mathsf{tm}} C$. Let now $\mathbf{B}' = (B', R, \lambda)$. Clearly $|E(\mathbf{B}')| < |E(\mathbf{B})|$. We will arrive to a contradiction by proving that $\mathbf{B}' \equiv^{(\mathcal{F},t)} \mathbf{B}$. Let $\mathbf{G} \in \mathcal{B}^{(t)}$. Note that $\mathbf{G} \oplus \mathbf{B}$ is the disjoint union of $\mathbf{G} \oplus \mathbf{B}'$ and C. As all graphs in $\mathcal{F}$ are connected, it follows that a (connected) graph $H \in \mathcal{F}$ is a topological minor of $\mathbf{G} \oplus \mathbf{B}$ if and only if H is a topological minor of $\mathbf{G} \oplus \mathbf{B}'$. We conclude that $\mathbf{B}' \equiv^{(\mathcal{F},t)} \mathbf{B}$, a contradiction. ◀

▶ **Lemma 24.** *Let $\mathcal{F}$ be a connected collection. Then, for every graph B, it holds that $\mathsf{rep}^{(\mathcal{F})}((B, \emptyset, \varnothing)) = \mathbf{B}_{\varnothing}$ if and only if $\mathcal{F} \not\preceq_{\mathsf{tm}} G$.*

Proof. Let $\mathbf{B} = (B, \emptyset, \varnothing)$ where B is a graph. Recall that if $\mathcal{F} \preceq_{\mathsf{tm}} B$, then $\mathsf{rep}^{(\mathcal{F})}(\mathbf{B}) = \mathbf{F}^{(\mathcal{F},t)}$. As $\mathcal{F}$ does not contain the empty graph, we have that $\mathbf{F}^{(\mathcal{F},t)} \neq \mathbf{B}_{\varnothing}$, therefore $\mathsf{rep}^{(\mathcal{F})}(\mathbf{B}) \neq \mathbf{B}_{\varnothing}$.

Suppose now that $\mathcal{F} \not\preceq_{\mathsf{tm}} B$. We have to prove that for every $\mathbf{G} \in \mathcal{B}^{(0)}$, $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B} \iff \mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_{\varnothing}$. Let $\mathbf{G} = (G, \emptyset, \varnothing) \in \mathcal{B}^{(0)}$. Note that $\mathbf{G} \oplus \mathbf{B}$ is the disjoint union of G and B and that $\mathbf{G} \oplus \mathbf{B}_{\varnothing} = G$. As $\mathcal{F} \not\preceq_{\mathsf{tm}} B$ and $\mathcal{F}$ is connected, it follows that the disjoint union of B and G contains some (connected) graph in $\mathcal{F}$ if and only if B does. This implies that $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B} \iff \mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_{\varnothing}$, as required. ◀

Folios. Let $\mathcal{F}$ be a regular collection. Given $t, r \in \mathbb{N}$, we define $\mathcal{A}^{(t)}_{\mathcal{F},r}$ as the set of all pairwise non-isomorphic boundaried graphs that contain at most r non-boundary vertices, whose label set is a subset of $[1, t]$, and whose underlying graph belongs in $\mathsf{ex}_{\mathsf{tm}}(\mathcal{F})$. Note that a graph in $\mathcal{A}^{(t)}_{\mathcal{F},r}$ is not necessarily a t-boundaried graph.

Given a t-boundaried graph $\mathbf{B}$ and an integer $r \in \mathbb{N}$, we define the $(\mathcal{F}, r)$-folio of $\mathbf{B}$, denoted by $\mathsf{folio}(\mathbf{B}, \mathcal{F}, r)$ the set containing all boundaried graphs in $\mathcal{A}^{(t)}_{\mathcal{F},r}$ that are topological minors of $\mathbf{B}$. Moreover, in case $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{B}$, we also include in $\mathsf{folio}(\mathbf{B}, \mathcal{F}, r)$ the graph $\mathbf{F}^{(\mathcal{F},t)}$.

We also define $\mathfrak{F}^{(t)}_{\mathcal{F},r} = 2^{\mathcal{A}^{(t)}_{\mathcal{F},r} \cup \{\mathbf{F}^{(\mathcal{F},t)}\}}$ and notice that $\{\mathsf{folio}(\mathbf{B}, \mathcal{F}, r) \mid \mathbf{B} \in \mathcal{B}^{(t)}\} \subseteq \mathfrak{F}^{(t)}_{\mathcal{F},r}$, i.e., $\mathfrak{F}^{(t)}_{\mathcal{F},r}$ contains all different $(\mathcal{F}, r)$-folios of t-boundaried graphs.

▶ **Lemma 25.** *Let $t \in \mathbb{N}$ and let $\mathcal{F}$ be a regular collection. For every t-boundaried graph $\mathbf{B}$ and every $r \in \mathbb{N}$, it holds that $|\mathsf{folio}(\mathbf{B}, \mathcal{F}, r)| = 2^{\mathcal{O}_{r+d}(t \log t)}$, where $d = \mathsf{size}(\mathcal{F})$. Moreover, $|\mathfrak{F}^{(t)}_{\mathcal{F},r}| = 2^{2^{\mathcal{O}_{r+d}(t \log t)}}$.*

Proof. Let $t \in \mathbb{N}$, let $\mathcal{F}$ be a regular collection, let $r \in \mathbb{N}$, and let $n = t + r$. We prove a stronger result, namely that $|\mathcal{A}^{(t)}_{\mathcal{F},r}| = 2^{\mathcal{O}_{r+d}(t \log t)}$. The claimed bound on $|\mathfrak{F}^{(t)}_{\mathcal{F},r}|$ then follows directly by definition of the set $\mathfrak{F}^{(t)}_{\mathcal{F},r}$. By [21], there exists a constant c such that for each $G \in \mathsf{ex}_{\mathsf{tm}}(\mathcal{F})$, $|E(G)| \leq c \cdot |V(G)|$. By definition, every underlying graph of an element of $\mathcal{A}^{(t)}_{\mathcal{F},r}$ is in $\mathsf{ex}_{\mathsf{tm}}(\mathcal{F})$. If we want to construct an element $\mathbf{G} = (G, R, \lambda)$ of $\mathcal{A}^{(t)}_{\mathcal{F},r}$ with at most n vertices, then there are asymptotically at most $c \cdot n \cdot \binom{n^2}{c \cdot n} \leq c \cdot n^{1+2 \cdot c \cdot n}$ choices for the edge set $E(G)$, at most $t \cdot \binom{n}{t} \leq t \cdot n^t$ choices for R, and $t^{|R|} \leq t^t$ choices for the function λ. We obtain that $\mathcal{A}^{(t)}_{\mathcal{F},r}$ is of size at most $n \cdot 2^{(1+2 \cdot c \cdot n) \log n} \cdot 2^{t \log t} = 2^{\mathcal{O}_{r+d}(t \log t)}$, and the lemma follows. ◀

The following lemma indicates that folios define a refinement of the equivalence relation $\equiv^{(\mathcal{F},t)}$.

▶ **Lemma 26.** *Let $\mathcal{F}$ be a regular collection and let $d = \mathsf{size}(\mathcal{F})$. Let also $\mathbf{B}_1$ and $\mathbf{B}_2$ be two t-boundaried graphs. If $\mathsf{folio}(\mathbf{B}_1, \mathcal{F}, d) = \mathsf{folio}(\mathbf{B}_2, \mathcal{F}, d)$, then $\mathbf{B}_1 \equiv^{(\mathcal{F},t)} \mathbf{B}_2$.*

Proof. Let $\mathbf{B}_1$ and $\mathbf{B}_2$ be two t-boundaried graphs such that $\mathsf{folio}(\mathbf{B}_1, \mathcal{F}, d) = \mathsf{folio}(\mathbf{B}_2, \mathcal{F}, d)$. We fix $\mathbf{G} \in \mathcal{B}^{(t)}$, and we need to prove that $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_1$ if and only if $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_2$.

Assume first that $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_1$. Then there exists a graph $F \in \mathcal{F}$ and a topological minor model (ϕ, σ) of F in $\mathbf{G} \oplus \mathbf{B}_1$. This topological minor model (ϕ, σ) can be naturally decomposed into two topological minor models (ϕ_0, σ_0) and (ϕ_1, σ_1) of two graphs F_0 and F_1 in $\mathcal{A}^{(t)}_{\mathcal{F},d}$, respectively, with $F_0 \odot F_1 = F$, such that (ϕ_0, σ_0) (resp. (ϕ_1, σ_1)) is a topological minor model of F_0 (resp. F_1) in the (boundaried) graph $\mathbf{G}$ (resp. $\mathbf{B}_1$). Since $\mathsf{folio}(\mathbf{B}_1, \mathcal{F}, d) = \mathsf{folio}(\mathbf{B}_2, \mathcal{F}, d)$, there exists a topological minor model (ϕ_2, σ_2) of F_1 in $\mathbf{B}_2$. Combining the topological minor models (ϕ_0, σ_0) and (ϕ_2, σ_2) gives rise to a topological minor model (ϕ', σ') of F in $\mathbf{G} \oplus \mathbf{B}_2$, and therefore $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_2$.

Conversely, assume that $\mathcal{F} \not\preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_1$, and assume for contradiction that there exists a graph $F \in \mathcal{F}$ and a topological minor model (ϕ, σ) of F in $\mathbf{G} \oplus \mathbf{B}_2$. Using the same arguments as above, (ϕ, σ) implies the existence of a topological minor model (ϕ', σ') of F in $\mathbf{G} \oplus \mathbf{B}_1$, contradicting the hypothesis that $\mathcal{F} \not\preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_1$. ◀

Lemmata 25 and 26 directly imply Lemma 8.

Branch decompositions of boundaried graphs. Let $\mathbf{G} = (G, R, \lambda)$ be a boundaried graph and let ρ be a vertex labeling of G where $\lambda \subseteq \rho$.

A *branch decomposition* of $\mathbf{G}$ is a pair (T, σ) where T is a ternary tree and $\sigma : E(G) \cup \{R\} \to L(T)$ is a bijection. Let $r = \sigma(R)$ and let e_r be the unique edge in T that is incident to r. We call r *the root* of T. Given an edge $e \in E(T)$, we define T_e as the one of the two connected components of $T \backslash \{e\}$ that does not contain the root r. We then define $\mathbf{G}_e = (G_e, R_e, \lambda_e)$ where $E(G_e) = \sigma^{-1}(L(T_e) \cap L(T))$, $V(G_e) = \bigcup E(G_e)$, R_e is the set containing every vertex of G that is an endpoint of an edge in $E(G_e)$ and also belongs in a set in $\{R\} \cup (E(G) \setminus E(G_e))$ (here we treat edges in $E(G) \setminus E(G_e)$ as 2-element sets), and $\lambda_e = \rho|_{R_e}$. We also set $t_e = |R_e|$ and observe that $\mathbf{G}_e$ is a t_e-boundaried graph. The *width* of (T, σ) is $\max\{t_e \mid e \in E(T)\}$. The *branchwidth* of $\mathbf{G}$, denoted by $\mathbf{bw}(\mathbf{G})$, is the minimum width over all branch decompositions of $\mathbf{G}$.

This is an extension of the definition of a branch decomposition on graphs, given in [28], to boundaried graphs. Indeed, of G is a graph, then a *branch decomposition* of G is a branch decomposition of $(G, \emptyset, \varnothing)$. We also define the *branchwidth* of G as $\mathbf{bw}(G) = \mathbf{bw}(G, \emptyset, \varnothing)$.

▶ **Lemma 27.** *Let $\mathbf{G} = (G, R, \lambda)$ be a boundaried graph. Then $\mathbf{bw}(\mathbf{G}) \leq \mathbf{bw}(G) + |R|$.*

Proof. Let (T', σ') be a branch decomposition of $\mathbf{G}' = (G, \emptyset, \varnothing)$ and let r be the root of T'. Recall that $\mathbf{G}'_e = (G'_e, R'_e, \lambda'_e), e \in E(T')$. We construct a branch decomposition (T, σ) of $\mathbf{G} = (G, R, \lambda)$ as follows: we set $T = T'$ and $\sigma = (\sigma' \setminus \{(\emptyset, r)\}) \cup \{(R, r)\}$. Note that $\mathbf{G}_e = (G'_e, R_e, \lambda_e), e \in E(T)$, where $R_e \subseteq R'_e \cup R$. This means that $|R_e| \leq |R'_e| + |R|$, therefore $\mathbf{bw}(\mathbf{G}) \leq \mathbf{bw}(G) + |R|$. ◀

The following proposition is a combination of the single-exponential linear-time constant-factor approximation of treewidth by Bodlaender *et al.* [4], with the fact that any graph G with $|E(G)| \geq 3$ satisfies that $\mathbf{bw}(G) \leq \mathbf{tw}(G) + 1 \leq \frac{3}{2}\mathbf{bw}(G)$ [28]; it is worth noting that from the proofs of these inequalities, simple polynomial-time algorithms for transforming a branch (resp. tree) decomposition into a tree (resp. branch) decompositions can be derived.

▶ **Proposition 28.** *There exists an algorithm that receives as input a graph G and a $w \in \mathbb{N}$ and either reports that $\mathbf{bw}(G) > w$ or outputs a branch decomposition (T, σ) of G of width $\mathcal{O}(w)$. Moreover, this algorithm runs in $2^{\mathcal{O}(w)} \cdot n$ steps.*

Proof of Lemma 9. Let $G \in \mathsf{ex}_{\mathsf{tm}}(\mathcal{F})$ and let $F \in \mathcal{F}$ be a planar subcubic graph. Since F is subcubic and $F \npreceq_{\mathsf{tm}} G$, it follows (see [10]) that $F \npreceq_{\mathsf{m}} G$, and since F is planar this implies by [27] that $\mathbf{tw}(G)$, hence $\mathbf{bw}(G)$ as well, is bounded by a function depending only on F. ◀

C Proof of Theorem 3

We provide a dynamic programming algorithm for the computation of $\mathbf{tm}_{\mathcal{F}}(G)$ for the general case where $\mathcal{F}$ is a regular collection. We first consider an, arbitrarily chosen, vertex labeling ρ of G. From Lemma 28, we may assume that we have a branch decomposition (T, σ) of $(G, \emptyset, \varnothing)$ of width $\mathcal{O}(w)$, where $w = \mathbf{tw}(G)$. This gives rise to the t_e-boundaried graphs $\mathbf{G}_e = (G_e, R_e, \lambda_e)$ for each $e \in E(T)$. Moreover, if r is the root of T, $\sigma^{-1}(r) = \emptyset = R_{e_r}$ and $\mathbf{G}_{e_r} = (G, \emptyset, \varnothing)$. Keep also in mind that $t_e = \mathcal{O}(w)$ for every $e \in E(T)$.

For each $e \in E(T)$, we say that $(L, \mathcal{C})$ is an *e-pair* if $L \subseteq R_e$ and $\mathcal{C} \in \mathfrak{F}^{(t'_e)}_{\mathcal{F},d}$ where $t'_e = t_e - |L|$. We also denote by $\mathcal{P}_e$ the set of all e-pairs. Clearly, $|\mathcal{P}_e| = \sum_{i \in [0,t_e]} 2^i \cdot |\mathfrak{F}^{(t_e - i)}_{\mathcal{F},d}|$, therefore, from Lemma 25, $|\mathcal{P}_e| = 2^{2^{\mathcal{O}_d(w \log w)}}$.

We then define the function $\mathbf{tm}^{(e)}_{\mathcal{F}} : \mathcal{P}_e \to \mathbb{N}$ such that if $(L, \mathcal{C}) \in \mathcal{P}_e$, then

$$\mathbf{tm}^{(e)}_{\mathcal{F}}(L, \mathcal{C}) = \min\{|S| \mid S \subseteq V(G_e) \wedge L = R_e \cap S \ \wedge \ \mathcal{C} = \mathsf{folio}(\mathbf{G}_e \setminus S, d)\} \cup \{\infty\}.$$

Note that $\mathcal{P}_{e_r} = \emptyset \times \mathfrak{F}^{(0)}_{\mathcal{F},d}$. Note also that the set $\mathcal{A}^{(0)}_{\mathcal{F},d}$ contains only those that do not contain some graph in $\mathcal{F}$ as a topological minor. Therefore

$$\mathbf{tm}_{\mathcal{F}}(G) = \min\{\mathbf{tm}^{(e_r)}_{\mathcal{F}}(\emptyset, \mathcal{C}) \mid \mathcal{C} \in 2^{\mathcal{A}^{(0)}_{\mathcal{F},d}}\}.$$

Hence, our aim is to give a way to compute $\mathbf{tm}^{(e)}_{\mathcal{F}}$ for every $e \in E(T)$. Our dynamic programming algorithm does this in a bottom-up fashion, starting from the edges that contain as endpoints leaves of T that are different from the root. Let $\ell \in L(T) \setminus \{r\}$ and let e_ℓ be the unique edge of T that contains it. Let also $\sigma^{-1}(e_\ell) = \{x, y\}$. Clearly, $\mathbf{G}_{e_\ell} = (\{x, y\}, \{\{x, y\}\})$ and

$$\mathcal{P}_{e_\ell} = \big\{(\{x, y\}, \mathcal{A}^{0}_{\mathcal{F},d})\big\} \cup (\big\{\{x\}, \{y\}\big\} \times \mathcal{A}^{(1)}_{\mathcal{F},d}) \cup (\{\emptyset\} \times \mathcal{A}^{(2)}_{\mathcal{F},d}).$$

As the size of the elements in $\mathcal{P}_{e_\ell}$ depends only on d, it is possible to compute $\mathbf{tm}^{(e_l)}_{\mathcal{F}}$ in $\mathcal{O}_d(1)$ steps.

Let $e \in \{e_r\} \cup E(T \setminus L(T))$, and let e_1 and e_2 be the two other edges of T that share an endpoint with e and where each path from them to r contains e. We also set

$$F_e = \big(R_{e_1} \cup R_{e_2}\big) \setminus R_e.$$

For the dynamic programming algorithm, it is enough to describe how to compute $\mathbf{tm}^{(e)}_{\mathcal{F}}$ given $\mathbf{tm}^{(e_i)}_{\mathcal{F}}, i \in [1, 2]$. For this, given an e-pair $(L, \mathcal{C}) \in \mathcal{P}_e$ it is possible to verify that

$$\begin{aligned}
\mathbf{tm}^{(e)}_{\mathcal{F}}(L, \mathcal{C}) \quad = \quad \min \big\{ & \mathbf{tm}^{(e_1)}_{\mathcal{F}}(L_1, \mathcal{C}_1) + \mathbf{tm}^{(e_2)}_{\mathcal{F}}(L_2, \mathcal{C}_2) - |L_1 \cap L_2| \mid \\
& (L_i, \mathcal{C}_i) \in \mathcal{P}_{e_i}, i \in [1, 2], \\
& L_i \setminus F_e = L \cap R_{e_i}, i \in [1, 2], \\
& L_1 \cap F_e = L_2 \cap F_e, \text{ and} \\
& \mathcal{C} = \bigcup_{(\mathbf{B}_1, \mathbf{B}_2) \in \mathcal{C}_1 \times \mathcal{C}_2} \mathsf{folio}\Big(\big((\mathbf{B}_1 \diamond Z_1) \odot (\mathbf{B}_2 \diamond Z_2)\big)|_Z, \mathcal{F}, t_e - |L|\Big) \\
& \text{where } Z = \rho(R_e \setminus L) \text{ and } Z_i = \rho(R_{e_i} \setminus L_i), i \in [1, 2]\big\}.
\end{aligned}$$

Note that given $\mathbf{tm}_{\mathcal{F}}^{(e_i)}, i \in [1,2]$ and a $(L, \mathbf{B}) \in \mathcal{P}_e$, the value of $\mathbf{tm}_{\mathcal{F}}^{(e)}(L, \mathbf{B})$ can be computed by the above formula in $\mathcal{O}(|\mathcal{P}_{e_1}| \cdot |\mathcal{P}_{e_2}|) = 2^{2^{\mathcal{O}_d(w \log w)}}$ steps. As $|\mathcal{P}_e| = 2^{2^{\mathcal{O}_d(w \log w)}}$, the computation of the function $\mathbf{tm}_{\mathcal{F}}^{(e)}$ requires again $2^{2^{\mathcal{O}_d(w \log w)}}$ steps. This means that the whole dynamic programming requires $2^{2^{\mathcal{O}_d(w \cdot \log w)}} \cdot |V(T)| = 2^{2^{\mathcal{O}_d(w \log w)}} \cdot |E(G)|$ steps. As $|E(G)| = \mathcal{O}(\mathbf{tw}(G) \cdot |V(G)|)$, the claimed running time follows.

D Proof of Theorem 4

We provide a dynamic programming algorithm for the computation of $\mathbf{tm}_{\mathcal{F}}(G)$. We first consider an, arbitrarily chosen, vertex labeling ρ of G. From Lemma 28, we may assume that we have a branch decomposition (T, σ) of $(G, \emptyset, \varnothing)$ of width at most $w = \mathcal{O}(\mathbf{bw}(G)) = \mathcal{O}(\mathbf{tw}(G))$ (we naturally extend the definition of branch decompositions to boundary graphs – see page 18 of Appendix B). This gives rise to the t_e-boundaried graphs $\mathbf{G}_e = (G_e, R_e, \lambda_e)$ for each $e \in E(T)$. Moreover, if r is the root of T, $\sigma^{-1}(r) = \emptyset = R_{e_r}$ and $\mathbf{G}_{e_r} = (G, \emptyset, \varnothing)$. Keep also in mind that $t_e = \mathcal{O}(w)$ for every $e \in E(T)$.

Our next step is to define the tables of the dynamic programming algorithm. Let $e \in E(T)$. We call the pair $(L, \mathbf{B})$ an *e-pair* if $L \subseteq R_e$ and $\mathbf{B} = (B, R, \lambda) \in \mathcal{R}^{(k', \mathcal{F})}$, where $k' = |R_e \setminus L| = t_e - |L|$. For each $e \in E(T)$, we denote by $\mathcal{P}_e$ the set of all e-pairs. Note that

$$|\mathcal{P}_e| \;=\; \sum_{i \in [0, t_e]} 2^i \cdot |\overline{\mathcal{R}}^{(\mathcal{F}, t_e - i)}| \;=\; (t_e + 1) \cdot 2^{t_e} \cdot 2^{\mathcal{O}_d(t_e \cdot \log t_e)} \;=\; 2^{\mathcal{O}_d(w \cdot \log w)},$$

where we have used Lemma 11 in the second equality. We define the function $\mathbf{tm}_{\mathcal{F}}^{(e)} : \mathcal{P}_e \to \mathbb{N}$ such that if $(L, \mathbf{B}) \in \mathcal{P}_e$, then

$$\mathbf{tm}_{\mathcal{F}}^{(e)}(L, \mathbf{B}) = \min\{|S| \mid S \subseteq V(G_e) \land L = R_e \cap S \;\land\; \mathbf{B} = \mathsf{rep}_{\mathcal{F}}(\mathbf{G}_e \setminus S)\} \cup \{\infty\}.$$

Note that $\mathcal{P}_{e_r} = \{(\emptyset, \mathbf{B}_{\varnothing}), (\emptyset, \mathbf{F}^{(\mathcal{F}, t)})\}$ where $\mathbf{B}_{\varnothing} = ((\emptyset, \emptyset), \emptyset, \varnothing)$. We claim that $\mathbf{tm}_{\mathcal{F}}(G) = \mathbf{tm}_{\mathcal{F}}^{(e_r)}(\emptyset, \mathbf{B}_{\varnothing})$. Indeed,

$$\begin{aligned}
\mathbf{tm}_{\mathcal{F}}(G) &= \min\{|S| \mid \mathcal{F} \not\preceq_{\mathsf{tm}} G \setminus S\} && \text{(from Equation (1))}\\
&= \min\{|S| \mid \mathbf{B}_{\varnothing} = \mathsf{rep}_{\mathcal{F}}((G \setminus S, \emptyset, \varnothing))\} && \text{(from Lemma 24 in page 17)}\\
&= \min\{|S| \mid \emptyset = \emptyset \cap S \;\land\; \mathbf{B}_{\varnothing} = \mathsf{rep}_{\mathcal{F}}(G \setminus S, \emptyset, \varnothing)\}\\
&= \min\{|S| \mid \emptyset = R_{r_e} \cap S \;\land\; \mathbf{B}_{\varnothing} = \mathsf{rep}_{\mathcal{F}}(\mathbf{G}_{e_r} \setminus S)\}\\
&= \mathbf{tm}_{\mathcal{F}}^{(e_r)}(\emptyset, \mathbf{B}_{\varnothing}).
\end{aligned}$$

Therefore, our aim is to give a way to compute $\mathbf{tm}_{\mathcal{F}}^{(e)}$ for every $e \in E(T)$. Our dynamic programming algorithm does this in a bottom-up fashion, starting from the edges that contain as endpoints leaves of T that are different to the root. Let $l \in L(T) \setminus \{r\}$ and let e_ℓ be the edge of T that contains it. Let also $\sigma^{-1}(e_\ell) = \{x, y\}$. Clearly, $\mathbf{G}_{e_\ell} = (\{x, y\}, \{\{x, y\}\})$ and

$$\mathcal{P}_{e_\ell} = \big\{(\{\{x, y\}\} \times \mathcal{R}^{(0, \mathcal{F})}) \cup (\{\{x\}, \{y\}\} \times \mathcal{R}^{(1, \mathcal{F})}) \cup (\{\emptyset\} \times \mathcal{R}^{(2, \mathcal{F})}).$$

As the size of the elements in $\mathcal{P}_{e_\ell}$ depends only on $\mathcal{F}$, it is possible to compute $\mathbf{tm}_{\mathcal{F}}^{(e_l)}$ in $\mathcal{O}_d(1)$ steps. Let $e \in \{e_r\} \cup E(T \setminus L(T))$, and let e_1 and e_2 be the two other edges of T that share an endpoint with e and where each path from them to r contains e. We also set $F_e = \big(R_{e_1} \cup R_{e_2}\big) \setminus R_e$. It is enough to describe how to compute $\mathbf{tm}_{\mathcal{F}}^{(e)}$ given $\mathbf{tm}_{\mathcal{F}}^{(e_i)}, i \in [1,2]$. For this, given an e-pair $(L, \mathbf{B}) \in \mathcal{P}_e$ where $\mathbf{B} = (B, R, \lambda)$, it holds that (see Appendix B for the undefined operations, which just formalize the natural way of gluing objects):

$$\begin{aligned}\mathbf{tm}_{\mathcal{F}}^{(e)}(L,\mathbf{B}) \quad = \quad & \min\big\{\mathbf{tm}_{\mathcal{F}}^{(e_1)}(L_1,\mathbf{B}_1)+\mathbf{tm}_{\mathcal{F}}^{(e_2)}(L_2,\mathbf{B}_2)-|L_1\cap L_2| \mid \\ & (L_i,\mathbf{B}_i)\in\mathcal{P}_{e_i}, i\in[1,2], \\ & L_i\setminus F_e = L\cap R_{e_i}, i\in[1,2], \\ & L_1\cap F_e = L_2\cap F_e, \text{ and} \\ & \mathbf{B}=\mathsf{rep}_{\mathcal{F}}\Big(\big((\mathbf{B}_1\diamond Z_1)\odot(\mathbf{B}_2\diamond Z_2)\big)|_Z, t_e-|L|\Big) \text{ where} \\ & \qquad Z=\rho(R_e\setminus L) \text{ and } Z_i=\rho(R_{e_i}\setminus L_i), i\in[1,2]\big\}.\end{aligned}$$

Note that given $\mathbf{tm}_{\mathcal{F}}^{(e_i)}, i\in[1,2]$ and a $(L,\mathbf{B})\in\mathcal{P}_e$, the value of $\mathbf{tm}_{\mathcal{F}}^{(e)}(L,\mathbf{B})$ can be computed by the above formula in $\mathcal{O}(|\mathcal{P}_{e_1}|\cdot|\mathcal{P}_{e_2}|)=2^{\mathcal{O}_d(w\cdot\log w)}$ steps. As $|\mathcal{P}_e|=2^{\mathcal{O}_d(w\cdot\log w)}$, the computation of the function $\mathbf{tm}_{\mathcal{F}}^{(e)}$ requires again $2^{\mathcal{O}_d(w\cdot\log w)}$ steps. This means that the whole dynamic programming requires $2^{\mathcal{O}_d(w\cdot\log w)}\cdot|E(T)|=2^{\mathcal{O}_d(w\cdot\log w)}\cdot\mathcal{O}(|E(G)|)$ steps. As $|E(G)|=\mathcal{O}(\mathbf{bw}(G)\cdot|V(G)|)$, the claimed running time follows.

E Proof of Lemma 11

Before we proceed with the proof of Lemma 11, we need a series of results. The proof of the following lemma uses ideas similar to the ones presented by Garnero *et al.* [15].

▶ **Lemma 29.** *There is a function $h_2:\mathbb{N}\times\mathbb{N}\to\mathbb{N}$ such that if $\mathcal{F}$ is a connected and planar collection, where $d=\mathsf{size}(\mathcal{F})$, $t\in\mathbb{N}$, $\mathbf{B}=(B,R,\lambda)\in\mathcal{R}^{(\mathcal{F},t)}\setminus\{\mathbf{F}^{(\mathcal{F},t)}\}$, $z\in\mathbb{N}$, and X is a subset of $V(B)$ such that $X\cap R=\emptyset$ and $|N_B(X)|\le z$, then $|X|\le h_2(z,d)$.*

Proof. We set $h_2(z,d)=2^{h_1(d,\mu(d)+z)\cdot(z+\mu(d)+1)+\zeta(\mu(d)+z)-1}+z$, where h_1 is the function of Lemma 8, μ is the function of Lemma 9, and $\zeta:\mathbb{N}\to\mathbb{N}$ such that $\zeta(x)=2^{\binom{x}{2}}$. Let $y=\mu(d)$, $q=h_1(d,y+z)\cdot(x+y+1)\cdot\zeta(y+z)$, $s=h_2(z,d)$, and observe that $s=2^{q-1}+z$. Towards a contradiction, we assume that $|X|>s$.

Let $\mathbf{B}=(B,R,\lambda)\in\mathcal{R}^{(\mathcal{F},t)}\setminus\{\mathbf{F}^{(\mathcal{F},t)}\}$ and let ρ be a vertex-labeling of B where $\lambda\subseteq\rho$. As $\mathbf{B}\neq\mathbf{F}^{(\mathcal{F},t)}$, it follows that

$$B\in\mathsf{ex}_{\mathsf{tm}}(\mathcal{F}). \tag{2}$$

We set $G=B[X\cup N_B(X)]$ and observe that $|V(G)|\ge|X|>s$. As G is a subgraph of B, (2) implies that

$$G\in\mathsf{ex}_{\mathsf{tm}}(\mathcal{F}), \tag{3}$$

and therefore, from Lemma 9, $\mathbf{bw}(G)\le y$. Let $R'=N_B(X)$ and $\lambda'=\rho|_{R'}$. We set $\mathbf{G}=(G,R',\lambda')$. From Lemma 27, $\mathbf{bw}(\mathbf{G})\le\mathbf{bw}(G)+|R'|\le y+|R'|=y+z$.

Note now that G has at most $|R'|$ connected components. Indeed, if it has more, then one of them, say C, will not intersect $|R'|$. This, together with the fact that $R\cap X=\emptyset$, implies that C is also a connected component of B whose vertex set is disjoint from R, a contradiction to Lemma 23. We conclude that $|E(G)|\ge|V(G)|-|R'|\ge|V(G)|-z>s-z=2^{q-1}$.

Let (T,σ) be a branch decomposition of $\mathbf{G}$ of width at most $y+z$. We also consider the graph $\mathbf{G}_e=(G_e,R_e,\lambda_e)$, for each $e\in E(T)$ (recall that $\lambda_e\subseteq\rho$). Observe that

$$\forall e\in E(T),\ |R_e|\le y+z. \tag{4}$$

We define $\mathcal{H}=\{\mathsf{rep}_{\mathcal{F}}(\mathbf{G}_e)\mid e\in E(T)\}$. From (4), $\mathcal{H}\subseteq\bigcup_{i\in[0,y+z]}\mathcal{R}^{(\mathcal{F},i)}$. From Lemma 8, $|\mathcal{H}|\le(y+z+1)\cdot h_1(d,y+z)$, therefore $q\ge|\mathcal{H}|\cdot\zeta(y+z)$. Let r be the root of T and let

P be a longest path in T that has r as an endpoint. As B has more than 2^{q-1} edges, T also has more than 2^{q-1} leaves different from r. This means that P has more than q edges. Recall that $q \geq |\mathcal{H}| \cdot \zeta(y+z)$. As a consequence, there is a set $\mathcal{S} \subseteq \{\mathbf{G}_e \mid e \in E(P)\}$ where $|\mathcal{S}| > \zeta(y+z)$ and $\mathsf{rep}_{\mathcal{F}}(\mathcal{S})$ contains only one boundaried graph (i.e., all the boundaried graphs in $\mathcal{S}$ have the same $\mathcal{F}$-representative). From Observation 3, there are two graphs $\mathbf{G}_{e_1}, \mathbf{G}_{e_2} \in \mathcal{S}$, $e_1 \neq e_2$, such that

$$\mathbf{G}_{e_1} \equiv^{(\mathcal{F},t)} \mathbf{G}_{e_2} \text{ and} \tag{5}$$

$$\mathbf{G}_{e_1} \sim \mathbf{G}_{e_2}. \tag{6}$$

W.l.o.g., we assume that e_1 is in the path in T between r and some endpoint of e_2. This implies that the underlying graph of $\mathbf{G}_{e_1}$ is a proper subgraph of the underlying graph of $\mathbf{G}_{e_2}$, therefore

$$|E(\mathbf{G}_{e_2})| < |E(\mathbf{G}_{e_1})|. \tag{7}$$

Recall that $\mathbf{G}_{e_i} = (G_{e_i}, R_{e_i}, \lambda_{e_i}), i \in [1,2]$. Let $B^- = B \setminus (V(G_{e_1}) \setminus R_{e_1})$ and we set $\mathbf{B}^- = (B^-, R_{e_1}, \lambda_{e_1})$. Clearly, $\mathbf{B}^- \sim \mathbf{G}_{e_1}$. This, combined with (6), implies that

$$\mathbf{B}^- \sim \mathbf{G}_{e_2}. \tag{8}$$

Let now $B^* = \mathbf{B}^- \oplus \mathbf{G}_{e_2}$. Combining (7) and (8), we may deduce that

$$|E(B^*)| < |E(B)|. \tag{9}$$

We now set $\mathbf{B}^* = (B^*, R, \lambda)$ and recall that $t = |R|$. Clearly, both $\mathbf{B}$ and $\mathbf{B}^*$ belong in $\mathcal{B}^{(t)}$.

We now claim that $\mathbf{B} \equiv^{(\mathcal{F},t)} \mathbf{B}^*$. For this, we consider any $\mathbf{D} = (D, R, \lambda) \in \mathcal{B}^{(t)}$. We define $\mathbf{B}^\star = (B^-, R, \lambda)$, $D^+ = \mathbf{D} \oplus \mathbf{B}^\star$, and $\mathbf{D}^+ = (D^+, R_{e_1}, \lambda_{e_1})$. Note that

$$\mathbf{D} \oplus \mathbf{B} = \mathbf{D}^+ \oplus \mathbf{G}_{e_1} \quad \text{and} \tag{10}$$

$$\mathbf{D} \oplus \mathbf{B}^* = \mathbf{D}^+ \oplus \mathbf{G}_{e_2}. \tag{11}$$

From (5), we have that $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{D}^+ \oplus \mathbf{G}_{e_1} \iff \mathcal{F} \preceq_{\mathsf{tm}} \mathbf{D}^+ \oplus \mathbf{G}_{e_2}$. This, together with (10) and (11), implies that $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{D} \oplus \mathbf{B} \iff \mathcal{F} \preceq_{\mathsf{tm}} \mathbf{D} \oplus \mathbf{B}^*$, therefore $\mathbf{B} \equiv^{(\mathcal{F},t)} \mathbf{B}^*$, and the claim follows.

We just proved that $\mathbf{B} \equiv^{(\mathcal{F},t)} \mathbf{B}^*$. This, together with (9), contradict the fact that $\mathbf{B} \in \mathcal{R}^{(\mathcal{F},t)}$. Therefore $|X| \leq s$, as required. ◀

Given a graph G and an integer y, we say that a vertex set $S \subseteq V(G)$ is a *branchwidth-y-modulator* if $\mathbf{bw}(G \setminus S) \leq y$. This notion is inspired from *treewidth-modulators*, which have been recently used in a series of papers (cf., for instance, [5, 14, 15, 19].

The following proposition is a (weaker) restatement of [14, Lemma 3.10 of the full version] (see also [19]).

▶ **Proposition 30.** *There exists a function $f_2 : \mathbb{N}_{\geq 1} \times \mathbb{N} \to \mathbb{N}$ such that if $d \in \mathbb{N}_{\geq 1}$, $y \in \mathbb{N}$, and G is a graph such that $G \in \mathsf{ex}_{\mathsf{tm}}(K_d)$ and G contains a* branchwidth-y-modulator *R, then there exists a partition $\mathcal{X}$ of $V(G)$ and an element $X_0 \in \mathcal{X}$ such that $R \subseteq X_0$,* $\max\{|X_0|, |\mathcal{X}| - 1\} \leq 2 \cdot |R|$, *and for every $X \in \mathcal{X} \setminus \{X_0\}$, $|N_G(X)| \leq f_2(d, y)$.*

▶ **Lemma 31.** *There is a function $h_3 : \mathbb{N} \to \mathbb{N}$ such that if $t \in \mathbb{N}$ and $\mathcal{F}$ is a connected and planar collection, where $d = \mathsf{size}(\mathcal{F})$, then every graph in $\mathcal{R}^{(t,\mathcal{F})}$ has at most $t \cdot h_3(d)$ vertices.*

Proof. We define $h_3 : \mathbb{N} \to \mathbb{N}$ so that $h_3(d) = 2 + h_2(f_2(d, \mu(d)), \mu(d))$ where h_2 is the function of Lemma 29, f_2 is the function of Proposition 30, and μ is the function of Lemma 9.

As $\mathbf{F}^{(\mathcal{F},t)}$ has at most d vertices, we may assume that $\mathbf{G} = (G, R, \lambda) \in \mathcal{R}^{(t,\mathcal{F})} \setminus \{\mathbf{F}^{(\mathcal{F},t)}\}$. Note that $G \in \mathsf{ex}_{\mathsf{tm}}(\mathcal{F})$, therefore, from Lemma 9, $\mathbf{bw}(G) \le \mu(d)$. We set $y = \mu(d)$ and we observe that R is a *branchwidth-y-modulator* of G. Therefore, we can apply Proposition 30 on G and R and obtain a partition $\mathcal{X}$ of $V(G)$ and an element $X_0 \in \mathcal{X}$ such that

$$R \subseteq X_0, \tag{12}$$

$$\max\{|X_0|, a\} \le 2 \cdot |R|, \text{ and} \tag{13}$$

$$\forall X \in \mathcal{X} \setminus \{X_0\} : \ |N_G(X)| \le f_2(d, y) \tag{14}$$

From (12) and (14), each $X \in \mathcal{X} \setminus \{X_0\}$ is a subset of $V(G)$ such that $X \cap R = \emptyset$ and $|N_G(X)| \le f_2(d, y)$. Therefore, from Lemma 29, for each $X \in \mathcal{X} \setminus \{X_0\}$, $|X| \le h_2(f_2(d, y), d)$. We obtain that

$$\begin{aligned} |G| &= |X_0| + \sum_{X \in \mathcal{X} \setminus \{X_0\}} |X| \\ &\le^{(13)} 2 \cdot |R| + |R| \cdot h_2(f_2(d, y), d) \\ &= t \cdot (2 + h_2(f_2(d, y), d)) \\ &= t \cdot h_3(d), \end{aligned}$$

as required. ◀

We are finally ready to prove Lemma 11.

Proof of Lemma 11. Before we proceed to the proof we need one more definition. Given $n \in \mathbb{N}$, we set $\mathcal{B}^{(\mathcal{F},t)}_{\le n} = \mathcal{A}^{(t)}_{\mathcal{F},n-t} \cup \{\mathbf{F}^{(\mathcal{F},t)}\}$.

Note that, from Lemma 31, $\mathcal{R}^{(\mathcal{F},t)} \subseteq \mathcal{B}^{(\mathcal{F},t)}_{\le n}$, where $n = t \cdot h_3(d)$. Also, from Lemma 9, all graphs in $\mathcal{B}^{(\mathcal{F},t)}_{\le n}$ have branchwidth at most $y = \max\{\mu(d), t\}$. The fact that $|\mathcal{B}^{(\mathcal{F},t)}_{\le n}| = 2^{\mathcal{O}_d(t \cdot \log t)}$ follows easily by applying Proposition 10 for n and y.

The algorithm claimed in the second statement of the lemma constructs a set of representatives $\mathcal{R}^{(\mathcal{F},t)}$ as follows: first it finds a partition $\mathcal{Q}$ of $\mathcal{B}^{(\mathcal{F},t)}_{\le n}$ into equivalence classes with respect to $\equiv^{(\mathcal{F},t)}$ and then picks an element with minimum number of edges from each set of this partition.

The computation of the above partition of $\mathcal{B}^{(\mathcal{F},t)}_{\le n}$ is based on the fact that, given two t-boundaried graphs $\mathbf{B}_1$ and $\mathbf{B}_2$, $\mathbf{B}_1 \equiv^{(\mathcal{F},t)} \mathbf{B}_2$ iff for every $\mathbf{G} \in \mathcal{B}^{(\mathcal{F},t)}_{\le n}$ $\mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_1 \iff \mathcal{F} \preceq_{\mathsf{tm}} \mathbf{G} \oplus \mathbf{B}_2$. This fact follows directly from Lemma 22 and taking into account that $\mathcal{R}^{(\mathcal{F},t)} \subseteq \mathcal{B}^{(\mathcal{F},t)}_{\le n}$.

Note that it takes $|\mathcal{B}^{(\mathcal{F},t)}_{\le n}|^3 \cdot \mathcal{O}_d(1) \cdot t^{\mathcal{O}(1)}$ steps to construct $\mathcal{Q}$. As $|\mathcal{B}^{(\mathcal{F},t)}_{\le n}| = 2^{\mathcal{O}_d(t \cdot \log t)}$, the construction of $\mathcal{Q}$, and therefore of $\mathcal{R}^{(\mathcal{F},t)}$ as well, can be done in the claimed number of steps. ◀

F Deferred contents of Section 4

Before proving Lemma 12, we first need an auxiliary lemma.

▶ **Lemma 32.** *Let n_0 be a positive integer. Assume that for each graph G' such that $1 \le n(G') \le n_0$, $C_4 \not\preceq_{\mathsf{tm}} G'$ if and only if G satisfies the C_4-condition. If G is a graph that does not contain a diamond as a subgraph and such that $n(G) = n_0$, then $n(G) - m(G) + \mathsf{c}_3(G) \le \mathsf{cc}(G)$.*

Proof. Let n_0 be a positive integer, and assume that for each graph G' such that $1 \leq n(G') \leq n_0$, $C_4 \not\preceq_{\mathsf{tm}} G'$ if and only if G satisfies the C_4-condition. Let G be a graph that does not contain a diamond as a subgraph and such that $n(G) = n_0$. Let $S \subseteq E(G)$ such that $C_4 \not\preceq_{\mathsf{tm}} G \setminus S$ and $\mathsf{cc}(G \setminus S) = \mathsf{cc}(G)$ (note that any minimal feedback edge set satisfies these conditions). We have, by hypothesis, that $G \setminus S$ satisfies the C_4-condition, so $n(G \setminus S) - m(G \setminus S) + \mathsf{c}_3(G \setminus S) = \mathsf{cc}(G \setminus S)$. Moreover, as G does not contain a diamond as a subgraph, each edge of G participates in at most one C_3, and thus $\mathsf{c}_3(G) - \mathsf{c}_3(G \setminus S) \leq |S|$. As by definition $n(G) = n(G \setminus S)$ and $m(G) - m(G \setminus S) = |S|$, we obtain that $n(G) - m(G) + \mathsf{c}_3(G) \leq \mathsf{cc}(G \setminus S) = \mathsf{cc}(G)$. ◀

Proof of Lemma 12. Let G be a non-empty graph, and assume first that $C_4 \not\preceq_{\mathsf{tm}} G$. This directly implies that G does not contain the diamond as a subgraph. In particular, any two cycles of G, which are necessarily C_3's, cannot share an edge. Let S be a set containing an arbitrary edge of each C_3 in G. By construction, $G \setminus S$ is a forest. As in a forest F, we have $n(F) - m(F) = \mathsf{cc}(F)$, and S is defined such that $|S| = \mathsf{c}_3(G)$ because each edge of G participates in at most one C_3, we obtain that $n(G) - m(G) + \mathsf{c}_3(G) = \mathsf{cc}(G)$. Thus, G satisfies the C_4-condition.

Conversely, assume now that G satisfies the C_4-condition. We prove that $C_4 \not\preceq_{\mathsf{tm}} G$ by induction on $n(G)$. If $n(G) \leq 3$, then $n(G) < n(C_4)$ and so $C_4 \not\preceq_{\mathsf{tm}} G$. Assume now that $n(G) \geq 4$, and that for each graph G' such that $1 \leq n(G') < n(G)$, if G' satisfies the C_4-condition, then $C_4 \not\preceq_{\mathsf{tm}} G'$. We prove that this last implication is also true for G. Note that, as two C_3 cannot share an edge in G, we have that $\mathsf{c}_3(G) \leq \frac{m(G)}{3}$. This implies that the minimum degree of G is at most 3. Indeed, if each vertex of G had degree at least 4, then $m(G) \geq 2n(G)$, which together with the relations $n(G) - m(G) + \mathsf{c}_3(G) = \mathsf{cc}(G)$ and $\mathsf{c}_3(G) \leq \frac{m(G)}{3}$ would imply that $\mathsf{cc}(G) < 0$, a contradiction. Let $v \in V(G)$ be a vertex with minimum degree. We distinguish two cases according to the degree of v.

If v has degree 0 or 1, then the graph $G \setminus \{v\}$ satisfies the C_4-condition as well, implying that $C_4 \not\preceq_{\mathsf{tm}} G \setminus \{v\}$. As v has degree at most 1, it cannot be inside a cycle, hence $C_4 \not\preceq_{\mathsf{tm}} G$.

Assume that v has degree 2 and participates in a C_3. As G does not contain a diamond as a subgraph, $C_4 \preceq_{\mathsf{tm}} G$ if and only if $C_4 \preceq_{\mathsf{tm}} G \setminus \{v\}$. Moreover $n(G \setminus \{v\}) = n(G) - 1$, $m(G \setminus \{v\}) = m(G) - 2$, $\mathsf{c}_3(G \setminus \{v\}) = \mathsf{c}_3(G) - 1$, and $\mathsf{cc}(G \setminus \{v\}) = \mathsf{cc}(G)$. This implies that $G \setminus \{v\}$ satisfies the C_4-condition, hence $C_4 \not\preceq_{\mathsf{tm}} G \setminus \{v\}$, and therefore $C_4 \not\preceq_{\mathsf{tm}} G$.

Finally, assume that v has degree 2 and does not belong to any C_3. Using the induction hypothesis and Lemma 32, we have that $n(G \setminus \{v\}) - m(G \setminus \{v\}) + \mathsf{c}_3(G \setminus \{v\}) \leq \mathsf{cc}(G \setminus \{v\})$. As $n(G \setminus \{v\}) = n(G) - 1$, $m(G \setminus \{v\}) = m(G) - 2$, $\mathsf{c}_3(G \setminus \{v\}) = \mathsf{c}_3(G)$, v has degree 2 in G, and G satisfies the C_4-condition, we obtain that $\mathsf{cc}(G \setminus \{v\}) = \mathsf{cc}(G) - 1$. This implies that $G \setminus \{v\}$ satisfies the C_4-condition, and thus $C_4 \not\preceq_{\mathsf{tm}} G \setminus \{v\}$. Since v disconnects one of the connected components of G it cannot participate in a cycle of G, hence $C_4 \not\preceq_{\mathsf{tm}} G$. ◀

Proof of Lemma 13. As $C_4 \not\preceq_{\mathsf{tm}} G$, by Lemma 12 G satisfies the C_4-condition. It follows that $\mathsf{c}_3(G) \leq \frac{1}{3}m(G)$. Moreover, as G is non-empty, we have that $1 \leq \mathsf{cc}(G)$. The lemma follows by using these inequalities in the equality $n(G) - m(G) + \mathsf{c}_3(G) = \mathsf{cc}(G)$. ◀

Computation of $\mathcal{A}_t$ in the dynamic programming algorithm. We distinguish several cases depending on the type of node t in the nice tree decomposition.

- **Leaf.** By definition of $\mathcal{A}_t$ we have $\mathcal{A}_t(\varnothing, \varnothing, \varnothing, 0, 0, 0) = \{\emptyset\}$.
- **Introduce vertex.** Let v be the insertion vertex of X_t, let t' be the child of t, let $\mathbf{s}$, $\mathbf{s}_0$, and $\mathbf{r}$ the functions defined as before, let $H = G_t\langle \mathbf{s}^{-1}(1), \mathbf{s}_0^{-1}(1)\rangle$, and let d_3 be the number of C_3's of H that contain the vertex v.

- If $C_4 \preceq_{\mathsf{tm}} H$ or if $v = v_0$ and $\mathbf{s}(v_0) = 0$, then by definition of $\mathcal{A}_t$ we have that $\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, \mathbf{r}, i, j, \ell) = \emptyset$.
- Otherwise, if $\mathbf{s}(v) = 0$, then,
  by the definition of $\mathcal{A}_t$, $\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, \mathbf{r}, i, j, \ell) = \mathcal{A}_{t'}(\mathbf{s}|_{X_{t'}}, \mathbf{s}_0|_{E_{t'}}, \mathbf{r}|_{E_{t'}}, i, j, \ell)$.
- Otherwise, if $v = v_0$, then by construction of the nice tree decomposition, we know that t' is a leaf of T and so $\mathbf{s} = \{(v_0, 1)\}$, $\mathbf{s}_0 = \mathbf{r} = \emptyset$, $j = \ell = i - 1 = 0$ and $\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, \mathbf{r}, i, j, \ell) = \mathsf{ins}(\{v_0\}, \mathcal{A}_{t'}(\varnothing, \varnothing, \varnothing, 0, 0, 0))$.
- Otherwise, we know that $v \neq v_0$, $\mathbf{s}(v) = 1$, and $C_4 \not\preceq_{\mathsf{tm}} H$. As $\mathbf{s}(v) = 1$, we have to insert v and we have to make sure that all vertices of $N_H[v]$ are in the same connected component of H. Moreover, by adding v we add one vertex, $|N(v)|$ edges, and d_3 C_3's. Therefore, we have that

$$\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, \mathbf{r}, i, j, \ell) = \mathsf{glue}(N_H[v], \mathsf{ins}(\{v\}, \mathcal{A}_{t'}(\mathbf{s}|_{X_{t'}}, \mathbf{s}_0|_{E_{t'}}, \mathbf{r}|_{E_{t'}}, i - 1, j - |N_H(v)|, \ell - d_3))).$$

Forget vertex. Let v be the forget vertex of X_t, let t' be the child of t, and let $\mathbf{s}$, $\mathbf{s}_0$, and $\mathbf{r}$ the functions defined as before. For each function, we have a choice on how it can be extended in t', and we potentially need to consider every possible such extension. Note the number of vertices, edges, or C_3's is not affected. We obtain that

$$\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, \mathbf{r}, i, j, \ell) = \mathcal{A}_{t'}(\mathbf{s} \cup \{(v, 0)\}, \mathbf{s}_0, \mathbf{r}, i, j, \ell) \biguplus_{\substack{\mathbf{s}': X_{t'} \to \{0,1\},\ \mathbf{s}'|_{X_t} = \mathbf{s},\ \mathbf{s}'(v) = 1 \\ \mathbf{s}'_0: \{v_0\} \times \mathbf{s}'^{-1}(1) \to \{0,1\},\ \mathbf{s}'_0|_{X_t} = \mathbf{s}_0 \\ \mathbf{r}': E(G_t\langle \mathbf{s}'^{-1}(1), \mathbf{s}_0'^{-1}(1)\rangle) \to \{0,1\},\ \mathbf{r}'|_{X_t} = \mathbf{r}}} \mathsf{proj}(\{v\}, A_{t'}(\mathbf{s}', \mathbf{s}'_0, \mathbf{r}', i, j, \ell)).$$

Join. Let t' and t'' be the two children of t, let $\mathbf{s}$, $\mathbf{s}_0$, and $\mathbf{r}$ be the functions defined as before, let $H = G_t\langle \mathbf{s}^{-1}(1), \mathbf{s}_0^{-1}(1)\rangle$, and let $S \subseteq E(H)$ be the set of edges that participate in a C_3 of H.
We join every pair of compatible entries $A_{t'}(\mathbf{s}', \mathbf{s}'_0, \mathbf{r}', i', j', \ell')$ and $A_{t''}(\mathbf{s}'', \mathbf{s}''_0, \mathbf{r}'', i'', j'', \ell'')$. For two such entries being compatible, we need $\mathbf{s}' = \mathbf{s}'' = \mathbf{s}$ and $\mathbf{s}'_0 = \mathbf{s}''_0 = \mathbf{s}_0$. Moreover, we do not want the solution graph to contain a diamond as a subgraph, and for this we need $\mathbf{r}'^{-1}(1) \cap \mathbf{r}''^{-1}(1) = S$. Indeed, either H contains the diamond as a subgraph, and then $A_{t'}(\mathbf{s}', \mathbf{s}'_0, \mathbf{r}', i', j', \ell') = A_{t''}(\mathbf{s}'', \mathbf{s}''_0, \mathbf{r}'', i'', j'', \ell'') = \{\emptyset\}$, or the diamond is created by joining two C_3's, one from t' and the other one from t'', sharing a common edge. This is possible only if $(\mathbf{r}'^{-1}(1) \cap \mathbf{r}''^{-1}(1)) \setminus S \neq \emptyset$. For the counters, we have to be careful in order not to count some element twice. We obtain that

$$\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, \mathbf{r}, i, j, \ell) = \biguplus_{\substack{\mathbf{r}', \mathbf{r}'': E(H) \to \{0,1\}, \\ \mathbf{r}'^{-1}(1) \cap \mathbf{r}''^{-1}(1) = S \\ i' + i'' = i + |V(H)| \\ j' + j'' = j + |E(H)| \\ \ell' + \ell'' = \ell + \mathsf{c}_3(H)}} \mathsf{join}(A_{t'}(\mathbf{s}, \mathbf{s}_0, \mathbf{r}', i', j', \ell'), A_{t''}(\mathbf{s}, \mathbf{s}_0, \mathbf{r}'', i'', j'', \ell'')).$$

Proof of Theorem 17. The algorithm works in the following way. For each node $t \in V(T)$ and for each entry M of its table, instead of storing $\mathcal{A}_t(M)$, we store $\mathcal{A}'_t(M) = \mathsf{reduce}(\mathcal{A}_t(M))$ by using Theorem 16. As each of the operation we use preserves representation by Proposition 15, we obtain that for each node $t \in V(T)$ and for each possible entry M, $\mathcal{A}'_t(M)$ represents $\mathcal{A}_t(M)$. In particular, we have that $\mathcal{A}'_r(M) = \mathsf{reduce}(\mathcal{A}_r(M))$ for each possible entry M. Using the definition of $\mathcal{A}_r$, Lemma 12, and Lemma 13, we have that

$\mathbf{tm}_{\{C_4\}}(G) \leq k$ if and only if for some $i \geq |V(G) \cup \{v_0\}| - k$ and some $j \leq \frac{2}{3}(i-1)$, we have $\mathcal{A}'_r(\varnothing, \varnothing, \varnothing, i, j, 1+j-i) \neq \emptyset$.

We now focus on the running time of the algorithm. The size of the intermediate sets of weighted partitions, for a leaf node and for an introduce vertex node are upper-bounded by $2^{|\mathbf{s}^{-1}(1)|}$. For a forget vertex node, as in the big union operation we take into consideration a unique extension of $\mathbf{s}$, at most 2 possible extensions of $\mathbf{s}_0$, and at most $2^{|\mathbf{s}^{-1}(1)|}$ possible extensions for $\mathbf{r}$, we obtain that the intermediate sets of weighted partitions have size at most $2^{|\mathbf{s}^{-1}(1)|} + 2 \cdot 2^{|\mathbf{s}^{-1}(1)|} \cdot 2^{|\mathbf{s}^{-1}(1)|} \leq 2^{2|\mathbf{s}^{-1}(1)|+2}$. For a join node, as in the big union operation we take into consideration at most $2^{|E(H)|}$ possible functions $\mathbf{r}'$ and as many functions $\mathbf{r}''$, at most $n + |\mathbf{s}^{-1}|$ choices for i' and i'', at most $\frac{3}{2}(n-1) + |E(H)|$ choices for j' and j'', and at most $\frac{1}{2}(n-1) + \frac{1}{3}|E(H)|$ choices for ℓ' and ℓ'', we obtain that the intermediate sets of weighted partitions have size at most $2^{|E(H)|} \cdot 2^{|E(H)|} \cdot (n + |\mathbf{s}^{-1}|) \cdot (\frac{3}{2}(n-1) + |E(H)|) \cdot (\frac{1}{2}(n-1) + \frac{1}{3}|E(H)|) \cdot 4^{|\mathbf{s}^{-1}(1)|}$. As each time we can check the condition $C_4 \not\preceq_{\mathsf{tm}} H$, by Lemma 13 $m(H) \leq \frac{3}{2}(n(H)-1)$, so we obtain that the intermediate sets of weighted partitions have size at most $6 \cdot n^3 \cdot 2^{5|\mathbf{s}^{-1}(1)|}$. Moreover, for each node $t \in V(T)$, the function reduce will be called as many times as the number of possible entries, i.e., at most $2^{\mathcal{O}(w)} \cdot n^3$ times. Thus, using Theorem 16, $\mathcal{A}'_t$ can be computed in time $2^{\mathcal{O}(w)} \cdot n^6$. The theorem follows by taking into account the linear number of nodes in a nice tree decomposition. ◀

G Single-exponential lower bound for any connected $\mathcal{F}$

▶ **Theorem 33.** *Let $\mathcal{F}$ be a connected collection. Neither $\mathcal{F}$-TM-Deletion nor $\mathcal{F}$-M-Deletion can be solved in time $2^{o(\mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ unless the ETH fails.*

Proof. Let $\mathcal{F}$ be a connected collection and recall that $w = \mathbf{tw}(G)$. Without loss of generality, we can assume that $\mathcal{F}$ is a topological minor antichain. We present a reduction from Vertex Cover to $\mathcal{F}$-TM-Deletion, both parameterized by the treewidth of the input graph, and then we explain the changes to be made to prove the lower bound for $\mathcal{F}$-M-Deletion. It is known that Vertex Cover cannot be solved in time $2^{o(w)} \cdot n^{\mathcal{O}(1)}$ unless the ETH fails [17] (in fact, it cannot be solved even in time $2^{o(n)}$).

First we select an essential pair (H, B) of $\mathcal{F}$. Let a be the first vertex of (H, B), b be the second vertex of (H, B), and A be the core of (H, B).

Let G be the input graph of the Vertex Cover problem and let $<$ be an arbitrary total order on $V(G)$. We build a graph G' starting from G. For each vertex v of G, we add a copy of A, which we call A^v, and we identify the vertices v and a. For each edge $e = \{v, v'\} \in E(G)$ with $v < v'$, we remove e, we add a copy of B, which we call B^e, and we identify the vertices v and a and the vertices v' and b. This concludes the construction of G'. Note that $|V(G')| = |V(G)| \cdot |V(A)| + |E(G)| \cdot |V(B) \setminus \{a, b\}|$ and that $\mathbf{tw}(G') = \max\{\mathbf{tw}(G), \mathbf{tw}(H)\}$.

We claim that there exists a solution of size at most k of Vertex Cover in G if and only if there is a solution of size at most k of $\mathcal{F}$-TM-Deletion in G'.

In one direction, assume that S is a solution of $\mathcal{F}$-TM-Deletion in G' with $|S| \leq k$. By definition of the problem, for each $e = \{v, v'\} \in E(G)$ with $v < v'$, either B^e contains an element of S or A^v contains an element of S. Let $S' = \{v \in V(G) \mid \exists v' \in V(G) : v < v', e = \{v, v'\} \in E(G), (V(B^e) \setminus \{v, v'\}) \cap S \neq \emptyset\} \cup \{v \in V(G) \mid V(A^v) \cap S \neq \emptyset\}$. Then S' is a solution of Vertex Cover in G and $|S'| \leq |S| \leq k$.

In the other direction, assume that we have a solution S of size at most k of Vertex Cover in G. We want to prove that S is also a solution of $\mathcal{F}$-TM-Deletion in G'. For this, we fix an arbitrary $H' \in \mathcal{F}$ and we show that H' is not a topological minor of $G' \setminus S$.

First note that the connected components of $G' \setminus S$ are either of the shape $A^v \setminus \{v\}$ if $v \in S$, $B^e \setminus e$ if $e \subseteq S$, or the union of A^v with zero, one, or more graphs $B^{\{v,v'\}} \setminus \{v'\}$ such that $\{v, v'\} \in E(G)$ if $v \in V(G) \setminus S$. As $\mathcal{F}$ is a topological minor antichain, for any $v \in V(G)$, $H' \not\preceq_{\mathsf{tm}} A^v \setminus \{v\}$ and for any $e \in E(G)$, $H' \not\preceq_{\mathsf{tm}} B^e \setminus e$. Moreover, let $v \in V(G) \setminus S$ and let K be the connected component of $G \setminus S$ containing v. K is the union of A^v and of every $B^{\{v,v'\}} \setminus \{v'\}$ such that $\{v, v'\} \in E(G)$. As, for each $v' \in V(G)$ such that $\{v, v'\} \in E(G)$, v' is not an isolated vertex in $B^{\{v,v'\}}$, by definition of B, for any $B' \in L(\mathsf{bct}(H'))$, $|E(B^{\{v,v'\}} \setminus \{v'\})| < |E(B')|$. This implies that for each leaf B' of $\mathsf{bct}(H')$ and for each $\{v, v'\} \in E(G)$, $B' \not\preceq_{\mathsf{tm}} B^{\{v,v'\}} \setminus \{v'\}$. Moreover, it follows by definition of $\mathcal{F}$ that $H' \not\preceq_{\mathsf{tm}} A^v$. This implies by Lemma 2 that H' is not a topological minor of K. Moreover, as H' is connected by hypothesis, it follows that that H' is not a topological minor of $G' \setminus S$ either. This concludes the proof for the topological minor version.

Finally, note that the same proof applies to $\mathcal{F}$-M-Deletion as well, just by replacing

- $\mathcal{F}$-TM-Deletion with $\mathcal{F}$-M-Deletion,
- topological minor with minor,
- $\preceq_{\mathsf{tm}}$ with $\preceq_{\mathsf{m}}$, and
- Lemma 2 with Lemma 34 below. ◀

The following lemma can be proved analogously to Lemma 2, using the same kind of argumentation with minors instead of topological minors.

▶ **Lemma 34.** *Let G be a connected graph, let v be a cut vertex of G, and let V be the vertex set of a connected component of $G \setminus \{v\}$. If H is a graph such that $H \preceq_{\mathsf{m}} G$ and for each leaf B of $\mathsf{bct}(H)$, $B \not\preceq_{\mathsf{m}} G[V \cup \{v\}]$, then $H \preceq_{\mathsf{m}} G \setminus V$.*

H Single-exponential lower bound for hitting P_3's and P_4's

We first need a simple observation and to introduce nice tree decompositions, which will be very helpful in the algorithms.

▶ **Observation 4.** *Let G be a graph and h be a positive integer. Then the following assertions are equivalent.*

- *G contains P_h as a topological minor.*
- *G contains P_h as a minor.*
- *G contains P_h as a subgraph.*

Moreover, the following assertions are also equivalent.

- *G contains C_h as a topological minor.*
- *G contains C_h as a minor.*

Nice tree decompositions. Let $\mathcal{D} = (T, \mathcal{X})$ be a tree decomposition of G, r be a vertex of T, and $\mathcal{G} = \{G_t \mid t \in V(T)\}$ be a collection of subgraphs of G, indexed by the vertices of T. We say that the triple $(\mathcal{D}, r, \mathcal{G})$ is a *nice tree decomposition* of G if the following conditions hold:

- $X_r = \emptyset$ and $G_r = G$,
- each node of $\mathcal{D}$ has at most two children in T,
- for each leaf $t \in V(T)$, $X_t = \emptyset$ and $G_t = (\emptyset, \emptyset)$. Such t is called a *leaf node*,
- if $t \in V(T)$ has exactly one child t', then either
  - $X_t = X_{t'} \cup \{v_{\text{insert}}\}$ for some $v_{\text{insert}} \notin X_{t'}$ and $G_t = G[V(G_{t'}) \cup \{v_{\text{insert}}\}]$. The node t is called *introduce vertex* node and the vertex v_{insert} is the *insertion vertex* of X_t,

- $X_t = X_{t'} \setminus \{v_{\text{forget}}\}$ for some $v_{\text{forget}} \in X_{t'}$ and $G_t = G_{t'}$. The node t is called *forget vertex* node and v_{forget} is the *forget vertex* of X_t.
- if $t \in V(T)$ has exactly two children t' and t'', then $X_t = X_{t'} = X_{t''}$, and $E(G_{t'}) \cap E(G_{t''}) = \emptyset$. The node t is called a *join* node.

As discussed in [20], given a tree decomposition, it is possible to transform it in polynomial time to a *nice* new one of the same width. Moreover, by Bodlaender *et al.* [4] we can find in time $2^{\mathcal{O}(\mathbf{tw})} \cdot n$ a tree decomposition of width $\mathcal{O}(\mathbf{tw})$ of any graph G. Hence, since in this section we focus on single-exponential algorithms, we may assume that a nice tree decomposition of width $w = \mathcal{O}(\mathbf{tw})$ is given with the input.

H.1 A single-exponential algorithm for $\{P_3\}$-TM-Deletion

It should be noted that a single-exponential algorithm for $\{P_3\}$-TM-Deletion is already known. Indeed, Tu *et al.* [29] presented an algorithm running in time $\mathcal{O}^*(4^{\mathbf{tw}})$, and very recently Baia *et al.* [1] improved it to $\mathcal{O}^*(3^{\mathbf{tw}})$. Nevertheless, for completeness we present in this section a simpler algorithm, but involving a greater constant than [1, 29].

We first give a simple structural characterization of the graphs that exclude P_3 as a topological minor.

▶ **Lemma 35.** *Let G be a graph. $P_3 \not\preceq_{\mathsf{tm}} G$ if and only if each vertex of G has degree at most 1.*

Proof. Let G be a graph. If G has a connected component of size at least 3, then clearly it contains a P_3. This implies that, if $P_3 \not\preceq_{\mathsf{tm}} G$, then each connected component of G has size at most 2 and so, each vertex of G has degree at most 1. Conversely, if each vertex of G has degree at most 1, then, as P_3 contains a vertex of degree 2, $P_3 \not\preceq_{\mathsf{tm}} G$. ◀

We present an algorithm using classical dynamic programming techniques over a tree decomposition of the input graph. Let G be an instance of $\{P_3\}$-TM-Deletion and let $((T, \mathcal{X}), r, \mathcal{G})$ be a nice tree decomposition of G.

We define, for each $t \in V(T)$, the set $\mathcal{I}_t = \{(S, S_0) \mid S, S_0 \subseteq X_t,\ S \cap S_0 = \emptyset\}$ and a function $\mathbf{r}_t : \mathcal{I}_t \to \mathbb{N}$ such that for each $(S, S_0) \in \mathcal{I}_t$, $\mathbf{r}(S, S_0)$ is the minimum ℓ such that there exists a set $\widehat{S} \subseteq V(G_t)$, called the *witness* of (S, S_0), that satisfies:

- $|\widehat{S}| \leq \ell$,
- $\widehat{S} \cap X_t = S$,
- $P_3 \not\preceq_{\mathsf{tm}} G_t \setminus \widehat{S}$, and
- S_0 is the set of each vertex of X_t of degree 0 in $G_t \setminus S$.

Note that with this definition, $\mathbf{tm}_{\mathcal{F}}(G) = \mathbf{r}_r(\emptyset, \emptyset)$. For each $t \in V(T)$, we assume that we have already computed $\mathbf{r}_{t'}$ for each children t' of t, and we proceed to the computation of $\mathbf{r}_t$. We distinguish several cases depending on the type of node t.

Leaf. $\mathcal{I}_t = \{(\emptyset, \emptyset)\}$ and $\mathbf{r}_t(\emptyset, \emptyset) = 0$.

Introduce vertex. If v is the insertion vertex of X_t and t' is the child of t, then for each $(S, S_0) \in \mathcal{I}_t$,

$$\begin{aligned}\mathbf{r}_t(S, S_0) \quad = \quad \min\big(& \{\mathbf{r}_{t'}(S', S_0) + 1 \mid (S', S_0) \in \mathcal{I}_{t'},\ S = S' \cup \{v\}\} \\ & \cup \{\mathbf{r}_{t'}(S, S'_0) \mid (S, S'_0) \in \mathcal{I}_{t'}, S_0 = S'_0 \cup \{v\},\ N_{G_t[X_t]}(v) \setminus S = \emptyset\} \\ & \cup \{\mathbf{r}_{t'}(S, S'_0) \mid (S, S'_0) \in \mathcal{I}_{t'}, S_0 = S'_0 \setminus \{u\},\ u \in S'_0,\ N_{G_t[X_t]}(v) \setminus S = \{u\}\}\ \big).\end{aligned}$$

Forget vertex. If v is the forget vertex of X_t and t' is the child of t, then for each $(S, S_0) \in \mathcal{I}_t$,

$$\mathbf{r}_t(S, S_0) \quad = \quad \min\{\mathbf{r}_{t'}(S', S'_0) \mid (S', S'_0) \in \mathcal{I}_{t'},\ S = S' \setminus \{v\},\ S_0 = S'_0 \setminus \{v\}\}$$

Join. If t' and t'' are the children of t, then for each $(S, S_0) \in \mathcal{I}_t$,

$$\begin{aligned}\mathbf{r}(S, S_0) \quad = \quad & \min\{\mathbf{r}(S', S'_0) + \mathbf{r}(S'', S''_0) - |S' \cap S''| \\ & \quad | \ (S', S'_0) \in \mathcal{I}_{t'}, (S'', S''_0) \in \mathcal{I}_{t''}, \\ & \quad S = S' \cup S'', \ S_0 = S'_0 \cap S''_0, \ X_t \setminus S \subseteq S'_0 \cup S''_0\}.\end{aligned}$$

Let us analize the running time of this algorithm. As, for each $t \in V(T)$, S and S_0 are disjoint subsets of X_t, we have that $|\mathcal{I}_t| \leq 3^{|X_t|}$. Note that if t is a leaf, then $\mathbf{r}_t$ can be computed in time $\mathcal{O}(1)$, if t is an introduce vertex or a forget vertex node, and t' is the child of t, then $\mathbf{r}_t$ can be computed in time $\mathcal{O}(|\mathcal{I}_{t'}| \cdot |X_t|)$, and if t is a join node, and t' and t'' are the two children of t, then $\mathbf{r}_t$ can be computed in time $\mathcal{O}(|\mathcal{I}_{t'}| \cdot |\mathcal{I}_{t''}| \cdot |X_t|)$.

We now show that for each $t \in V(T)$, the function $\mathbf{r}_t$ is correctly computed by the algorithm.

Leaf. This follows directly from the definition of $\mathbf{r}_t$.

Introduce vertex. Let v be the insertion vertex of X_t. As v is the insertion vertex, we have that $N_{G_t[X_t]}(v) = N_{G_t}(v)$, and so for each value we add to the set, we can find a witness of (S, S_0) of size bounded by this value.

Conversely, let $(S, S_0) \in \mathcal{I}_t$ and let $\widehat{S}$ be a witness. If $v \in S$, then $(S \setminus \{v\}, S_0) \in \mathcal{I}_{t'}$ and $\mathbf{r}(S \setminus \{v\}, S_0) \leq |\widehat{S}| - 1$, if $v \in S_0$ then $(S, S_0 \setminus \{v\}) \in \mathcal{I}_{t'}$ and $\mathbf{r}(S, S_0 \setminus \{v\}) \leq |\widehat{S}|$, and if $v \in X_t \setminus (S \cup S_0)$, then by definition v has a unique neighbor, say u, in $G_t \setminus \widehat{S}$, moreover $u \in X_t \setminus (S \cup S_0)$, v is the unique neighbor of u in $G_t \setminus \widehat{S}$, $(S, S_0 \cup \{u\}) \in \mathcal{I}_{t'}$, and $\mathbf{r}(S, S_0 \cup \{u\}) \leq |\widehat{S}|$.

Forget vertex. This also follows directly from the definition of $\mathbf{r}_t$.

Join. Let $(S', S'_0) \in \mathcal{R}_{t'}$ and let $(S'', S''_0) \in \mathcal{I}_{t''}$ with witnesses $\widehat{S}'$ and $\widehat{S}''$, respectively. If $S = S' \cup S''$ and $S'_0 \cup S''_0 = X_t \setminus S$, then the condition $X_t \setminus S \subseteq S'_0 \cup S''_0$ ensures that $G_t \setminus (\widehat{S}' \cup \widehat{S}'')$ has no vertex of degree at least 2 and so $\widehat{S}' \cup \widehat{S}''$ is a witness of $(S, S'_0 \cap S''_0) \in \mathcal{I}_t$ of size at most $\mathbf{r}_{t'}(S', S'_0) + \mathbf{r}_{t'}(S'', S''_0) - |S' \cap S''|$.

Conversely, let $(S, S_0) \in \mathcal{I}_t$ with witness $\widehat{S}$. If $\widehat{S}' = \widehat{S} \cap V(G_{t'})$ and $\widehat{S}'' = \widehat{S} \cap V(G_{t''})$, then by definition of $\widehat{S}$, $\widehat{S}'$ is a witness of some $(S', S'_0) \in \mathcal{I}_{t'}$, and $\widehat{S}''$ is a witness of some $(S'', S''_0) \in \mathcal{I}_{t''}$ such that $S = S' = S''$, $S'_0 \cup S''_0 = X_t \setminus S$, and $S_0 = S'_0 \cap S''_0$, and we have $\mathbf{r}_{t'}(S', S'_0) + \mathbf{r}_{t'}(S'', S''_0) - |S| \leq |\widehat{S}|$.

The following theorem summarizes the above discussion.

▶ **Theorem 36.** *If a nice tree decomposition of G of width w is given, $\{P_3\}$-TM-Deletion can be solved in time $\mathcal{O}(9^w \cdot w \cdot n)$.*

H.2 A single-exponential algorithm for $\{P_4\}$-TM-Deletion

Similarly to what we did for $\{P_3\}$-TM-Deletion, we start with a structural definition of the graphs that exclude P_4 as a topological minor.

▶ **Lemma 37.** *Let G be a graph. $P_4 \not\preceq_{\mathsf{tm}} G$ if and only if each connected component of G is either a C_3 or a star.*

Proof. First note that if each connected component of G is either a C_3 or a star, then $P_4 \not\preceq_{\mathsf{tm}} G$. Conversely, assume that $P_4 \not\preceq_{\mathsf{tm}} G$. Then each connected component of G of size at least 4 should contain at most 1 vertex of degree at least 2, hence such component is a star. On the other hand, the only graph on at most 3 vertices that is not a star is C_3. The lemma follows. ◀

As we did for $\{P_3\}$-TM-Deletion, we present an algorithm using classical dynamic programming techniques over a tree decomposition of the input graph. Let G be an instance of $\{P_4\}$-Deletion, and let $((T, \mathcal{X}), r, \mathcal{G})$ be a nice tree decomposition of G.

We define, for each $t \in T$, the set $\mathcal{I}_t$ to be the set of each tuple $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-})$ such that $\{S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}\}$ is a partition of X_t and the function $\mathbf{r}_t : \mathcal{I}_t \to \mathbb{N}$ such that, for each $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_t$, $\mathbf{r}_t(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-})$ is the minimum ℓ such that there exists a triple $(\widehat{S}, \widehat{S}_*, \widehat{S}_{3-}) \subseteq V(G_t) \times V(G_t) \times V(G_t)$, called the *witness* of $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-})$, which satisfies the following properties:

- $\widehat{S}$, $\widehat{S}_*$, and $\widehat{S}_{3-}$ are pairwise disjoint,
- $\widehat{S} \cap X_t = S$, $\widehat{S}_* \cap X_t = S_*$, and $\widehat{S}_{3-} \cap X_t = S_{3-}$,
- $|\widehat{S}| \leq \ell$,
- $P_4 \not\preceq_{\mathsf{tm}} G_t \setminus \widehat{S}$,
- S_{1+} is a set of vertices of degree 0 in $G_t \setminus \widehat{S}$,
- each vertex of S_{1-} has a unique neighbor in $G_t \setminus \widehat{S}$ and this neighbor is in $\widehat{S}_*$,
- each connected component of $G_t[\widehat{S}_{3-}]$ is a C_3,
- there is no edge in $G_t \setminus \widehat{S}$ between a vertex of $\widehat{S}_{3-}$ and a vertex of $V(G_t) \setminus (\widehat{S} \cup \widehat{S}_{3-})$,
- there is no edge in $G_t \setminus \widehat{S}$ between a vertex of S_{3+} and a vertex of $V(G_t) \setminus (\widehat{S} \cup S_{3+})$, and
- there is no edge in $G_t \setminus \widehat{S}$ between two vertices of S_*.

Intuitively, $\widehat{S}$ corresponds to a partial solution in G_t. Note that, by Lemma 37, each component of $G_t \setminus \widehat{S}$ must be either a star or a C_3. With this in mind, $\widehat{S}_*$ is the set of vertices that are centers of a star in $G_t \setminus \widehat{S}$, S_{1+} is the set of leaves of a star that are not yet connected to a vertex of $\widehat{S}_*$, S_{1-} is the set of leaves of a star that are already connected to a vertex of $\widehat{S}_*$, $\widehat{S}_{3-}$ is the set of vertices that induce C_3's in G_t, and S_{3+} is a set of vertices that will induce C_3's when further edges will appear.

Note that with this definition, $\mathbf{tm}_{\mathcal{F}}(G) = \mathbf{r}_r(\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$. For each $t \in V(T)$, we assume that we have already computed $\mathbf{r}_{t'}$ for each children t' of t, and we proceed to the computation of $\mathbf{r}_t$. We distinguish several cases depending on the type of node t.

Leaf. $\mathcal{I}_t = \{(\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)\}$ and $\mathbf{r}_t(\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset) = 0$.

Introduce vertex. If v is the insertion vertex of X_t and t' is the child of t, then, for each $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_t$,

$$
\begin{aligned}
\mathbf{r}_t(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \quad = \quad & \min \Big(\ \{\mathbf{r}_{t'}(S', S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) + 1 \\
& \qquad \mid (S', S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{R}_{t'},\ S = S' \cup \{v\}\} \\
\cup \quad & \{\mathbf{r}_{t'}(S, S'_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \\
& \qquad \mid (S, S'_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{R}_{t'}, \\
& \qquad\ S_{1+} = S'_{1+} \cup \{v\},\ N_{G_t[X_t \setminus S]}(v) = \emptyset\} \\
\cup \quad & \{\mathbf{r}_{t'}(S, S_{1+}, S'_{1-}, S_*, S_{3+}, S_{3-}) \\
& \qquad \mid (S, S_{1+}, S'_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{R}_{t'}, \\
& \qquad\ S_{1-} = S'_{1-} \cup \{v\},\ z \in S_*,\ N_{G_t[X_t \setminus S]}(v) = \{z\}\} \\
\cup \quad & \{\mathbf{r}_{t'}(S, S'_{1+}, S'_{1-}, S'_*, S_{3+}, S_{3-}) \\
& \qquad \mid (S, S'_{1+}, S'_{1-}, S'_*, S_{3+}, S_{3-}) \in \mathcal{R}_{t'}, \\
& \qquad\ S_* = S'_* \cup \{v\},\ N_{G_t[X_t \setminus S]}(v) \subseteq S'_{1+}, \\
& \qquad\ S_{1+} = S'_{1+} \setminus N_{G_t[X_t \setminus S]}(v),\ S_{1-} = S'_{1-} \cup N_{G_t[X_t \setminus S]}(v)\} \\
\cup \quad & \{\mathbf{r}_{t'}(S, S_{1+}, S_{1-}, S_*, S'_{3+}, S_{3-}) \\
& \qquad \mid (S, S_{1+}, S_{1-}, S_*, S'_{3+}, S_{3-}) \in \mathcal{R}_{t'}, \\
& \qquad\ S_{3+} = S'_{3+} \cup \{v\}, \\
& \qquad\ [N_{G_t[X_t \setminus S']}(v) = \emptyset] \text{ or} \\
& \qquad\ [z \in S'_{3+},\ N_{G_t[X_t \setminus S]}(v) = \{z\},\ N_{G_t[X_t \setminus S]}(z) = \{v\}]\} \\
\cup \quad & \{\mathbf{r}_{t'}(S, S_{1+}, S_{1-}, S_*, S'_{3+}, S'_{3-}) \\
& \qquad \mid (S, S_{1+}, S_{1-}, S_*, S'_{3+}, S_{3-}) \in \mathcal{R}_{t'}, \\
& \qquad\ S_{3+} = S'_{3+} \setminus \{z, z'\},\ S_{3-} = S'_{3-} \cup \{z, z', v\}, \\
& \qquad\ z, z' \in S'_{3+},\ N_{G_t[X_t \setminus S]}(v) = \{z, z'\}, \\
& \qquad\ N_{G_t[X_t \setminus S]}(z) = \{v, z'\},\ N_{G_t[X_t \setminus S]}(z') = \{v, z\}\} \ \Big).
\end{aligned}
$$

Forget vertex. If v is the forget vertex of X_t and t' is the child of t, then, for each $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_t$,

$$
\begin{aligned}
\mathbf{r}_t(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \quad = \quad & \min\{\mathbf{r}_{t'}(S', S_{1+}, S'_{1-}, S'_*, S_{3+}, S'_{3-}) \\
& \quad \mid (S', S_{1+}, S'_{1-}, S'_*, S_{3+}, S'_{3-}) \in \mathcal{I}_{t'}, \\
& \quad\ S = S' \setminus \{v\},\ S_{1-} = S'_{1-} \setminus \{v\}, \\
& \quad\ S_* = S'_* \setminus \{v\},\ S_{3-} = S'_{3-} \setminus \{v\}\}.
\end{aligned}
$$

Join. If t' and t'' are the children of t, then for each $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_t$, $\mathbf{r}_t(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-})$ is

$$
\begin{aligned}
& \min\{\mathbf{r}_{t'}(S, S'_{1+}, S'_{1-}, S_*, S'_{3+}, S'_{3-}) + \mathbf{r}_{t'}(S, S'_{1+}, S'_{1-}, S_*, S'_{3+}, S'_{3-}) - |S| \\
& \qquad \mid (S, S'_{1+}, S'_{1-}, S_*, S'_{3+}, S'_{3-}) \in \mathcal{I}_{t'},\ (S, S''_{1+}, S''_{1-}, S_*, S''_{3+}, S''_{3-}) \in \mathcal{I}_{t''}, \\
& \qquad (S'_{1+} \cup S'_{1-}) \cap (S''_{3+} \cup S''_{3-}) = (S''_{1+} \cup S''_{1-}) \cap (S'_{3+} \cup S'_{3-}) = \emptyset, \\
& \qquad \forall v \in S'_{1-} \cap S''_{1-},\ \exists z \in S_* : N_{G_t[X_t \setminus S]}(v) = \{z\}, \\
& \qquad \forall v \in S'_{3-} \cap S''_{3-},\ \exists z, z' \in S'_{3-} \cap S''_{3-} : v, z, z' \text{ induce a } C_3 \text{ in } G_t[X_t \setminus S]\}.
\end{aligned}
$$

Let us analyze the running time of this algorithm. As, for each $t \in V(T)$, S, S_{1+}, S_{1-}, S_*, S_{3+}, and S_{3-} form a partition of X_t, we have that $|\mathcal{I}_t| \leq 6^{|X_t|}$. Note that if t is a leaf, then $\mathbf{r}_t$ can be computed in time $\mathcal{O}(1)$, if t is an introduce vertex or a forget vertex node, and t' is the child of t, then $\mathbf{r}_t$ can be computed in time $\mathcal{O}(|\mathcal{I}_{t'}| \cdot |X_t|)$, and if t is a join node, and t' and t'' are the two children of t, then $\mathbf{r}_t$ can be computed in time $\mathcal{O}(|\mathcal{I}_{t'}| \cdot |\mathcal{I}_{t''}| \cdot |X_t|)$.

We now show that for each $t \in V(T)$, $\mathbf{r}_t$ is correctly computed by the algorithm. For each $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_t$, it can be easily checked that each value ℓ we compute respects, $\mathbf{r}_t(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \leq \ell$. Conversely, we now argue that for each $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_t$, the computed value ℓ is such that for any witness $(\widehat{S}, \widehat{S}_*, \widehat{S}_{3-})$ of $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-})$ is such that $\ell \leq |\widehat{S}|$. We again distinguish the type of node t.

Leaf. This follows directly from the definition of $\mathbf{r}_t$.

Introduce vertex. Let v be the insertion vertex of X_t, let $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{R}_t$, and let $(\widehat{S}, \widehat{S}_*, \widehat{S}_{3-})$ be a witness.

- If $v \in S$, then $(S \setminus \{v\}, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_{t'}$ and $\mathbf{r}_{t'}(S \setminus \{v\}, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \leq |\widehat{S}| - 1$.
- If $v \in S_{1+}$, then v is of degree 0 in $G_t \setminus \widehat{S}$, hence $(S, S_{1+} \setminus \{v\}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_{t'}$ and $\mathbf{r}_{t'}(S, S_{1+} \setminus \{v\}, S_{1-}, S_*, S_{3+}, S_{3-}) \leq |\widehat{S}|$.
- If $v \in S_{1-}$, then v has a unique neighbor that is in $\widehat{S}_*$. As v is the insertion vertex of X_t, it implies that $N_{G_t}(v) \subseteq S_*$, and so $(S, S_{1+}, S_{1-} \setminus \{v\}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_{t'}$ and $\mathbf{r}_{t'}(S, S_{1+}, S_{1-} \setminus \{v\}, S_*, S_{3+}, S_{3-}) \leq |\widehat{S}|$.
- If $v \in S_*$, then every neighbor of v is in S_{1-} and has degree 1 in $G_t \setminus \widehat{S}$. Thus, $(S, S_{1+} \cup N_{G_t[X_t \setminus S]}(v), S_{1-} \setminus N_{G_t[X_t \setminus S]}(v), S_* \setminus \{v\}, S_{3+}, S_{3-}) \in \mathcal{I}_{t'}$ and $\mathbf{r}_{t'}(S, S_{1+} \cup N_{G_t[X_t \setminus S]}(v), S_{1-} \setminus N_{G_t[X_t \setminus S]}(v), S_* \setminus \{v\}, S_{3+}, S_{3-}) \leq |\widehat{S}|$.
- If $v \in S_{3+}$, then $(S, S_{1+}, S_{1-}, S_*, S_{3+} \setminus \{v\}, S_{3-}) \in \mathcal{I}_{t'}$ and $\mathbf{r}_{t'}(S, S_{1+}, S_{1-}, S_*, S_{3+} \setminus \{v\}, S_{3-}) \leq |\widehat{S}|$.
- If $v \in S_{3-}$, then there exist z and z' in S_{3-} such that $\{v, z, z'\}$ induce a C_3 in $G_t \setminus \widehat{S}$ and there is no edge in $G_t \setminus \widehat{S}$ between a vertex of $\{v, z, z'\}$ and a vertex of $V(G_t \setminus \widehat{S}) \setminus \{x, z, z'\}$. So $(S, S_{1+}, S_{1-}, S_*, S_{3+} \cup \{z, z'\}, S_{3-} \setminus \{x, z, z'\}) \in \mathcal{I}_{t'}$ and $\mathbf{r}_{t'}(S, S_{1+}, S_{1-}, S_*, S_{3+} \cup \{z, z'\}, S_{3-} \setminus \{x, z, z'\}) \leq |\widehat{S}|$.

Forget vertex. Let v be the forget vertex of X_t, let $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_t$, and let $(\widehat{S}, \widehat{S}_*, \widehat{S}_{3-})$ be a witness. If v has degree 0 in $G_t \setminus \widehat{S}$, then $(S, S_{1+}, S_{1-}, S_* \cup \{v\}, S_{3+}, S_{3-}) \in \mathcal{I}_{t'}$ and $\mathbf{r}_{t'}(S, S_{1+}, S_{1-}, S_* \cup \{v\}, S_{3+}, S_{3-}) \leq |\widehat{S}|$. If v has degree at least 1 in $G_t \setminus \widehat{S}$, then $N_{G_t \setminus \widehat{S}}(v) \cap S_{3+} = \emptyset$, as otherwise there would be an edge in $G_t \setminus \widehat{S}$ between a vertex of S_{3+} and a vertex of $V(G_t) \setminus (\widehat{S} \cup S_{3+})$. So, one of the following case occurs:

- $v \in \widehat{S}$, $(S \cup \{v\}, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_{t'}$, and $\mathbf{r}_{t'}(S \cup \{v\}, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \leq |\widehat{S}|$,
- $v \in \widehat{S}_*$, $(S, S_{1+}, S_{1-}, S_* \cup \{v\}, S_{3+}, S_{3-}) \in \mathcal{I}_{t'}$, and $\mathbf{r}_{t'}(S, S_{1+}, S_{1-}, S_* \cup \{v\}, S_{3+}, S_{3-}) \leq |\widehat{S}|$,
- $N_{G_t \setminus \widehat{S}}(v) \subseteq \widehat{S}_*$, $(S, S_{1+}, S_{1-} \cup \{v\}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_{t'}$, and $\mathbf{r}_{t'}(S, S_{1+}, S_{1-} \cup \{v\}, S_*, S_{3+}, S_{3-}) \leq |\widehat{S}|$, or
- $v \in \widehat{S}_{3-}$, $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-} \cup \{v\}) \in \mathcal{I}_{t'}$, and $\mathbf{r}_{t'}(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-} \cup \{v\}) \leq |\widehat{S}|$

Join. Let $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) \in \mathcal{I}_t$, and let $(\widehat{S}, \widehat{S}_*, \widehat{S}_{3-})$ be a witness. Let t' and t'' be the two children of t. We define $\widehat{S}' = \widehat{S} \cap V(G_{t'})$, $\widehat{S}'' = \widehat{S} \cap V(G_{t''})$, $\widehat{S}'_* = \widehat{S}_* \cap V(G_{t'})$, $\widehat{S}''_* = \widehat{S}_* \cap V(G_{t''})$, $\widehat{S}'_{3-} \subseteq \widehat{S}_{3-} \cap V(G_{t'})$, and $\widehat{S}''_{3-} \subseteq \widehat{S}_{3-} \cap V(G_{t''})$, such that each connected component of $G_t[\widehat{S}'_{3-}]$ (resp. $G_t[\widehat{S}''_{3-}]$) is a C_3 and $G_{t'} \setminus (\widehat{S}' \cup \widehat{S}'_{3-})$ (resp. $G_{t''} \setminus (\widehat{S}'' \cup \widehat{S}''_{3-})$) is a forest). Then we define

- $S' = \widehat{S}' \cap X_t$,

- $S'_{1+} = S_{1+} \cup \{v \in S_{1-} \mid N_{G_t \setminus \widehat{S}}(v) \not\subseteq \widehat{S}'_*\}$,
- $S'_{1-} = \{v \in S_{1-} \mid N_{G_t \setminus \widehat{S}}(v) \subseteq \widehat{S}'_*\}$,
- $S'_* = S_* \cap V(G_{t'})$,
- $S'_{3-} = \widehat{S}'_{3-} \cap X_t$, and
- $S'_{3+} = S_{3+} \cup (S_{3-} \setminus S'_{3-})$.

Note that $(S', S'_{1+}, S'_{1-}, S'_*, S'_{3+}, S'_{3-}) \in \mathcal{I}'_t$. We define $(S'', S''_{1+}, S''_{1-}, S''_*, S''_{3+}, S''_{3-}) \in \mathcal{I}''_t$ similarly. Moreover we can easily check that

- $S = S' = S'', S_* = S'_* = S''_*$,
- $(S'_{1+} \cup S'_{1-}) \cap (S''_{3+} \cup S''_{3-}) = (S''_{1+} \cup S''_{1-}) \cap (S'_{3+} \cup S'_{3-}) = \emptyset$,
- $\forall v \in S'_{1-} \cap S''_{1-}, \exists z \in S_* : N_{G_t[X_t \setminus S]}(v) = \{z\}$,
- $\forall v \in S'_{3-} \cap S''_{3-}, \exists z, z' \in S'_{3-} \cap S''_{3-} : v, z, z'$ induce a C_3 in $G_t[X_t \setminus S]$,
- $(S, S_{1+}, S_{1-}, S_*, S_{3+}, S_{3-}) = (S, S'_{1+} \cap S''_{1+}, S'_{1-} \cup S''_{1-}, S_*, S'_{3+} \cap S''_{3+}, S'_{3-} \cup S''_{3-})$, and
- $\mathbf{r}_{t'}(S', S'_{1+}, S'_{1-}, S'_*, S'_{3+}, S'_{3-}) + \mathbf{r}_{t''}(S'', S''_{1+}, S''_{1-}, S''_*, S''_{3+}, S''_{3-}) - |S| \leq |\widehat{S}|$.

This concludes the proof of correctness of the algorithm. The following theorem summarizes the above discussion.

▶ **Theorem 38.** *If a nice tree decomposition of G of width w is given, $\{P_4\}$-*Deletion *can be solved in time $\mathcal{O}(36^w \cdot w \cdot n)$.*

I Deferred contents of Section 5

In this section we present the deferred contents of Section 5. Namely, in Subsection I.1 we present the proofs, figures, and results that have been omitted from the general construction. Then, in Subsection I.2 (resp. Subsection I.3) we explain how to conclude the reduction when $\mathcal{F}$ is a subset of $\mathcal{P}$ (resp. subset of $\mathcal{K}$). Finally, we explain in Subsection I.4 the changes to be made to prove the hardness result for the minor version.

I.1 Deferred contents of the general construction

Proof of Proposition 21. In any solution we need, for each $i \in [1, s]$, to take two vertices among x_i, z_{2i-1}, and z_{2i}. This implies that any solution is of size at least $2s$. Moreover, for every $i \in [1, s]$, the set $\{x_j \mid j \neq i\} \cup \{z_{2j-1} \mid j \in [1, i]\} \cup \{z_{2j} \mid j \in [i, s]\}$ is a solution of $\mathcal{F}$-TM-Deletion in the H-choice gadget of size $2s$. This can be seen using Lemma 2 and the fact that $\mathcal{F}$ is a topological minor antichain, as we did in the proof of Theorem 33. Finally, note that any solution need to contain at least $s+1$ vertices from $\{z_i \mid i \in [0, 2s+1]\}$. This implies that in any solution S of size at most $2s$, at least one vertex from $\{x_i \mid i \in [1, s]\}$ is not in S. ◀

Definition of the σ-general H-solution. Let $\sigma : [1, k] \to [1, k]$ be a permutation. If $\{(\sigma(i), i) \mid i \in [1, k]\}$ induces a k-clique in G, then we say that σ is a solution of $k \times k$ Permutation Clique on (G, k). We denote by $e^{\sigma}_{p,q}$ the edge $\{(\sigma(p), p), (\sigma(q), q)\}$. With such σ, we associate a *σ-general H-solution* S such that

- $S \subseteq V(G')$,
- $|S| = 3|E(G)| + 2k^2$,
- for each $j \in [1, k]$, S restricted to the H-choice gadget connecting $\{t_{i,j} \mid i \in [1, k]\}$ is a solution of $\mathcal{F}$-TM-Deletion of size $2k$ such that $\{t_{i,j} \mid i \in [1, k]\} \setminus \{t_{\sigma(j),j}\} \subseteq S$ and $t_{\sigma(j),j} \notin S$,

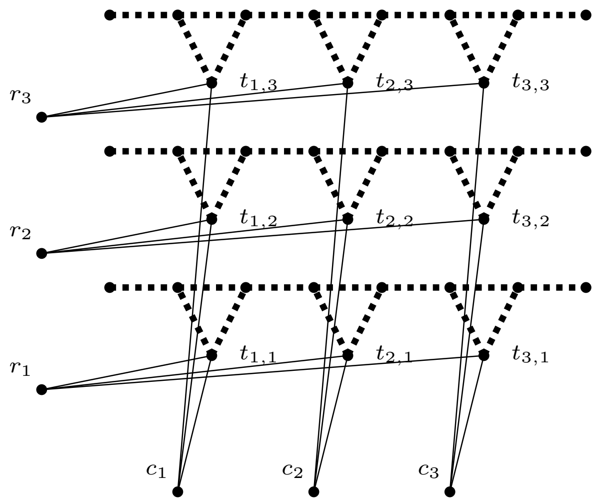

Figure 1 The general H-construction, where the dotted parts correspond to H-edge gadgets, without the encoding of the edges of $E(G)$.

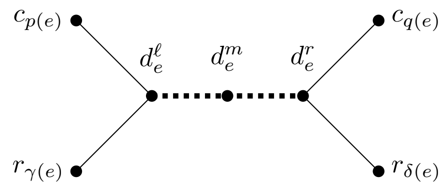

Figure 2 The encoding of an edge $e = \{(p(e), \gamma(e)), (q(e), \delta(e))\}$ of G, where the dotted lines correspond to a double H-edge gadget.

- for each $p, q \in [1, k]$, $p < q$, such that $e^\sigma_{p,q} \in E(G)$, S restricted to the H-choice gadget connecting $\{d^\ell_e \mid e \in E(p, q)\}$ is a solution of $\mathcal{F}$-TM-Deletion of size $2|E(p, q)|$ such that $\{d^\ell_e \mid e \in E(p, q) \setminus \{e^\sigma_{p,q}\}\} \subseteq S$ and $d^\ell_{e^\sigma_{p,q}} \notin S$,
- for each $p, q \in [1, k]$, $\{d^r_e \mid e \in E(p, q) \setminus \{e^\sigma_{p,q}\}\} \subseteq S$ and $d^r_{e^\sigma_{p,q}} \notin S$, and
- for each $p, q \in [1, k]$, $d^m_{e^\sigma_{p,q}} \in S$.

Note that, with this construction of S, we already impose $3|E(G)| + 2k^2$ vertices to be in S, and therefore no other vertex of G' can be in S.

Definition of the associated permutation of a set satisfying the permutation property. Given $S \subseteq V(G')$, we say that S satisfies the *permutation property* if

- for each $e \in E(G)$ such that $d^\ell_e \notin S$ and for each $j \in [1, k] \setminus \{\gamma(e)\}$, $t_{p(e),j} \in S$, and
- for each $e \in E(G)$ such that $d^r_e \notin S$ and for each $j \in [1, k] \setminus \{\delta(e)\}$, $t_{q(e),j} \in S$.

▶ **Lemma 39.** *If S is a subset of $V(G')$ such that*
- $|S| \leq 3|E(G)| + 2k^2$,
- *for each H-choice gadget, S restricted to this H-choice gadget is a solution of $\mathcal{F}$-TM-Deletion of minimum size,*
- *for each edge $e \in E(G)$, S is such that either $\{d^\ell_e, d^m_e, d^r_e\} \cap S = \{d^\ell_e, d^r_e\}$ or $\{d^\ell_e, d^m_e, d^r_e\} \cap S = \{d^m_e\}$, and*
- *S satisfies the permutation property,*

then

- *there is a unique permutation* $\sigma : [1,k] \to [1,k]$ *such that* $S \cap \{t_{i,j} \mid i,j \in [1,k]\} = \{t_{i,j} \mid i,j \in [1,k]\} \setminus \{t_{\sigma(j),j} \mid j \in [1,k]\}$ *and*
- $\{(\sigma(j), j) \mid j \in [1,k]\}$ *induces a clique in* G.

Proof. By hypothesis, for each $1 \le p < q \le k$, we know that there is an edge $e \in E(p,q)$ such that $d_e^\ell \notin S$ and $d_e^r \notin S$. As S satisfies the permutation property, this implies that for each set $\{t_{i,j} \mid i \in [1,k]\}$, $j \in [1,k]$, at most one vertex is not in S. As we have supposed that for each set $\{t_{i,j} \mid j \in [1,k]\}$, $i \in [1,k]$, at least one vertex is not in S, this implies that there is a unique permutation $\sigma : [1,k] \to [1,k]$ such that $S \cap \{t_{i,j} \mid i,j \in [1,k]\} = \{t_{i,j} \mid i,j \in [1,k]\} \setminus \{t_{\sigma(j),j} \mid j \in [1,k]\}$. Moreover, for each $1 \le p < q \le k$, $e_{p,q}^\sigma \in E(G)$. Indeed, if $e_{p,q}^\sigma \notin E(G)$, then there exists $e \in E(p,q)$ such that $\sigma(\gamma(e)) \neq p$ or $\sigma(\delta(e)) \neq q$ and such that $d_e^\ell \notin S$ and $d_e^r \notin S$. Assume without loss of generality, that $\sigma(\gamma(e)) \neq p$. As $t_{p,\sigma^{-1}(p)} \notin S$, this contradicts the fact that S satisfies the permutation property. ◀

We call the permutation given by Lemma 39 *the associated permutation* of S.

I.2 Reduction for paths

▶ **Theorem 40.** *Given an integer* $h \ge 6$, *the* $\{P_h\}$-TM-DELETION *problem cannot be solved in time* $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$, *unless the ETH fails.*

Proof. Let $h \ge 6$ be an integer and let $\mathcal{F} = \{P_h\}$. Let (G,k) be an instance of $k \times k$ PERMUTATION CLIQUE and let (G', ℓ) be the general P_h-construction of (G,k) described in Section 5. We build G'' starting from G' by adding, for each $i,j \in [1,k]$, a pendant path of size $h-6$ to $t_{i,j}$. This completes the construction of G''. Note that if $h = 6$, then $G'' = G'$.

Let σ be a solution of $k \times k$ PERMUTATION CLIQUE on (G,k) and let S be the σ-general P_h-solution. To show that $G'' \setminus S$ does not contain P_h as a topological minor, we show that each connected component of $G'' \setminus S$ does not contain P_h as a topological minor. Note that, by definition of S, the only connected components of $G'' \setminus S$ that can contain P_h are the connected components that contain a vertex in $\{c_i \mid i \in [1,k]\} \cup \{r_j \mid j \in [1,k]\}$. Moreover, each of these connected components is a subgraph of G'' induced by $\{r_j, c_{\sigma(j)}, t_{\sigma(j),j}\} \cup \{d_e^\ell \mid e \in E(G), \sigma(\gamma(e)) = p(e), \sigma(\delta(e)) = q(e), p(e) = j\} \cup \{d_e^r \mid e \in E(G), \sigma(\gamma(e)) = p(e), \sigma(\delta(e)) = q(e), q(e) = j\}$ and the vertices of the path of size $h-6$ pendant to $t_{\sigma(j),j}$, for some $j \in [1,k]$. These connected components, depicted in Figure 3 for the case $h = 8$, do not contain P_h as a topological minor, as it can be easily checked that a longest path in them has $h-1$ vertices. Therefore, S is a solution of $\mathcal{F}$-TM-DELETION on (G'', ℓ).

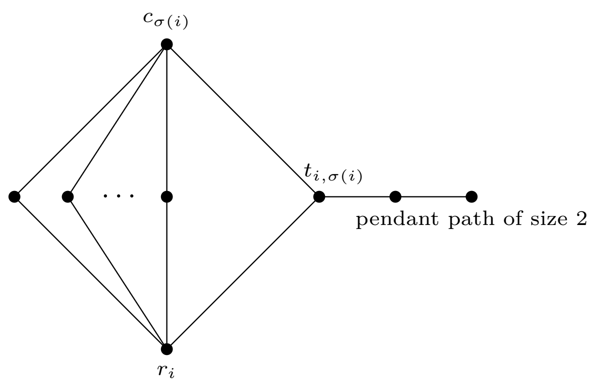

Figure 3 The connected component of $G'' \setminus S$ that contains the vertex r_i, $i \in [1,k]$ where $h = 8$.

Conversely, let S be a solution of $\mathcal{F}$-TM-DELETION on (G'', ℓ). We first show that S satisfies the permutation property. Let $e \in E(G)$ be such that $d_e^\ell \notin S$ and assume that $t_{p(e),j} \notin S$ for some $j \in [1,k] \setminus \{\gamma(e)\}$. By construction of the P_h-choice gadget, we know that there exists $i_0 \in [1,k]$ such that $t_{i_0,\gamma(e)} \notin S$. This implies that the path $r_j t_{p(e),j} c_{p(e)} d_e^\ell r_{\gamma(e)} t_{i_0,\gamma(e)}$ together with the pendant path with $h-6$ vertices attached to $t_{i_0,\gamma(e)}$ form a path with h vertices. As the same argument also works if $d_e^r \notin S$, it follows that S satisfies the permutation property, concluding the proof. ◀

I.3 Reduction for subsets of $\mathcal{K}$

▶ **Theorem 41.** *Given a regular collection $\mathcal{F} \subseteq \mathcal{K}$, the $\mathcal{F}$-TM-DELETION problem cannot be solved in time $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$, unless the ETH fails.*

Proof. Let $\mathcal{F} \subseteq \mathcal{K}$ be a regular collection. We assume without loss of generality that $\mathcal{F}$ is a topological minor antichain. Let (H, B) be an essential pair of $\mathcal{F}$, let a be the first vertex of (H, B), let b be the second vertex of (H, B), let $\overline{B} = (V(B), E(B) \setminus \{a, b\})$, and let C be the core of (H, B). Let (G, k) be an instance of $k \times k$ PERMUTATION CLIQUE and let (G', ℓ) be the general H-construction of (G, k). We build G'' starting from G' by adding a new vertex r_0, adding a copy of the core of (H, B) and identifying a and r_0, and adding for each $j \in [1,k]$ a copy of $\overline{B}$ in which we identify a and r_0, and b and r_j. This completes the construction of G''.

Let σ be a solution of $k \times k$ PERMUTATION CLIQUE on (G, k) and let S be the σ-general H-solution. We show that $\mathcal{F} \npreceq_{\mathsf{tm}} G'' \setminus S$. For this, let us fix $H' \in \mathcal{F}$. Note that, by definition of S, the only connected component of $G'' \setminus S$ that can contain H' is the one containing $\{c_i \mid i \in [1,k]\} \cup \{r_j \mid j \in [1,k]\}$. Let K be this connected component, depicted in Figure 4. Note that for each $j \in [0,k]$, r_j is a cut vertex of K. Moreover, we know that $H' \npreceq_{\mathsf{tm}} C$ and for each $B' \in L(\mathsf{bct}(H'))$, $B' \npreceq_{\mathsf{tm}} \overline{B}$ and $B' \npreceq_{\mathsf{tm}} K_{2,k}$. This implies, using Lemma 2, that H' is not a topological minor of K. Therefore, S is a solution of $\mathcal{F}$-TM-DELETION on (G'', ℓ).

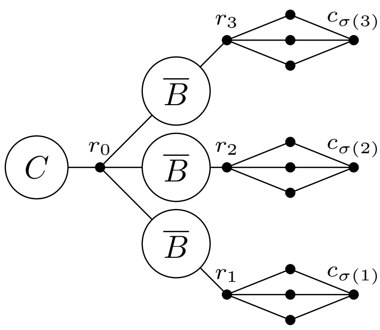

Figure 4 The main connected component of $G'' \setminus S$.

Conversely, let S be a solution of $\mathcal{F}$-TM-DELETION on (G'', ℓ). We show that S satisfies the permutation property. Let $e \in E(G)$ be such that $d_e^\ell \notin S$ and assume that $t_{p(e),j} \notin S$ for some $j \in [1,k] \setminus \{\gamma(e)\}$. This implies that there exists in $G'' \setminus S$ a $(r_0, r_{\gamma(e)})$-path that uses the vertices r_0, r_j, $t_{p(e),j}$, $c_{p(e)}$, d_e^ℓ, and $r_{\gamma(e)}$ and that does not use any edge of the graph $\overline{B} = (V(B), E(B) \setminus \{a, b\})$ between r_0 and $r_{\gamma(e)}$. By construction, this implies that H is a topological minor of $G'' \setminus S$. As the same argument also works if $d_e^r \notin S$, it follows that S satisfies the permutation property, concluding the proof. ◀

I.4 The minor version

Theorems 40 and 41 allow to prove the statement of Theorem 18 for topological minors. The equivalent statement of Theorem 18 for minors follows from the same proof as for the topological minor version by applying the following local modifications:

- we replace $\mathcal{F}$-TM-Deletion with $\mathcal{F}$-M-Deletion,
- we replace topological minor with minor,
- we replace $\preceq_{\mathsf{tm}}$ with $\preceq_{\mathsf{m}}$, and
- we replace Lemma 2 with Lemma 34.