Optimal Algorithms for Hitting (Topological) Minors on Graphs of Bounded Treewidth*†

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

1 **Université de Montpellier, LIRMM, Montpellier, France**
baste@lirmm.fr

2 **AlGCo project-team, CNRS, LIRMM, France and Departamento de Matemática, Universidade Federal do Ceará, Fortaleza, Brazil**
ignasi.sau@lirmm.fr

3 **AlGCo project-team, CNRS, LIRMM, France and Department of Mathematics, National and Kapodistrian University of Athens, Greece**
sedthilk@thilikos.info

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.IPEC.2017.4

* This work has been supported by project DEMOGRAPH (ANR-16-CE40-0028).
† A full version of this article is permanently available at https://arxiv.org/abs/1704.07284.

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 $\mathcal{F}$-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 [15].

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 [14] for explicit formulas), by Courcelle's theorem [5], $\mathcal{F}$-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 [12].

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 [12] is again $\mathcal{O}^*(2^{o(\mathbf{tw})})$. This gap remained open for a while, until Cygan *et al.* [6] 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 [2, 8].

Concerning Vertex Planarization, Jansen *et al.* [13] 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 [19] proved that this running time is *optimal* under the ETH, by using the framework introduced by Lokshtanov *et al.* [17] 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 $i \leq 3$ [6], and that the case $i = 4$ is tight by item **2** above (as K_4 is planar).

[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 [12] for more details.
[3] A *connected* collection $\mathcal{F}$ is a collection containing only connected graphs.

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.

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

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.

The results discussed in the last three items are summarized in Table 1. Note that the cases with $i \leq 3$ were already known [6, 12], except when $\mathcal{F} = \{P_3\}$.

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.* [3], and which has been subsequently used in [9, 10, 14]. For technical reasons, we use *branch* decompositions instead of tree decompositions, whose associated widths are equivalent from a parametric point of view [20].

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. the full version). 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 11).

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 9).

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.* [2], exploiting again the structure of graphs that do not contain C_4 as a minor (cf. Lemma 14). 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 [19] for Vertex Planarization. We present a general hardness result (cf. Theorem 20) that applies to wide families of connected collections $\mathcal{F}$. Then, our superexponential lower bounds, as well as the result of Marcin Pilipczuk [19] itself, are corollaries of this general result. Combining Theorem 20 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 [7, 14]. 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.* [16] to prove lower bounds based on the *Strong* Exponential Time Hypothesis.

Finally, let us mention that Bonnet *et al.* [4] 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 superexponential lower bounds are presented in Section 5. The general lower bound for connected collections and the proofs of all the results marked with '($\star$)' can be found in the full version.

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 the full version.

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 [11]. 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 3** ($\star$). *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 4** ($\star$). *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 *proper collection*. For any proper 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 proper 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 proper 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 proper 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 proper 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 $\mathcal{F}$ 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 $tm_{\mathcal{F}}$

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

▶ **Theorem 5.** *If $\mathcal{F}$ is a proper 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 6.** *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 7.** *If $\mathcal{F}$ is a proper 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 8.** *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 9.** *Let $\mathcal{F}$ be a proper 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 9 and Observation 1, Theorems 7 and 8 follow directly from Theorems 5 and 6, respectively. The rest of this section is dedicated to the proofs of Theorems 5 and 6. 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 [3,9] (see also [10,14]), and can be found in the full version. 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 proper 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 equivalence 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 $\mathsf{folio}(\mathbf{B},\mathcal{F},r)$, as the set containing all boundaried graphs in $\mathcal{A}^{(t)}_{\mathcal{F},r}$ that are topological minors of $\mathbf{B}$.

▶ **Lemma 10** ($\star$). *There exists a function $h_1 : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ such that if $\mathcal{F}$ is a proper 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 11** ($\star$). *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 5; the proof can be found in the full version. In order to prove Theorem 6, we need Lemma 13 below, which should be contrasted with Lemma 10. Its proof, which can be found in the full version, uses, among others, the following result of Baste *et al.* [1] on the number of labeled graphs of bounded treewidth.

▶ **Proposition 12** (Baste *et al.* [1]). *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 13** ($\star$). *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 6 can be found in the full version. The main difference with respect to the proof of Theorem 5 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 [18].

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

We proceed to use the dynamic programming techniques introduced by Bodlaender *et al.* [2] 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 14 a structural characterization of the graphs that exclude C_4 as a (topological) minor.

▶ **Lemma 14** ($\star$). *Let G be a graph. $C_4 \not\preceq_{\mathsf{tm}} G$ if and only if G satisfies the C_4-condition.*

▶ **Lemma 15** ($\star$). *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.* [2] 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 16** (Bodlaender *et al.* [2]). *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 17** (Bodlaender *et al.* [2]). *The union, insert, shift, glue, project, and join operations preserve representation.*

▶ **Theorem 18** (Bodlaender *et al.* [2]). *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 [2, Section 3.5] 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.* [2] 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:

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

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 15 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 14, and Lemma 15 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 the full version 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 the full version.

▶ **Theorem 19** (⋆). $\{C_4\}$-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 20.** *Let $\mathcal{F}$ be a proper 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 [19] 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 20 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 21.** *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 20 for $\mathcal{F}$-TM-Deletion, and we explain in the full version how to modify the proof to obtain the result for $\mathcal{F}$-M-Deletion. To prove Theorem 20, we reduce from $k \times k$ Permutation Clique ($k \times k$ P. Clique for short), defined by Lokshtanov *et al.* [17]. 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.* [17] 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 proper 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 [19]. 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 22.** *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 22. 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 [19, Lemma 5].

▶ **Proposition 23** ($\star$)**.** *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 illustrated in the full version.

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^ℓ_e, d^m_e, and d^r_e, and four edges $\{d^\ell_e, c_{p(e)}\}$, $\{d^\ell_e, r_{\gamma(e)}\}$, $\{d^r_e, c_{q(e)}\}$, and $\{d^r_e, r_{\delta(e)}\}$. We introduce a double H-edge gadget between d^ℓ_e and d^r_e through d^m_e. The encoding of an edge $e \in E(G)$ is also illustrated in the full version. 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 $\{d^\ell_e \mid e \in E(p,q)\}$.

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 23 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^m_e or both d^ℓ_e and d^r_e. The extra budget given for each edge permits to include d^m_e in the solution. If the H-choice gadget connected to d^ℓ_e already chooses d^ℓ_e to be in the solution, then we can use the extra budget given for the edge e to choose d^r_e instead of d^m_e. In the case d^m_e 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 $\{d^\ell_e, d^m_e, d^r_e\} \cap S = \{d^\ell_e, d^r_e\}$ or $\{d^\ell_e, d^m_e, d^r_e\} \cap S = \{d^m_e\}$ 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^r_e and d^m_e, $e \in E(G)$. 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 the full version 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. the full version) 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 allows to prove 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 the full version.

References

1 Julien Baste, Marc Noy, and Ignasi Sau. On the number of labeled graphs of bounded treewidth. *CoRR*, abs/1604.07273, 2016. To appear in *Proc. of WG 2017*.

2 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86–111, 2015.

3 Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (meta) kernelization. *Journal of the ACM*, 63(5):44:1–44:69, 2016.

4 Édouard Bonnet, Nick Brettell, O-joung Kwon, and Dániel Marx. Generalized feedback vertex set problems on bounded-treewidth graphs: chordality is the key to single-exponential parameterized algorithms. *CoRR*, abs/1704.06757, 2017.

5 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

6 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry 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.

7 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar $\mathcal{F}$-Deletion: Approximation, Kernelization and Optimal FPT Algorithms. In *Proc. of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 470–479, 2012.

8 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *Journal of the ACM*, 63(4):29:1–29:60, 2016.

9 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios 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.

10 Valentin Garnero, Christophe Paul, Ignasi Sau, and Dimitrios M. Thilikos. Explicit linear kernels via dynamic programming. *SIAM Journal on Discrete Mathematics*, 29(4):1864–1894, 2015.

11 John E. Hopcroft and Robert Endre Tarjan. Efficient algorithms for graph manipulation. *Communations of ACM*, 16(6):372–378, 1973.

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

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

14 Eun Jung Kim, Alexander Langer, Christophe Paul, Felix Reidl, Peter Rossmanith, Ignasi Sau, and Somnath Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. *ACM Transactions on Algorithms*, 12(2):21:1–21:41, 2016.

15 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.

16 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.

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

18 Doowon Paik, Sudhakar M. Reddy, and Sartaj Sahni. Deleting vertices to bound path length. *IEEE Trans. Computers*, 43(9):1091–1096, 1994. doi:10.1109/12.312117.

19 Marcin Pilipczuk. A tight lower bound for Vertex Planarization on graphs of bounded treewidth. *Discrete Applied Mathematics*, 231:211–216, 2017.

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