

Hitting Minors on Bounded Treewidth Graphs. I. General Upper Bounds

Julien Baste, Ignasi Sau Valls, Dimitrios M. Thilikos

► **To cite this version:**

Julien Baste, Ignasi Sau Valls, Dimitrios M. Thilikos. Hitting Minors on Bounded Treewidth Graphs. I. General Upper Bounds. *SIAM Journal on Discrete Mathematics*, Society for Industrial and Applied Mathematics, 2020, 34 (3), pp.1623-1648. 10.1137/19M1287146 . lirmm-02989921

HAL Id: lirmm-02989921

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-02989921>

Submitted on 5 Nov 2020

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

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

HITTING MINORS ON BOUNDED TREewidth GRAPHS. I. GENERAL UPPER BOUNDS*

JULIEN BASTE[†], IGNASI SAU[‡], AND DIMITRIOS M. THILIKOS[‡]

Abstract. For a finite collection of graphs \mathcal{F} , the \mathcal{F} -M-DELETION problem consists in, given a graph G and an integer k , deciding 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 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 $f_{\mathcal{F}}(\text{tw}) \cdot n^{\mathcal{O}(1)}$ on n -vertex graphs. We prove that $f_{\mathcal{F}}(\text{tw}) = 2^{2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})}}$ for every collection \mathcal{F} , that $f_{\mathcal{F}}(\text{tw}) = 2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})}$ if \mathcal{F} contains a planar graph, and that $f_{\mathcal{F}}(\text{tw}) = 2^{\mathcal{O}(\text{tw})}$ if in addition the input graph G is planar or embedded in a surface. We also consider the version of the problem where the graphs in \mathcal{F} are forbidden as *topological* minors, called \mathcal{F} -TM-DELETION. We prove similar results for this problem, except that in the last two algorithms, instead of requiring \mathcal{F} to contain a planar graph, we need it to contain a *subcubic* planar graph. This is the first of a series of articles on this topic.

Key words. parameterized complexity, graph minors, treewidth, hitting minors, topological minors, dynamic programming, Exponential Time Hypothesis.

AMS subject classifications. G.2.2

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 belong to the wider family of *graph modification problems* and have a big expressive power, as instantiations of them correspond to several well-studied 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 classification result of Lewis and Yannakakis [36].

We are interested in the parameterized complexity of \mathcal{F} -DELETION when the parameter is the treewidth of the input graph (formally defined in Section 2). Since the property of containing a graph as a (topological) minor can be expressed in Monadic Second Order logic (see [32] for explicit formulas), by Courcelle's theorem [12], \mathcal{F} -DELETION can be solved in time $\mathcal{O}^*(f(\text{tw}))$ on graphs with treewidth at most tw , where f is some computable function¹. Our objective is to determine, for a fixed collection \mathcal{F} , which is the *smallest* such function f that one can (asymptotically)

*An extended abstract containing some of the results of this article appeared in the *Proc. of the 12th International Symposium on Parameterized and Exact Computation (IPEC 2017)* [3].

Funding: Work supported by French projects DEMOGRAPH (ANR-16-CE40-0028) and ES-IGMA (ANR-17-CE23-0010), and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 388217545.

[†]LIRMM, Univ Montpellier, Montpellier, France, and Sorbonne Université, Laboratoire d'Informatique de Paris 6, LIP6, Paris, France (julien.baste@uni-ulm.de).

[‡]LIRMM, Univ Montpellier, CNRS, Montpellier, France (ignasi.sau@lirmm.fr, sedthilk@thilikos.info).

¹The notation $\mathcal{O}^*(\cdot)$ suppresses polynomial factors depending on the size of the input graph.

hope for, subject to reasonable complexity assumptions.

This line of research has attracted some attention in the parameterized complexity community during the last years. For instance, VERTEX COVER is easily solvable in time $\mathcal{O}^*(2^{\mathcal{O}(\text{tw})})$, called *single-exponential*, by standard dynamic-programming techniques, and no algorithm with running time $2^{o(\text{tw})} \cdot n^{\mathcal{O}(1)}$ exists, unless the Exponential Time Hypothesis (ETH)² fails [29].

For FEEDBACK VERTEX SET, standard dynamic programming techniques give a running time of $\mathcal{O}^*(2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})})$, while the lower bound under the ETH [29] is again $2^{o(\text{tw})} \cdot n^{\mathcal{O}(1)}$. This gap remained open for a while, until Cygan et al. [14] presented an optimal algorithm running in time $\mathcal{O}^*(2^{\mathcal{O}(\text{tw})})$, introducing the celebrated *Cut&Count* technique. This article triggered several other techniques to obtain single-exponential algorithms for so-called *connectivity problems* on graphs of bounded treewidth, mostly based on algebraic tools [7, 23].

Concerning VERTEX PLANARIZATION, Jansen et al. [30] presented an algorithm of time $\mathcal{O}^*(2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})})$ as a crucial subroutine in an FPT-algorithm parameterized by k . Marcin Pilipczuk [39] proved that this running time is *optimal* under the ETH, by using the framework introduced by Lokshtanov et al. [37] for proving superexponential lower bounds.

Our results and techniques. We present the following algorithms for \mathcal{F} -DELETION parameterized by treewidth:

1. For every \mathcal{F} , \mathcal{F} -DELETION can be solved in time $\mathcal{O}^*(2^{2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})}})$.
2. For every collection \mathcal{F} containing at least one planar graph³ (resp. subcubic planar graph), \mathcal{F} -M-DELETION (resp. \mathcal{F} -TM-DELETION) can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})})$.
3. If the input graph G is planar or, more generally, embedded in a surface of bounded genus, then \mathcal{F} -M-DELETION (resp. \mathcal{F} -TM-DELETION if additionally \mathcal{F} contains a subcubic planar graph) can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(\text{tw})})$ for every collection \mathcal{F} .

Let us provide some ideas of the techniques that we use in our algorithms. Our first algorithm running in time $\mathcal{O}^*(2^{2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})}})$ is not complicated and the running time is probably quite natural for anyone used to dynamic programming on graphs of bounded treewidth. The encoding that we use in the tables of the dynamic programming algorithm is based in the notion of *folio*, which had been already used in previous work [1, 42]. Informally speaking, the folio of a graph contains all the “partial models” of the graphs in \mathcal{F} that survive after the removal of a partial solution (see Section 2 for the formal definition). It is easy to see that keeping track of the folios is enough to solve both problems and that the size of a folio is bounded by $2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})}$ (cf. Lemma 4.3), hence the number of distinct folios is at most $2^{2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})}}$, yielding the claimed running time.

The algorithm running in time $\mathcal{O}^*(2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})})$, when \mathcal{F} contains a (subcubic) planar graph, uses the machinery of bounded graphs, equivalence relations, and representatives originating in the seminal work of Bodlaender et al. [9] (see also [10]) and subsequently used, for instance, in [25, 24, 32]. The main conceptual difference with respect to the algorithm discussed above is that the encoding in the tables of

²The ETH implies that 3-SAT on n variables cannot be solved in time $2^{o(n)}$; see [29] for more details.

³In the conference version of this paper [3], we further required that all the graphs in \mathcal{F} are connected; here we improve the result by dropping this assumption.

the dynamic programming algorithm, which we use to construct the partial solutions, is not based on the notion of folio anymore, but on the notion of *representative* of an appropriately defined equivalence relation. Intuitively, such a representative corresponds to a possible behavior of a boundaried graph (associated with the subgraph of the input graph G rooted at a bag of a tree decomposition) with respect to the eventual occurrences of graphs in \mathcal{F} when “gluing” another unknown boundaried graph to it, corresponding to the subgraph of G that has not been explored yet (again, see Section 2 for the formal definition). 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 5.3) and this is crucially used in order to bound the number of representatives (cf. Proposition 7.5). For technical reasons, in all our algorithms we use *branch* decompositions instead of tree decompositions, whose associated widths are equivalent from a parametric point of view [41].

The algorithm running in time $\mathcal{O}^*(2^{\mathcal{O}(\text{tw})})$, when the input graph G is planar, exploits sphere-cut decompositions [47, 19], a special type of branch decompositions of planar graphs with nice topological properties. We prove that, if we use sphere-cut decompositions and we apply essentially the same dynamic programming algorithm discussed above, the number of representatives can be upper-bounded by the number of (unlabeled) planar graphs on $\mathcal{O}(\text{tw})$ vertices, which are $2^{\mathcal{O}(\text{tw})}$ many [48]. With some more technical details, we extend this single-exponential algorithm to graphs embedded in surfaces by using surface-cut decompositions, introduced by Rué et al. [44].

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 3.7).

Results in other articles of the series. In the second article of this series [5], we show that if $\mathcal{F} \in \{\{P_3\}, \{P_4\}, \{K_{1,i}\}, \{C_4\}, \{\text{paw}\}, \{\text{chair}\}, \{\text{banner}\}\}$, then \mathcal{F} -TM-DELETION can be solved in single-exponential time. Note that all these graphs have maximum degree at most three, except $K_{1,i}$ for $i \geq 4$, and therefore the corresponding algorithms also apply to the \mathcal{F} -M-DELETION problem. In the third article of this series [6], we focus on lower bounds under the ETH. Namely, we prove that for any collection \mathcal{F} containing only connected graphs of size at least two, \mathcal{F} -DELETION cannot be solved in time $2^{\mathcal{O}(\text{tw})} \cdot n^{\mathcal{O}(1)}$, even if the input graph G is planar, and we also provide superexponential lower bounds for a number of collections \mathcal{F} . In particular, we prove a lower bound of $2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$ when \mathcal{F} contains a single graph that is either P_5 or is not a minor of the banner (that is, the graph consisting of a C_4 plus a pendent edge), with the exception of $K_{1,i}$ for the topological minor version. These lower bounds, together with the ad hoc single-exponential algorithms given in [6] and the algorithm described in item 2 above, cover all the cases of \mathcal{F} -M-DELETION where \mathcal{F} consists of a single connected planar graph H , yielding a tight dichotomy in terms of H . In a recent article [4], we presented an algorithm for \mathcal{F} -M-DELETION in time $\mathcal{O}^*(2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})})$ for any collection \mathcal{F} , yielding together with the lower bounds in [6] a dichotomy for \mathcal{F} -M-DELETION where \mathcal{F} consists of a single connected (not necessarily planar) graph H .

Organization of the paper. In Section 2 we give some preliminaries, and in Section 3 we formally state the results of this article. In Section 4 we introduce the formalism of boundaried graphs and their equivalence classes, and prove several technical lemmas. In Section 5 we define branch decompositions of boundaried graphs and prove some basic properties. We prove the result of item 1 in Section 6. In Section 7 we provide improved bounds on the sets of representatives in the case where \mathcal{F} contains a planar (subcubic) graph, and we use this result in Section 8 to prove the

result of item 2. Finally, we prove the result of item 3 in Section 9 (for planar graphs) and Section 10 (for bounded-genus graphs). We conclude the article in Section 11 with some open questions for further research.

2. Preliminaries. In this section we provide some preliminaries to be used in the following sections.

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 $\mathbb{N}_{\geq p} = \mathbb{N} \setminus [0, p-1]$.

We use \emptyset to denote the empty set and \emptyset to denote the empty function, i.e., the unique subset of $\emptyset \times \emptyset$. Given a function $f : A \rightarrow B$ 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 two. We also denote by 2^S the set of all the subsets of S . If S is a collection of objects where the operation \cup is defined, then we denote $\mathbf{U}S = \bigcup_{X \in S} X$.

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

Graphs. All the graphs that we consider in this paper are undirected, finite, and without loops or multiple edges (except for the graph θ_s , sometimes called *pumpkin* in the literature [31]). We use standard graph-theoretic notation, and we refer the reader to [18] 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 zero. 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 $\rho : V(G) \rightarrow \mathbb{N}^+$. 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 $\deg_G(v) = |N_G(v)|$. A graph is called *subcubic* if all its vertices have degree at most three.

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 two 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 $\deg_P(v) = 2$. 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 , P_r , and C_r , the complete graph, the path, and the cycle on r vertices, respectively.

Minors and topological minors. Given two graphs H and G and two functions $\phi : V(H) \rightarrow V(G)$ and $\sigma : E(H) \rightarrow \mathcal{P}(G)$, 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 ,

- ϕ is an injective function, 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 $\phi(V(H))$, 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 $H \preceq_{\text{tm}} G$, if there is a topological minor model (ϕ, σ) of H in G .

Given two graphs H and G and a function $\phi : V(H) \rightarrow 2^{V(G)}$, 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,
- if x, y are two distinct vertices of H , then $\phi(x) \cap \phi(y) = \emptyset$, 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 $H \preceq_m G$, if there is a minor model ϕ of H in G .

Let H be a graph. We define the set of graphs $\text{tpm}(H)$ 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. The following two observations follow easily from the above definitions.

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

An explicit function f_1 as in Observation 1 can be obtained by replacing every vertex $v \in V(H)$ with $\deg_H(v) \geq 4$ by a tree whose leaf set is $N_H(v)$ and with no internal vertices of degree two. Clearly, for each vertex $v \in V(H)$, the number of such trees (each yielding a distinct graph in $\text{tpm}(H)$) and their size depend only on $\deg_H(v)$.

OBSERVATION 2. *Given two graphs H and G , H is a minor of G if and only if there exists some graph in $\text{tpm}(H)$ that is a topological minor of G .*

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 $\text{size}(\mathcal{F}) = \max\{|V(H)| \mid H \in \mathcal{F}\}$. 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 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 , $\mathcal{F} \preceq_{\text{tm}} G$ (resp. $\mathcal{F} \preceq_m G$) if and only if there exists a graph $H \in \mathcal{F}$ such that $H \preceq_{\text{tm}} G$ (resp. $H \preceq_m G$). We also denote $\text{ext}_{\text{tm}}(\mathcal{F}) = \{G \mid \mathcal{F} \not\preceq_{\text{tm}} G\}$, i.e., $\text{ext}_{\text{tm}}(\mathcal{F})$ is the class of graphs that do not contain any graph in \mathcal{F} as a topological minor. The set $\text{ext}_m(\mathcal{F})$ is defined analogously.

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 $\{x, y, z\} \subseteq V(G)$ 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 $\text{tw}(G)$, 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 [20, 13] 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 $f(k) \cdot |I|^c$.

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:

$$(2.1) \quad \mathbf{tm}_{\mathcal{F}}(G) = \min\{|S| \mid S \subseteq V(G) \wedge G \setminus S \in \mathbf{ex}_{\mathbf{tm}}(\mathcal{F})\}.$$

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} . The corresponding decision problems are formally defined as follows.

\mathcal{F} -TM-DELETION

Input: A graph G and a $k \in \mathbb{N}$.

Parameter: The treewidth of G .

Question: Is $\mathbf{tm}_{\mathcal{F}}(G) \leq k$?

\mathcal{F} -M-DELETION

Input: A graph G and a $k \in \mathbb{N}$.

Parameter: The treewidth of G .

Question: Is $\mathbf{m}_{\mathcal{F}}(G) \leq k$?

Note that in both the above 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 $H_1 \preceq_{\mathbf{tm}} H_2$, 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 tw be the number of vertices and the treewidth of the input graph of the considered problem, respectively.

3. Formal statement of the results. The purpose of the remainder of the article is to prove the following results.

THEOREM 3.1. *If \mathcal{F} is a proper collection, where $d = \text{size}(\mathcal{F})$, then there exists an algorithm that solves \mathcal{F} -TM-DELETION in $2^{2^{\mathcal{O}_d(\text{tw} \cdot \log \text{tw})}} \cdot n$ steps.*

THEOREM 3.2. *If \mathcal{F} is a proper collection, where $d = \text{size}(\mathcal{F})$, then there exists an algorithm that solves \mathcal{F} -M-DELETION in $2^{2^{\mathcal{O}_d(\text{tw} \cdot \log \text{tw})}} \cdot n$ steps.*

THEOREM 3.3. *If \mathcal{F} is a proper collection containing a subcubic planar graph, where $d = \text{size}(\mathcal{F})$, then there exists an algorithm that solves \mathcal{F} -TM-DELETION in $2^{\mathcal{O}_d(\text{tw} \cdot \log \text{tw})} \cdot n$ steps.*

THEOREM 3.4. *If \mathcal{F} is a proper collection containing a planar graph, where $d = \text{size}(\mathcal{F})$, then there exists an algorithm that solves \mathcal{F} -M-DELETION in $2^{\mathcal{O}_d(\text{tw} \cdot \log \text{tw})} \cdot n$ steps.*

THEOREM 3.5. *If \mathcal{F} is a proper collection containing a subcubic planar graph, where $d = \text{size}(\mathcal{F})$, then there exists an algorithm that solves \mathcal{F} -TM-DELETION on planar graphs in $2^{\mathcal{O}_d(\text{tw})} \cdot n + \mathcal{O}(n^3)$ steps.*

THEOREM 3.6. *If \mathcal{F} is a proper collection, where $d = \text{size}(\mathcal{F})$, then there exists an algorithm that solves \mathcal{F} -M-DELETION on planar graphs in $2^{\mathcal{O}_d(\text{tw})} \cdot n + \mathcal{O}(n^3)$ steps.*

It is worth mentioning that the lower bounds given in [6] imply that the algorithms of Theorems 3.5 and 3.6 are asymptotically *tight* under the ETH. Note also that in both theorems one can assume that \mathcal{F} contains only planar graphs, as an input planar graph G does not contain any nonplanar graph as a (topological) minor. In Section 10, we present a generalization of Theorems 3.5 and 3.6 to input graphs embedded in surfaces of arbitrary genus.

The following lemma is a direct consequence of Observation 2.

LEMMA 3.7. *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 $\mathcal{F}' = \bigcup_{F \in \mathcal{F}} \text{tpm}(F)$.*

It is easy to see that for every (planar) graph F , the set $\text{tpm}(F)$ contains a subcubic (planar) graph; see the paragraph after Observation 1. Combining this observation with Lemma 3.7 and Observation 1, Theorems 3.2, 3.4, and 3.6 follow directly from Theorems 3.1, 3.3, and 3.5, respectively. Therefore, the following sections are devoted to the proofs of Theorems 3.1, 3.3, and 3.5.

4. Boundaried graphs and their equivalence classes. Many of the following definitions were introduced in [9, 24] (see also [25, 32]).

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 $\lambda : R \rightarrow \mathbb{N}^+$ 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 boundaried 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 $\mathbf{B}_{\emptyset} = ((\emptyset, \{\emptyset\}), \emptyset, \emptyset)$ to denote the (unique) 0-boundaried *empty boundaried graph*.

Given a t -boundaried graph $\mathbf{G} = (G, R, \lambda)$, we define $\psi_{\mathbf{G}} : R \rightarrow [1, t]$ 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 $\sigma : V(\mathbf{G}_1) \rightarrow V(\mathbf{G}_2)$ that is an isomorphism $\sigma : V(G_1) \rightarrow V(G_2)$ 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) .

We extend the definition of \preceq_{tm} so that $\mathbf{G}_1 \preceq_{\text{tm}} \mathbf{G}_2$ denotes that \mathbf{G}_1 is a topological minor of \mathbf{G}_2 and, given a $\mathcal{X} \subseteq \mathcal{B}^{(t)}$, $\mathcal{X} \preceq_{\text{tm}} \mathbf{G}_2$ denotes that at least one of the t -boundaried graphs in \mathcal{X} is a topological minor of the t -boundaried graph \mathbf{G}_2 .

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 $\psi_{\mathbf{G}_1}^{-1}(i)$ and the vertex $\psi_{\mathbf{G}_2}^{-1}(i)$. 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 use the notation $\mathbf{G}|_I = (G, \lambda^{-1}(I), \lambda|_{\lambda^{-1}(I)})$, 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 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 $\lambda' : R \rightarrow I$ is a bijection and $\psi_{\mathbf{G}'} = \lambda$.

Equivalence relations. 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

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

It is easy to verify that $\equiv^{(\mathcal{F}, t)}$ is an equivalence relation. 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, among those with minimum number of edges, with minimum number of vertices (if there are more than one such graphs, pick one arbitrarily). Given a t -boundaried graph \mathbf{G} we denote by $\text{rep}_{\mathcal{F}}(\mathbf{G})$ 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, $\text{rep}_{\mathcal{F}}(\mathbf{B}) = \mathbf{B}$.

Note that if $\mathbf{B} = (B, R, \lambda)$ is a t -boundaried graph and $\mathcal{F} \preceq_{\text{tm}} B$, then $\text{rep}_{\mathcal{F}}(\mathbf{B})$ is, by definition, a consecutive t -boundaried graph whose underlying graph is a graph $H \in \mathcal{F}$ with minimum number of edges (and out of those, with minimum number of vertices), 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 to $\text{ext}_{\text{tm}}(\mathcal{F})$.

We need the following three lemmas. 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 4.1. *Let \mathcal{F} be a proper 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 and only if $\forall \mathbf{G} \in \mathcal{R}^{(\mathcal{F},t)} \mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{B}_1 \iff \mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{B}_2$.*

We will need the following simple lemma in the proof of Lemma 7.2.

LEMMA 4.2. *Let \mathcal{F} be a proper collection, let $t \in \mathbb{N}$, and let $\mathbf{B} = (B, R, \lambda) \in \mathcal{R}^{(\mathcal{F},t)}$. Then B contains at most $\text{size}(\mathcal{F})$ isolated vertices outside of R .*

Proof. If $\mathbf{B} = \mathbf{F}^{(\mathcal{F},t)}$, then by definition B is a graph in \mathcal{F} possibly completed by isolated vertices in R , so the lemma follows trivially in this case. So we may assume that $\mathcal{F} \not\preceq_{\text{tm}} B$, and suppose towards a contradiction, that B contains $\text{size}(\mathcal{F}) + 1$ isolated vertices in $V(B) \setminus R$. We define a boundaried graph $\mathbf{B}' = (B', R', \lambda') \in \mathcal{B}^{(t)}$ such that $R' = R$, $\lambda' = \lambda$, and B' is obtained from B by removing one of the isolated vertices in $V(B) \setminus R$. Note that $|E(B')| = |E(B)|$ and $|V(B')| < |V(B)|$. We claim that $\mathbf{B} \equiv^{(\mathcal{F},t)} \mathbf{B}'$, which contradicts the hypothesis that $\mathbf{B} \in \mathcal{R}^{(\mathcal{F},t)}$.

Indeed, consider an arbitrary $\mathbf{G} \in \mathcal{B}^{(t)}$. Suppose first that $\mathcal{F} \preceq_{\text{tm}} \mathbf{B}' \oplus \mathbf{G}$. Since B' is a subgraph of B , $\mathbf{B}' \oplus \mathbf{G}$ is a subgraph of $\mathbf{B} \oplus \mathbf{G}$, hence $\mathcal{F} \preceq_{\text{tm}} \mathbf{B} \oplus \mathbf{G}$ as well. Conversely, suppose that $H \preceq_{\text{tm}} \mathbf{B} \oplus \mathbf{G}$ for some graph $H \in \mathcal{F}$. Since all the graphs in \mathcal{F} , in particular H , have at most $\text{size}(\mathcal{F})$ vertices, any topological minor model of H in $\mathbf{B} \oplus \mathbf{G}$ uses at most $\text{size}(\mathcal{F})$ isolated vertices in $V(B) \setminus R$. Therefore, by possibly using another isolated vertex in $V(B') \setminus R$ instead of the removed one, $\mathbf{B}' \oplus \mathbf{G}$ also contains a topological minor model of H , hence $\mathcal{F} \preceq_{\text{tm}} \mathbf{B}' \oplus \mathbf{G}$. \square

Folios. Let \mathcal{F} be a proper collection. Given $t, r \in \mathbb{N}$, we define $\mathcal{A}_{\mathcal{F},r}^{(t)}$ 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 to $\text{ex}_{\text{tm}}(\mathcal{F})$. Note that a graph in $\mathcal{A}_{\mathcal{F},r}^{(t)}$ 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 $\text{folio}(\mathbf{B}, \mathcal{F}, r)$, as the set containing all boundaried graphs in $\mathcal{A}_{\mathcal{F},r}^{(t)}$ that are topological minors of \mathbf{B} . Moreover, in case $\mathcal{F} \preceq_{\text{tm}} \mathbf{B}$, we also include in $\text{folio}(\mathbf{B}, \mathcal{F}, r)$ the graph $\mathbf{F}^{(\mathcal{F},t)}$.

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

LEMMA 4.3. *Let $t \in \mathbb{N}$ and let \mathcal{F} be a proper collection. For every t -boundaried graph \mathbf{B} and every $r \in \mathbb{N}$, it holds that $|\text{folio}(\mathbf{B}, \mathcal{F}, r)| = 2^{\mathcal{O}_{r+d}(t \log t)}$, where $d = \text{size}(\mathcal{F})$. Moreover, $|\mathfrak{F}_{\mathcal{F},r}^{(t)}| = 2^{2^{\mathcal{O}_{r+d}(t \log t)}}$.*

Proof. Let $t \in \mathbb{N}$, let \mathcal{F} be a proper collection, let $r \in \mathbb{N}$, and let $n = t + r$. We prove a stronger result, namely that $|\mathcal{A}_{\mathcal{F},r}^{(t)}| = 2^{\mathcal{O}_{r+d}(t \log t)}$. The claimed bound on $|\mathfrak{F}_{\mathcal{F},r}^{(t)}|$ then follows directly by definition of the set $\mathfrak{F}_{\mathcal{F},r}^{(t)}$. By [35], there exists a constant c such that for each $G \in \text{ex}_{\text{tm}}(\mathcal{F})$, $|E(G)| \leq c \cdot |V(G)|$. By definition, every underlying graph of an element of $\mathcal{A}_{\mathcal{F},r}^{(t)}$ is in $\text{ex}_{\text{tm}}(\mathcal{F})$. If we want to construct an element $\mathbf{G} = (G, R, \lambda)$ of $\mathcal{A}_{\mathcal{F},r}^{(t)}$ 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 $\mathcal{A}_{\mathcal{F},r}^{(t)}$ 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. \square

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

LEMMA 4.4. *Let \mathcal{F} be a proper collection and let $d = \text{size}(\mathcal{F})$. Let also \mathbf{B}_1 and \mathbf{B}_2 be two t -boundaried graphs. If $\text{folio}(\mathbf{B}_1, \mathcal{F}, d) = \text{folio}(\mathbf{B}_2, \mathcal{F}, d)$, 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 $\text{folio}(\mathbf{B}_1, \mathcal{F}, d) = \text{folio}(\mathbf{B}_2, \mathcal{F}, d)$. We fix $\mathbf{G} \in \mathcal{B}^{(t)}$, and we need to prove that $\mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{B}_1$ if and only if $\mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{B}_2$.

Assume first that $\mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{B}_1$. 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 $\mathcal{A}_{\mathcal{F}, d}^{(t)}$, 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 $\text{folio}(\mathbf{B}_1, \mathcal{F}, d) = \text{folio}(\mathbf{B}_2, \mathcal{F}, d)$, 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 $\mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{B}_2$.

Conversely, assume that $\mathcal{F} \not\preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{B}_1$, 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 $\mathcal{F} \not\preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{B}_1$. \square

Lemmas 4.3 and 4.4 directly imply the following.

LEMMA 4.5. *There exists a function $h_1 : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ 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 $d = \text{size}(\mathcal{F})$. Moreover $h_1(d, t) = 2^{2^{\mathcal{O}_d(t \cdot \log t)}}$.*

5. 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 $\sigma : E(G) \cup \{R\} \rightarrow L(T)$ 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 $T \setminus \{e\}$ 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 to 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 $\lambda_e = \rho|_{R_e}$, i.e., ρ serves as a universal labeling of G that imposes a labeling of the vertices of all boundaried graphs that are obtained from subgraphs of G . 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 $\text{bw}(\mathbf{G})$, 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 [41], to boundaried graphs. Indeed, if G is a graph, then a *branch decomposition* of G is a branch decomposition of $(G, \emptyset, \emptyset)$. We also define the *branchwidth* of G as $\text{bw}(G) = \text{bw}(G, \emptyset, \emptyset)$.

LEMMA 5.1. *Let $\mathbf{G} = (G, R, \lambda)$ be a boundaried graph. Then $\text{bw}(\mathbf{G}) \leq \text{bw}(G) + |R|$.*

Proof. Let (T', σ') be a branch decomposition of $\mathbf{G}' = (G, \emptyset, \emptyset)$ 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 $\text{bw}(\mathbf{G}) \leq \text{bw}(G) + |R|$. \square

The following lemma is a combination of the single-exponential, linear-time, and constant-factor approximation of treewidth by Bodlaender et al. [8], with the fact that any graph G with $|E(G)| \geq 3$ satisfies that $\text{bw}(G) \leq \text{tw}(G) + 1 \leq \frac{3}{2}\text{bw}(G)$ [41]; 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) decomposition can be derived.

LEMMA 5.2. *There exists an algorithm that receives as input a graph G and a $w \in \mathbb{N}$ and either reports that $\text{bw}(G) > w$ 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.*

LEMMA 5.3. *There exists a function $\mu : \mathbb{N} \rightarrow \mathbb{N}$ such that for every planar subcubic collection \mathcal{F} , every graph in $\text{extm}(\mathcal{F})$ has branchwidth at most $y = \mu(d)$ where $d = \text{size}(\mathcal{F})$.*

Proof. Let $G \in \text{extm}(\mathcal{F})$ and let $F \in \mathcal{F}$ be a planar subcubic graph. Since F is subcubic and $F \not\leq_{\text{tm}} G$, it follows (see [18]) that $F \not\leq_m G$, and since F is planar this implies by [40] that $\text{tw}(G)$, hence $\text{bw}(G)$ as well, is bounded by a function depending only on F . \square

6. Proof of Theorem 3.1. We already have all the ingredients to prove Theorem 3.1.

Proof of Theorem 3.1. We provide a dynamic programming algorithm for the computation of $\text{tm}_{\mathcal{F}}(G)$ for the general case where \mathcal{F} is a proper collection. We first consider an, arbitrarily chosen, vertex labeling ρ of G . From Lemma 5.2, we may assume that we have a branch decomposition (T, σ) of $(G, \emptyset, \emptyset)$ of width $\mathcal{O}(w)$, where $w = \text{tw}(G)$. 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 $\mathbf{G}_{e_r} = (G, \emptyset, \emptyset)$. Keep also in mind that $t_e = \mathcal{O}(\text{tw}(G))$ 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 $\mathcal{C} \in \mathfrak{F}_{\mathcal{F}, d}^{(t'_e)}$ where $t'_e = t_e - |L|$. We also denote by \mathcal{P}_e the set of all e -pairs. Clearly, $|\mathcal{P}_e| = \sum_{i \in [0, t_e]} \binom{t_e}{i} \cdot |\mathfrak{F}_{\mathcal{F}, d}^{(t_e - i)}|$, and therefore, from Lemma 4.3, $|\mathcal{P}_e| = 2^{2^{\mathcal{O}_d(w \log w)}}$.

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

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

In the above definition, if such a set S does not exist, we set the value of the function to ∞ . Note that $\mathcal{P}_{e_r} = \{\emptyset\} \times \mathfrak{F}_{\mathcal{F}, d}^{(0)}$. Note also that the set $\mathcal{A}_{\mathcal{F}, d}^{(0)}$ contains only those graphs that do not contain some graph in \mathcal{F} as a topological minor. Therefore

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

Hence, our aim is to give a way to compute $\text{tm}_{\mathcal{F}}^{(e)}$ 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 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 $\sigma^{-1}(\ell) = \{x, y\}$. Clearly, $\mathbf{G}_{e_\ell} = (\{x, y\}, \{\{x, y\}\})$ and

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

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

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

For the dynamic programming algorithm, 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 $(L, \mathcal{C}) \in \mathcal{P}_e$ it is possible to verify that

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

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 $\mathcal{O}_d(|\mathcal{P}_{e_1}| \cdot |\mathcal{P}_{e_2}|) = 2^{2\mathcal{O}_d(w \log w)}$ 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 $|E(G)| = \mathcal{O}(\text{tw}(G) \cdot |V(G)|)$, the claimed running time follows. \square

7. Improved bounds when excluding a planar graph. We now prove the following result.

LEMMA 7.1. *Let $t \in \mathbb{N}$ and \mathcal{F} be a proper collection containing a subcubic planar graph, where $d = \text{size}(\mathcal{F})$, and let $\mathcal{R}^{(\mathcal{F}, t)}$ be a set of representatives for $\equiv^{(\mathcal{F}, t)}$. 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.*

Before we proceed with the proof of Lemma 7.1, we need a series of results. The proof of the following lemma uses ideas similar to the ones presented by Garnero et al. [25].

LEMMA 7.2. *There is a function $h_2 : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ such that if \mathcal{F} is a proper collection containing a subcubic planar graph, where $d = \text{size}(\mathcal{F})$, $t \in \mathbb{N}$, $\mathbf{B} = (B, R, \lambda) \in \mathcal{R}^{(\mathcal{F}, t)} \setminus \{\mathbf{F}^{(\mathcal{F}, t)}\}$, $z \in \mathbb{N}$, and X is a subset of $V(B)$ such that $X \cap R = \emptyset$ and $|N_B(X)| \leq z$, then $|X| \leq h_2(z, d)$.*

Proof. We set $h_2(z, d) = 2^{h_1(d, \mu(d)+z) \cdot (z + \mu(d)+1) + \zeta(\mu(d)+z)} + z + d$, where h_1 is the function of Lemma 4.5, μ is the function of Lemma 5.3, and $\zeta : \mathbb{N} \rightarrow \mathbb{N}$ is defined as $\zeta(x) = 2^{\binom{x}{2}}$. 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 $s = 2^q + z + d$. 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

$$(7.1) \quad B \in \text{extm}(\mathcal{F}).$$

We set $G = B[X \cup N_B(X)]$ and observe that $|V(G)| \geq |X| > s$. As G is a subgraph of B , (7.1) implies that

$$(7.2) \quad G \in \text{extm}(\mathcal{F}),$$

and therefore, from Lemma 5.3, $\text{bw}(G) \leq y$. Let $R' = N_B(X)$ and $\lambda' = \rho|_{R'}$. We set $\mathbf{G} = (G, R', \lambda')$. From Lemma 5.1, $\text{bw}(\mathbf{G}) \leq \text{bw}(G) + |R'| \leq y + |R'| = y + z$.

We now proceed to bound the number of isolated vertices of G . Note that every isolated vertex of G either belongs to R' or belongs to X and was already an isolated vertex in B . Since, by Lemma 4.2, B has at most d isolated vertices in $V(B) \setminus R$, and $X \cap B = \emptyset$, it follows that G has at most $|R'| + d$ isolated vertices.

Hence, since for any connected component C of G that is not an isolated vertex it holds that $|E(C)| \geq |V(C)|/2$, we conclude that

$$(7.3) \quad |E(G)| \geq \frac{|V(G)| - |R'| - d}{2} \geq \frac{|V(G)| - z - d}{2} > \frac{s - z - d}{2} = 2^{q-1}.$$

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

$$(7.4) \quad \forall e \in E(T), |R_e| \leq y + z.$$

We define $\mathcal{H} = \{\text{rep}_{\mathcal{F}}(\mathbf{G}_e) \mid e \in E(T)\}$. From (7.4), $\mathcal{H} \subseteq \bigcup_{i \in [0, y+z]} \mathcal{R}^{(\mathcal{F}, i)}$. From Lemma 4.5, $|\mathcal{H}| \leq (y+z+1) \cdot h_1(d, y+z)$, therefore $q \geq |\mathcal{H}| \cdot \zeta(y+z)$. Let r be the root of T and let P be a longest path in T that has r as an endpoint. As by (7.3), G 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 $\text{rep}_{\mathcal{F}}(\mathcal{S})$ 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

$$(7.5) \quad \mathbf{G}_{e_1} \equiv^{(\mathcal{F}, t)} \mathbf{G}_{e_2} \text{ and}$$

$$(7.6) \quad \mathbf{G}_{e_1} \sim \mathbf{G}_{e_2}.$$

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

$$(7.7) \quad |E(\mathbf{G}_{e_2})| < |E(\mathbf{G}_{e_1})|.$$

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 (7.6), implies that

$$(7.8) \quad \mathbf{B}^- \sim \mathbf{G}_{e_2}.$$

Let now $B^* = \mathbf{B}^- \oplus \mathbf{G}_{e_2}$. Combining (7.7) and (7.8), we may deduce that

$$(7.9) \quad |E(B^*)| < |E(B)|.$$

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

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 $\mathbf{B}^* = (B^-, R, \lambda)$, $D^+ = \mathbf{D} \oplus \mathbf{B}^*$, and $\mathbf{D}^+ = (D^+, R_{e_1}, \lambda_{e_1})$. Note that

$$(7.10) \quad \mathbf{D} \oplus \mathbf{B} = \mathbf{D}^+ \oplus \mathbf{G}_{e_1} \quad \text{and}$$

$$(7.11) \quad \mathbf{D} \oplus \mathbf{B}^* = \mathbf{D}^+ \oplus \mathbf{G}_{e_2}.$$

From (7.5), we have that $\mathcal{F} \preceq_{\text{tm}} \mathbf{D}^+ \oplus \mathbf{G}_{e_1} \iff \mathcal{F} \preceq_{\text{tm}} \mathbf{D}^+ \oplus \mathbf{G}_{e_2}$. This, together with (7.10) and (7.11), implies that $\mathcal{F} \preceq_{\text{tm}} \mathbf{D} \oplus \mathbf{B} \iff \mathcal{F} \preceq_{\text{tm}} \mathbf{D} \oplus \mathbf{B}^*$, 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 (7.9) contradict the fact that $\mathbf{B} \in \mathcal{R}^{(\mathcal{F}, t)}$. Therefore $|X| \leq s$, as required. \square

Given a graph G and an integer y , we say that a vertex set $S \subseteq V(G)$ is a *branchwidth- y -modulator* if $\text{bw}(G \setminus S) \leq y$. This notion is inspired from *treewidth-modulators*, which have been recently used in a series of papers (cf., for instance, [9, 25, 24, 32]).

The following proposition is a (weaker) restatement of [24, Lemma 3.10 of the full version] (see also [32]).

PROPOSITION 7.3. *There exists a function $f_2 : \mathbb{N}_{\geq 1} \times \mathbb{N} \rightarrow \mathbb{N}$ such that if $d \in \mathbb{N}_{\geq 1}$, $y \in \mathbb{N}$, and G is a graph such that $G \in \text{ex}_{\text{tm}}(K_d)$ 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 7.4. *There is a function $h_3 : \mathbb{N} \rightarrow \mathbb{N}$ such that if $t \in \mathbb{N}$ and \mathcal{F} is a proper collection containing a subcubic planar graph, where $d = \text{size}(\mathcal{F})$, then every graph in $\mathcal{R}^{(\mathcal{F}, t)}$ has at most $t \cdot h_3(d)$ vertices.*

Proof. We define $h_3 : \mathbb{N} \rightarrow \mathbb{N}$ so that $h_3(d) = 2 + h_2(f_2(d, \mu(d)), \mu(d))$ where h_2 is the function of Lemma 7.2, f_2 is the function of Proposition 7.3, and μ is the function of Lemma 5.3.

As $\mathcal{F}^{(\mathcal{F}, t)}$ has at most d vertices, we may assume that $\mathbf{G} = (G, R, \lambda) \in \mathcal{R}^{(\mathcal{F}, t)} \setminus \{\mathbf{F}^{(\mathcal{F}, t)}\}$. Note that $G \in \text{ex}_{\text{tm}}(\mathcal{F})$, therefore, from Lemma 5.3, $\text{bw}(G) \leq \mu(d)$. We set $y = \mu(d)$ and we observe that R is a *branchwidth- y -modulator* of G . Therefore, we can apply Proposition 7.3 on G and R and obtain a partition \mathcal{X} of $V(G)$ and an element $X_0 \in \mathcal{X}$ such that

$$(7.12) \quad R \subseteq X_0,$$

$$(7.13) \quad \max\{|X_0|, |\mathcal{X}| - 1\} \leq 2 \cdot |R|, \quad \text{and}$$

$$(7.14) \quad \forall X \in \mathcal{X} \setminus \{X_0\} : |N_G(X)| \leq f_2(d, y).$$

From (7.12) and (7.14), each $X \in \mathcal{X} \setminus \{X_0\}$ is a subset of $V(G)$ such that $X \cap R = \emptyset$ and $|N_G(X)| \leq f_2(d, y)$. Therefore, from Lemma 7.2, for each $X \in \mathcal{X} \setminus \{X_0\}$, $|X| \leq h_2(f_2(d, y), d)$. We obtain that

$$\begin{aligned} |G| &= |X_0| + \sum_{X \in \mathcal{X} \setminus \{X_0\}} |X| \\ &\stackrel{(7.13)}{\leq} 2 \cdot |R| + |R| \cdot h_2(f_2(d, y), d) \\ &= t \cdot (2 + h_2(f_2(d, y), d)) \\ &= t \cdot h_3(d), \end{aligned}$$

as required. \square

The next proposition follows from the results of Baste et al. [2] on the number of labeled graphs of bounded treewidth.

PROPOSITION 7.5 (Baste et al. [2]). *Let $n, q \in \mathbb{N}$. The number of labeled graphs with at most n vertices and branchwidth at most q is $2^{\mathcal{O}_q(n \cdot \log n)}$.*

We are now ready to prove Lemma 7.1.

Proof of Lemma 7.1. Before we proceed to the proof we need one more definition. Given $n \in \mathbb{N}$, we set $\mathcal{B}_{\leq n}^{(\mathcal{F}, t)} = \mathcal{A}_{\mathcal{F}, n-t}^{(t)} \cup \{\mathbf{F}^{(\mathcal{F}, t)}\}$.

Note that, from Lemma 7.4, $\mathcal{R}^{(\mathcal{F}, t)} \subseteq \mathcal{B}_{\leq n}^{(\mathcal{F}, t)}$, where $n = t \cdot h_3(d)$. Also, from Lemma 5.3, all graphs in $\mathcal{B}_{\leq n}^{(\mathcal{F}, t)}$ have branchwidth at most $y = \max\{\mu(d), t\}$. The fact that $|\mathcal{B}_{\leq n}^{(\mathcal{F}, t)}| = 2^{\mathcal{O}_d(t \cdot \log t)}$ follows easily by applying Proposition 7.5 for n and q .

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 $\mathcal{B}_{\leq n}^{(\mathcal{F}, t)}$ 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 $\mathcal{B}_{\leq n}^{(\mathcal{F}, t)}$ is based on the fact that, given two t -boundaried graphs \mathbf{B}_1 and \mathbf{B}_2 , $\mathbf{B}_1 \equiv^{(\mathcal{F}, t)} \mathbf{B}_2$ if and only if, for every $\mathbf{G} \in \mathcal{B}_{\leq n}^{(\mathcal{F}, t)}$, $\mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{B}_1 \iff \mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{B}_2$. This fact follows directly from Lemma 4.1 and taking into account that $\mathcal{R}^{(\mathcal{F}, t)} \subseteq \mathcal{B}_{\leq n}^{(\mathcal{F}, t)}$.

Note that it takes $|\mathcal{B}_{\leq n}^{(\mathcal{F}, t)}|^3 \cdot \mathcal{O}_d(1) \cdot t^{\mathcal{O}(1)}$ steps to construct \mathcal{Q} , by using the topological minor containment algorithm of Grohe et al. [27]. As $|\mathcal{B}_{\leq n}^{(\mathcal{F}, t)}| = 2^{\mathcal{O}_d(t \cdot \log t)}$, the construction of \mathcal{Q} , and therefore of $\mathcal{R}^{(\mathcal{F}, t)}$ as well, can be done in the claimed number of steps. \square

8. Proof of Theorem 3.3. We are now ready to prove Theorem 3.3. The main difference with respect to the proof of Theorem 3.1 is an improvement on the size of the tables of the dynamic programming algorithm, namely $|\mathcal{P}_e|$, where the fact that the collection \mathcal{F} contains a planar subcubic graph is exploited in order to bound the treewidth.

Proof of Theorem 3.3. We provide a dynamic programming algorithm for the computation of $\text{tm}_{\mathcal{F}}(G)$. We first consider an, arbitrarily chosen, vertex labeling ρ of G . From Lemma 5.2, we may assume that we have a branch decomposition (T, σ) of $(G, \emptyset, \emptyset)$ of width at most $w = \mathcal{O}(\text{bw}(G)) = \mathcal{O}(\text{tw}(G))$. 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 $\mathbf{G}_{e_r} = (G, \emptyset, \emptyset)$. Keep also in mind that $t_e = \mathcal{O}(\text{tw}(G))$ for every $e \in E(T)$.

Our next step is to define the tables of the dynamic programming algorithm. For a positive integer t , we let $\overline{\mathcal{R}}^{(\mathcal{F}, t)} = \mathcal{R}^{(\mathcal{F}, t)} \setminus \{\mathbf{F}^{(\mathcal{F}, t)}\}$. Let $e \in E(T)$. We call the pair (L, \mathbf{B}) an e -pair if

1. $L \subseteq R_e$, and
2. $\mathbf{B} = (B, R, \lambda) \in \overline{\mathcal{R}}^{(\mathcal{F}, k')}$ 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

$$\begin{aligned} |\mathcal{P}_e| &= \sum_{i \in [0, t_e]} \binom{t_e}{i} \cdot |\overline{\mathcal{R}}^{(\mathcal{F}, t_e - i)}| \\ &= (t_e + 1) \cdot 2^{t_e} \cdot 2^{\mathcal{O}_d(t_e \cdot \log t_e)} \quad (\text{from Lemma 7.1}) \\ &= 2^{\mathcal{O}_d(w \cdot \log w)}. \end{aligned}$$

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

$$\mathbf{tm}_{\mathcal{F}}^{(e)}(L, \mathbf{B}) = \min\{|S| \mid S \subseteq V(G_e) \wedge L = R_e \cap S \wedge \mathbf{B} = \text{rep}_{\mathcal{F}}(\mathbf{G}_e \setminus S)\},$$

where the value ‘ ∞ ’ is assigned when such a set S does not exist.

Note that $\mathcal{P}_{e_r} = \{(\emptyset, \mathbf{B}) \mid \mathbf{B} \in \overline{\mathcal{R}}^{(\mathcal{F}, 0)}\}$, and therefore

$$\begin{aligned} \mathbf{tm}_{\mathcal{F}}(G) &= \min\{|S| \mid S \subseteq V(G) \wedge \mathcal{F} \not\prec_{\text{tm}} G \setminus S\} \quad (\text{from Equation (2.1)}) \\ &= \min\{|S| \mid S \subseteq V(G_{e_r}) \wedge \emptyset = R_{e_r} \cap S \wedge \mathcal{F} \not\prec_{\text{tm}} G_{e_r} \setminus S\} \\ &= \min_{\mathbf{B} \in \overline{\mathcal{R}}^{(\mathcal{F}, 0)}} \{\min\{|S| \mid S \subseteq V(G_{e_r}) \wedge \emptyset = R_{e_r} \cap S \wedge \mathbf{B} = \text{rep}_{\mathcal{F}}((G_{e_r} \setminus S, \emptyset, \emptyset))\} \\ &= \min_{\mathbf{B} \in \overline{\mathcal{R}}^{(\mathcal{F}, 0)}} \{\mathbf{tm}_{\mathcal{F}}^{(e_r)}(\emptyset, \mathbf{B})\}. \end{aligned}$$

Therefore, in order to compute $\mathbf{tm}_{\mathcal{F}}(G)$, it is enough to compute $\mathbf{tm}_{\mathcal{F}}^{(e)}$ for every $e \in E(T)$. Note that, by Lemma 7.1, we may assume that we have at hand the set $\mathcal{R}^{(\mathcal{F}, t)}$ of representatives for every $t \leq w$. 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 $\sigma^{-1}(\ell) = \{x, y\}$. Clearly, $\mathbf{G}_{e_\ell} = (\{x, y\}, \{\{x, y\}\})$ and

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

As the size of the elements in \mathcal{P}_{e_ℓ} depends only on \mathcal{F} , it is possible to compute $\mathbf{tm}_{\mathcal{F}}^{(e_\ell)}$ 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 $F_e = (R_{e_1} \cup R_{e_2}) \setminus R_e$. For the dynamic programming algorithm, 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 is possible to verify that

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

Note that given $\mathbf{tm}_{\mathcal{F}}^{(e_i)}$, $i \in [1, 2]$ and a pair $(L, \mathbf{B}) \in \mathcal{P}_e$, the value of $\mathbf{tm}_{\mathcal{F}}^{(e)}(L, \mathbf{B})$ can be computed by the above formula in $\mathcal{O}_d(|\mathcal{P}_{e_1}| \cdot |\mathcal{P}_{e_2}|) = 2^{\mathcal{O}_d(w \cdot \log w)}$ steps. Indeed,

by Lemma 7.4, the representatives in the sets $\mathcal{R}^{(\mathcal{F}, t')}$, for $t' \in [0, t]$, have size $\mathcal{O}_d(w)$, and therefore in the above equation, the boundaried graph $((\mathbf{B}_1 \diamond Z_1) \odot (\mathbf{B}_2 \diamond Z_2))|_Z$ has size at most twice the maximum size of a representative, hence $\mathcal{O}_d(w)$ as well. Hence, $\text{rep}_{\mathcal{F}}\left(\left((\mathbf{B}_1 \diamond Z_1) \odot (\mathbf{B}_2 \diamond Z_2)\right)|_Z\right)$ can be computed by creating a repository (in a preprocessing phase) containing the representative of every graph on at most $\mathcal{O}_d(w)$ vertices, which can be done, by Lemma 4.1, in time $2^{\mathcal{O}_d(w \cdot \log w)}$ by using the set $\mathcal{R}^{(\mathcal{F}, t)}$ provided by Lemma 7.1.

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 algorithm 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 $|E(G)| = \mathcal{O}(\text{bw}(G) \cdot |V(G)|)$, the claimed running time follows. \square

We conclude this section by observing that the dynamic programming algorithm presented in the proof of Theorem 3.3 is robust, in the sense that it does not explicitly use the fact that the equivalence relation $\equiv^{(\mathcal{F}, t)}$ is defined for the particular *topological minor* containment relation. Indeed, all that the algorithm uses about $\equiv^{(\mathcal{F}, t)}$ is that the size of any representative is $\mathcal{O}_d(t)$ (Lemma 7.4) and this allows to conclude that the number of equivalence classes is $2^{\mathcal{O}_d(t \cdot \log t)}$ (Lemma 7.1), where $d = \text{size}(\mathcal{F})$. Moreover, as discussed at the end of the proof of Theorem 3.3, given a graph of size at most twice the maximum size of a representative, in order to perform the join operation we need to compute (or even precompute) its representative within the claimed asymptotic running time.

Hence, as far as these conditions are fulfilled, the algorithm of Theorem 3.3 could also be applied for the \mathcal{F} - \mathfrak{C} -DELETION problem where \mathfrak{C} is an arbitrary containment relation, and the equivalence relation $\equiv_{\mathfrak{C}}^{(\mathcal{F}, t)}$ is defined according to \mathfrak{C} . This discussion is summarized in the following theorem, which we state here for further reference.

THEOREM 8.1. *Let \mathfrak{C} be a graph containment relation, let \mathcal{F} be a proper collection of graphs, let $d = \text{size}(\mathcal{F})$, let t be a positive integer, and let $\equiv_{\mathfrak{C}}^{(\mathcal{F}, t)}$ be the equivalence relation on t -boundaried graphs defined by \mathfrak{C} . Suppose that there exist three functions $f_s, f_n, f_r : \mathbb{N}^2 \rightarrow \mathbb{N}$ such that*

- *any minimum-sized representative of $\equiv_{\mathfrak{C}}^{(\mathcal{F}, t)}$ has size at most $f_s(d, t)$,*
- *the number of equivalence classes of $\equiv_{\mathfrak{C}}^{(\mathcal{F}, t)}$ is at most $f_n(d, t)$, and*
- *given a t -boundaried graph of size at most $2f_s(d, t)$, it is possible to compute a minimum-size representative of its equivalence class of $\equiv_{\mathfrak{C}}^{(\mathcal{F}, t)}$ in time $f_r(d, t)$.*

Then the \mathcal{F} - \mathfrak{C} -DELETION problem can be solved in time $\max\{f_n(d, t), f_r(d, t)\}^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(t)} \cdot n$ on n -vertex graphs of treewidth at most t .

Note that Theorem 3.3 is a particular case of Theorem 8.1 with \mathfrak{C} being the topological minor containment relation, \mathcal{F} containing a planar graph, $f_s(d, t) = \mathcal{O}_d(t)$, and $f_n(d, t) = f_c(d, t) = 2^{\mathcal{O}_d(t \cdot \log t)}$.

9. Single-exponential algorithm for planar graphs. The purpose of this section is to prove Theorem 3.5. We do this by proving that, on planar graphs, our dynamic programming approach can be easily modified in order to yield an algorithm of single-exponential dependence on tw . Later, in Section 10 we extend the result of this section to graphs of bounded genus.

We start with some definitions about planar graphs. We denote the sphere by Σ_0 , i.e., $\Sigma_0 = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1\}$. We say that a graph G is Σ_0 -embedded if it is embedded in Σ_0 without edge crossings. More generally, given a disk Δ (that is a subset of Σ_0 homeomorphic to $\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$), a Δ -embedded graph is

a graph that is embedded in Δ without edge crossings and without edges intersecting its boundary. To simplify notation, when we consider a Σ_0 -embedded graph or a Δ -embedded graph, we treat it both as the pair $(V(G), E(G))$ and as the set of its points in the embedding.

An *O-arc* of Σ_0 is a subset of Σ_0 that is homeomorphic to the circle $\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}$ and an *I-arc* is a subset of Σ_0 that is homeomorphic to the open interval $(0, 1)$. Given a Σ_0 -embedded graph G and a set $\Lambda \subseteq \Sigma_0$, we say that Λ is *G-normal* if $\Lambda \cap G \subseteq V(G)$. If an *O-arc* is *G-normal*, then we call it a *noose* of G . Two nooses N_1 and N_2 of G are *confluent* if one of the two disks bounded by N_1 contains N_2 . The *length* of a noose N is the number of vertices it meets, i.e., $|N| = |V(G) \cap N|$.

Sphere-cut decompositions. Let $\mathbf{G} = (G, R, \lambda) \in \mathcal{B}^{(t)}$, for some $t \in \mathbb{N}$ and let Δ be a disk of Σ_0 . We say that \mathbf{G} is a *Δ -embedded t -boundaried graph* if G is a Δ -embedded graph and $G \cap \mathbf{bd}(\Delta) = R$, where we use $\mathbf{bd}(\Delta)$ to denote the boundary of Δ .

A branch-decomposition (T, σ) of a Δ -embedded boundaried graph \mathbf{G} with root $r \in V(T)$ is called a *sphere-cut decomposition* (or, in short, *sc-decomposition*), if for every edge e of T , there exists a noose N_e of G , such that⁴:

- The nooses in $\{N_e \mid e \in E(T)\}$ are pairwise confluent,
- $N_{e_r} = \mathbf{bd}(\Delta)$,
- for every $e \in E(T)$, it holds that $G_e = \Delta_e \cap G$, where Δ_e is the disk bounded by N_e such that $\Delta_e \subseteq \Delta_{e_r}$,

Notice that for every $e \in E(T)$, \mathbf{G}_e can be seen as a Δ_e -embedded t_e -boundaried graph.

The following proposition has been proved in [47, 19] (for the running time, see [28]).

PROPOSITION 9.1. *Let G be a 2-connected Σ -embedded planar graph such that $|E(G)| \geq 2$. There exists a sphere-cut decomposition of G of width equal to $\mathbf{bw}(G)$. Moreover, a sphere-cut decomposition of G of optimal width can be computed in $\mathcal{O}(n^3)$ steps.*

Oriented disk-embedded boundaried graphs. An *oriented disk-embedded t -boundaried graph* \mathbf{G} (in short, *ode- t -boundaried graph*) is a quadruple $\mathbf{G} = (G, \Delta, v, \mathbf{s})$ where Δ is a disk, G is a Δ -embedded graph, $|G \cap \mathbf{bd}(\Delta)| = t$, $v \in G \cap \mathbf{bd}(\Delta)$, and $\mathbf{s} \in \{+, -\}$. We call \mathbf{s} *orientation* of \mathbf{G} . We say that the set $G \cap \mathbf{bd}(\Delta)$, denoted by $R(\mathbf{G})$, is the *boundary* of \mathbf{G} . We use $\mathcal{E}_o^{(t)}$ to denote all ode- t -boundaried graphs. Given a $\mathbf{G} \in \mathcal{E}_o^{(t)}$, we define $\overline{\mathbf{G}} = (G, R, \lambda)$ where $R = R(\mathbf{G})$, $\lambda(v) = 1$, and the rest of the vertices of R are indexed by consecutive numbers in $\{2, \dots, |R|\}$, following their order on $\mathbf{bd}(\Delta)$ by keeping the interior of Δ on the right or on the left depending on whether $\mathbf{s} = +$ or $\mathbf{s} = -$. Clearly, $\overline{\mathbf{G}}$ is a consecutive t -boundaried graph.

Notice that, given a sphere-cut decomposition (T, σ) of a Δ -embedded boundaried graph \mathbf{G} , for every $e \in E(T)$, $v \in R_e$, and $\mathbf{s} \in \{+, -\}$, the quadruple $(G_e, \Delta_e, v, \mathbf{s})$ is an ode- t -boundaried graph. We will use those ode- t -boundaried graphs for our dynamic programming algorithm on sphere-cut decompositions. More precisely, the tables of the dynamic programming will consist of this type of objects, and note that, thanks to the topological structure of the separators in a sc-decomposition, they carry all the information that we need in order to glue them together, namely, Δ_e , v , and \mathbf{s} .

⁴This definition is slightly different from the original definition given in [47, 19], but it can be easily seen that they are equivalent. Also, note that we consider \mathbf{G} to be Δ -embedded instead of Σ_0 -embedded; this will be useful to preserve the recursive properties given by the separators of a sc-decomposition.

We will prove in Lemma 9.3 that their number is single-exponential and this will yield the running time of Theorem 3.5. To this end, we need to adapt the gluing operation for ode- t -boundaried graphs and, based on this, define a suitable equivalence relation from which the tables will be constructed.

Let $\mathbf{G}_i = (G_i, \Delta_i, v_i, \mathbf{s}_i), i \in [2]$ be two graphs in $\mathcal{E}_\circ^{(t)}$. Notice that the graph $\overline{\mathbf{G}}_1 \oplus \overline{\mathbf{G}}_2$ defines a Σ_0 -embedded graph: just take the union of the points of the embeddings of G_1 and G_2 by identifying same index vertices (in case $\mathbf{s}_1 = \mathbf{s}_2$, we redraw one of the graphs by reflecting it outside its disk). We extend \oplus by defining $\mathbf{G}_1 \oplus \mathbf{G}_2 = \overline{\mathbf{G}}_1 \oplus \overline{\mathbf{G}}_2$.

We define an equivalence relation $\equiv_\circ^{(\mathcal{F}, t)}$ on $\mathcal{E}_\circ^{(t)}$ as follows: Given $\mathbf{G}_1, \mathbf{G}_2 \in \mathcal{E}_\circ^{(t)}$, we write $\mathbf{G}_1 \equiv_\circ^{(\mathcal{F}, t)} \mathbf{G}_2$ if

$$\forall \mathbf{G} \in \mathcal{E}_\circ^{(t)} \mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{G}_1 \iff \mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{G}_2.$$

Notice that we can see each ode- t -boundaried graph $\mathbf{G} = (G, \Delta, v, \mathbf{s})$ as an embedding in Δ of the boundaried graph $\overline{\mathbf{G}}$. This implies that, if we ignore the way combinatorial graphs are embedded, $\equiv_\circ^{(\mathcal{F}, t)}$ induces a coarsening of the restriction of $\equiv^{(\mathcal{F}, t)}$ to the combinatorial boundaried graphs in $\mathcal{E}_\circ^{(t)}$. As for every combinatorial planar graph there is a finite number of ways to embed it in Σ_0 (up to topological isomorphism), Lemma 4.5 can also bound the number of equivalence classes of $\equiv_\circ^{(\mathcal{F}, t)}$. As we did for $\equiv^{(\mathcal{F}, t)}$, we set up a set $\mathcal{R}_\circ^{(\mathcal{F}, t)}$ of representatives of $\equiv_\circ^{(\mathcal{F}, t)}$, defined analogously.

LEMMA 9.2. *For every $t \in \mathbb{N}$, the number of different ode- t -boundaried graphs in $\mathcal{E}_\circ^{(t)}$ with at most $\mathcal{O}(t)$ vertices is $2^{\mathcal{O}(t)}$.*

Proof. The proof uses the well-known fact that there are $2^{\mathcal{O}(n)}$ Σ_0 -embedded graphs on n vertices (up to topological isomorphism). This can be easily verified as follows. As a consequence of the results of Tutte [48] the number of planar triangulations on n vertices is $2^{\mathcal{O}(n)}$ (see also [11] for more refined bounds). Since planar triangulations are 3-connected graphs, by Whitney's theorem [49], they have a unique embedding in Σ_0 . Taking into account that every Σ_0 -embedded graph can be completed into a Σ_0 -embedded triangulation by adding at most $3n - 6$ edges, it follows that the number of Σ_0 -embedded graphs on n vertices is at most $(256/27)^{n + \mathcal{O}(\log n)} \cdot 2^{3n-6} = 2^{\mathcal{O}(n)}$.

Given a Δ -embedded graph G and a set $F \subseteq E(G)$, we say that the pair (G, F) is a *partially edge-annotated Δ -embedded graph*. We extend the concept of topological isomorphism to partially edge-annotated Δ -embedded graphs by demanding annotated edges to be mapped to annotated edges and non-annotated edges to be mapped to non-annotated edges. Combining this with the above upper bound for the number of Σ_0 -embedded graphs and the fact that there are at most $2^{3t-6} = 2^{\mathcal{O}(t)}$ ways to choose F , we conclude that there are at most $2^{\mathcal{O}(t)}$ different partially edge-annotated Δ -embedded graphs.

We correspond each tuple $(G, \Delta, v, \mathbf{s}) \in \mathcal{E}_\circ^{(t)}$ to the partially edge-annotated Δ -embedded graph (\hat{G}, F) where F are the edges corresponding to the G -normal I -arcs of the set $\mathbf{bd}(\Delta) \setminus V(G)$ and $\hat{G} = (V(G), E(G) \cup F)$ (notice that some of the edges in F may already be edges of G). Clearly \hat{G} is also a Δ -embedded graph. Therefore, the number of different elements in $\mathcal{E}_\circ^{(t)}$ with at most $\mathcal{O}(t)$ vertices is bounded by the number of different partially edge-annotated Δ -embedded graphs (that is $2^{\mathcal{O}(t)}$, as shown above), multiplied by the k possible choices of $v \in R$ and the 2 possible choices of $\mathbf{s} \in \{+, -\}$. The lemma follows. \square

LEMMA 9.3. *Let $t \in \mathbb{N}$ and \mathcal{F} be a proper collection containing a subcubic planar graph, where $d = \text{size}(\mathcal{F})$, and let $\mathcal{R}_\circ^{(\mathcal{F}, t)}$ be a set of representatives for $\equiv_\circ^{(\mathcal{F}, t)}$. Then*

$|\mathcal{R}_\circ^{(\mathcal{F},t)}| = 2^{\mathcal{O}_d(t)}$. Moreover, there exists an algorithm that given \mathcal{F} and t , constructs a set of representatives $\mathcal{R}_\circ^{(\mathcal{F},t)}$ in $2^{\mathcal{O}_d(t)}$ steps.

Proof. Observe first that Lemma 7.2 holds for $\mathcal{R}_\circ^{(\mathcal{F},t)}$, instead of $\mathcal{R}^{(\mathcal{F},t)}$, with essentially the same proof. Indeed, Lemmas 4.2, 5.3, and 5.1 are not affected by this change and, as we argued before, there is an analogue of Lemma 4.5 bounding the size of $\mathcal{R}_\circ^{(\mathcal{F},t)}$. The only difference is that now we have that each boundaried graph, say $\mathbf{X} \in \mathcal{B}^{(t)}$, that appears in the proof is now a member of $\mathcal{E}_\circ^{(t)}$ and therefore, during the gluing operations, we replace it by $\bar{\mathbf{X}}$ that is the boundaried graph corresponding to \mathbf{X} .

Based on the above observation, Lemma 7.4 also holds for $\mathcal{R}_\circ^{(\mathcal{F},t)}$ instead of $\mathcal{R}^{(\mathcal{F},t)}$, as it uses the analogue of Lemma 7.2 as well as Lemma 5.3 and Proposition 7.3 (that is also not affected by the change). Now combining this version of Lemma 7.4 with Lemma 9.2, we conclude that $|\mathcal{R}_\circ^{(\mathcal{F},t)}| = 2^{\mathcal{O}_d(t)}$. The algorithm claimed in the second statement of the lemma is the same as the corresponding one in the proof of Lemma 7.1, with the only difference that now, instead of enumerating t -boundaried graphs, we enumerate $\text{ode-}t$ -boundaried graphs that are $2^{\mathcal{O}_d(t)}$ many by Lemma 9.2. \square

We are now ready to present the proof of Theorem 3.5.

Proof of Theorem 3.5. Let (G', k) be an instance of \mathcal{F} -TM-DELETION. First of all, we consider a tree decomposition (\mathcal{X}, T') of G' whose bags are its blocks and each block is seen as a 1-boundaried graph whose boundary is some cut-vertex of G (this orientation of the blocks can be done by arbitrarily rooting the tree T' of the tree decomposition). Given this, we can process each one of the blocks by doing conventional dynamic programming in order to join them, using as tables those of the algorithm given in the proof of Theorem 3.1. This processing will cost $\mathcal{O}_d(1)$ steps per block, therefore $\mathcal{O}_d(n)$ steps in total. In what follows, we explain how to process each one of the blocks.

We consider a block $\mathbf{G} = (G, R, \lambda)$ of G' where $|R| \leq 1$ (if G is not the root block of (T', \mathcal{X}) , then $|R| = 1$, otherwise $|R| = 0$). As each block is 2-connected, we can construct, using the algorithm of Proposition 9.1, an optimal sphere-cut decomposition (T, σ) of \mathbf{G} in $\mathcal{O}(|V(G)|^3)$ steps. Notice that for each $e \in E(T)$, the graph G can be seen as the graph $\mathbf{G}_e \oplus \mathbf{G}'_e$ where $\mathbf{G}_e = (G_e, \Delta_e, v, +)$ and $\mathbf{G}'_e = (G'_e, \Delta_e, v, -)$, where $G'_e = G \cap (\Sigma_0 \setminus \text{int}(\Delta_r))$ and v is the minimum index vertex in R_e , where $\text{int}(\Delta_r)$ denotes the interior of the disk Δ_r . Given these conventions, the dynamic programming is the same as the one in the proof of Theorem 3.3. The only difference is that now we use $\mathcal{R}_\circ^{(\mathcal{F},t)}$ instead of $\mathcal{R}^{(\mathcal{F},t)}$. This latter change implies, because of Lemma 9.3, that the tables of the dynamic programming for the edge $e \in E(T)$ have $2^{\mathcal{O}_d(t_e)}$ entries. Therefore the algorithm runs in $2^{\mathcal{O}_d(\text{tw})} \cdot n$ steps in total. \square

10. Single-exponential algorithm for graphs embedded in surfaces.

In this section, we give a concise description of how Theorem 3.5 can be extended to graphs embedded in surfaces. The idea is again to treat the graphs processed by the dynamic programming of the proof of Theorem 3.3 as embedded graphs. However, when it comes to embeddings in surfaces, this is more technical to describe formally. For this, we need to provide some more definitions.

Surfaces are seen as connected compact 2-manifolds without boundaries. It is known (see e.g., [38]) that any surface Σ can be obtained, up to homeomorphism, by adding $\mathbf{eg}(\Sigma)$ crosscaps to the sphere, where $\mathbf{eg}(\Sigma)$ is called the *Euler genus* of Σ . We say that a graph G is Σ_g -embedded if it is embedded without crossings in a surface

whose Euler genus is g . Theorems 3.5 and 3.6 can be extended as follows.

THEOREM 10.1. *If \mathcal{F} is a proper collection containing a subcubic planar graph, where $d = \text{size}(\mathcal{F})$, then there exists an algorithm that solves \mathcal{F} -TM-DELETION on Σ_g -embedded graphs in $2^{\mathcal{O}_{d+g}(\text{tw})} \cdot n^3$ steps.*

Proof. The proof is an adaptation of the proof of Theorem 3.5 to more general surfaces, using some of the tools developed in [44]. A *noose* of a Σ_g -embedded graph G is any G -normal O -arc of Σ . We say that a noose N is *non-contractible* if none of the connected components of $G \setminus N$ is homeomorphic to the interior of a disk. We say that a Σ_g -embedded graph is *polyhedrally Σ_g -embedded* if it is 3-connected and every non-contractible noose of G has size at least 3. Given a graph G , a *polyhedral decomposition* of a Σ_g -embedded graph G is a triple (A, \mathcal{X}, T) where $A \subseteq V(G)$, $|A| = \mathcal{O}(g)$, and (\mathcal{X}, T) is a tree decomposition of $G^A := G \setminus A$ where

- the adhesion of (\mathcal{X}, T) is at most 2,
- for every $t \in V(T)$, the torso of X_i is polyhedrally Σ_{g_i} -embedded, for some $g_i \leq g$.

In the above definition, the *adhesion* of (\mathcal{X}, T) is the maximum size of the intersection between two bags of (\mathcal{X}, T) . Moreover, the *torso* of a bag X_i is the graph obtained by $G[X_i]$ if we make adjacent each pair of vertices that both belong to the intersection of X_i with some other bag. We also call A *apex set* of G . We remark that, during the course of this proof, the presence of A in the notation G^A should be interpreted as the “absence” of A from some more general graph G .

According to [44], given a Σ_g -embedded graph G , a polyhedral decomposition of G can be constructed in $\mathcal{O}(n^3)$ steps. We now consider such a polyhedral decomposition of G , we root it at the union of the apex set A and the intersection of its bag with its parent bag in T and, as we did the beginning of the proof of Theorem 3.5, we process the bags of (\mathcal{X}, T) in a bottom-up fashion. The only difference is that now, given that we have a way to process each of the the “almost polyhedral” bags, we do conventional dynamic programming, as in Theorem 3.1, for tables that have $\mathcal{O}(g)$ instead of at most 1 vertices. This reduces the proof to the case where the input graph, we call it G , contains a set A of $\mathcal{O}(g)$ apex vertices such that are $G^A = G \setminus A$ is a polyhedrally $\Sigma_{g'}$ -embedded graph for some $g' \leq g$. We also see G^A as a Σ_g -embedded t -boundaried graph \mathbf{G}^A whose boundary has size at most 2. To deal with this, we will use an analogue of sphere-cut decompositions for graphs that are polyhedrally embedded in higher-genus surfaces, called *surface-cut decomposition*, introduced in [44]. This requires some definitions in order to extend the concept of Δ -embedded t -boundaried graph to surfaces.

Given a Σ_g -embedded graph G , any G -normal O -arc of Σ_g is called *noose* N of G . We say that a collection $\mathcal{N} = \{N_1, \dots, N_s\}$ of O -arcs in Σ_g is a *surface-cut* if they are pairwise confluent, $\bigcup_{1 \leq i < j \leq s} N_i \cap N_j$ is finite, and $\Sigma_g \setminus \bigcup \mathcal{N}$ has exactly two connected components. We use the notation $C(\mathcal{N}) = \bigcup \mathcal{N}$. We say that a subset Ψ of Σ_g is a *g -disk* if it is the closure of one of the connected components of $\Sigma_g \setminus C(\mathcal{N})$ for some surface-cut collection \mathcal{N} of Σ_g . Notice that there are $\mathcal{O}_g(1)$ different g -disks up to homeomorphism. We say that two g -disks Ψ_1 and Ψ_2 are *g -complementary*, if $\Psi_1 \cup \Psi_2 = \Sigma_g$ and $\Psi_1 \cap \Psi_2$ is the boundary of both Ψ_1 and Ψ_2 . We also refer to the g -disks Ψ_1 and Ψ_2 as *the two disks bounded by \mathcal{N}* . Given a g -disk Ψ , we say that a graph G is *Ψ -embedded* if G is embedded in Ψ without edge crossings and without edges intersecting its boundary. Notice that we can see $\mathbf{bd}(\Psi)$, i.e., the boundary of Ψ , as a graph if we fix the vertex set to be $R(G) := V(G) \cap \mathbf{bd}(\Psi)$ (called *frontier*

vertices); we call this graph *frontier* of the Ψ -embedded graph G and we denote it by F_G . We also denote by F_G^A the graph obtained by F_G after taking its disjoint union with some set A of isolated vertices (called *apex* vertices). Given that $a = |A|$, we call F_G^A *a-enhanced frontier graph* of the Ψ -embedded graph G . We denote by $\mathcal{L}_{\Psi,a}^{(t)}$ the set obtained taking all *a-enhanced frontier graphs* of all Ψ -embedded graphs with $t - a$ vertices in their boundary, and renaming their vertices by numbers in $[t]$. Moreover, we update $\mathcal{L}_{\Psi,a}^{(t)}$ by repetitively removing one graph from any pair of isomorphic graphs in this set, as long as such a pair exists. Notice that after this relaxation, all graphs in $\mathcal{L}_{\Psi,a}^{(t)}$ are pairwise non-isomorphic. Moreover, it is easy to prove that if $a = \mathcal{O}(g)$, then $|\mathcal{L}_{\Psi,a}^{(t)}| = 2^{\mathcal{O}_g(t)}$. Intuitively, the set $\mathcal{L}_{\Psi,a}^{(t)}$ encodes all permissible ways the boundaries of two t -boundaried graphs can be identified so that such a gluing will result in a Σ_g -embeddable graph, and its cardinality corresponds to the number of automorphisms of every frontier graph F_G enhanced with a additional apex vertices. Later, we will demand that, while gluing boundaried graphs, frontier vertices are identified to frontier vertices and apex vertices are identified to apex vertices.

Given a g -disk Ψ , a t -boundaried graph $\mathbf{G} = (G, R, \lambda)$ and a set $A \subseteq V(G)$, we say that (\mathbf{G}, A) is an *a-almost Ψ -embedded t -boundaried graph* if $A \subseteq R$, $a = |A|$, $G^A := G \setminus A$ is a Ψ -embedded graph, $G^A \cap \mathbf{bd}(\Delta) = R \setminus A$, and λ is an isomorphism from F_G^A to the unique graph in $\mathcal{L}_{\Psi,a}^{(t)}$ that is isomorphic to F_G^A by an isomorphism that maps A to apex vertices. We denote by $\mathcal{E}_{\Psi,a}^{(t)}$ the set of all *a-almost Ψ -embedded t -boundaried graphs* and by $\mathcal{E}_g^{(t)}$ the union of all $\mathcal{E}_{\Psi,a}^{(t)}$ for all possible Ψ and a . Given a $(\mathbf{G}, A) \in \mathcal{E}_{\Psi,a}^{(t)}$ and a $(\mathbf{G}', A') \in \mathcal{E}_{\Psi',a'}^{(t')}$, we say that (\mathbf{G}, A) and (\mathbf{G}', A') are *complementary* if $\Psi' = \tilde{\Psi}$, $t = t'$, F_G^A is isomorphic to $F_{G'}^{A'}$, and $\lambda(A) = \lambda(A')$. We denote by $\mathcal{C}(\mathbf{G}, A)$ the set of all members of $\mathcal{E}_g^{(t)}$ that are complementary to (\mathbf{G}, A) . For simplicity, when we refer to a member of $(\mathbf{G}, \emptyset) \in \mathcal{E}_{\Psi,0}^{(t)}$ we write \mathbf{G} instead of (\mathbf{G}, \emptyset) and we call \mathbf{G} *Ψ -embedded t -boundaried graph*.

The concept of a surface-cut decomposition was introduced in [44] in order to accelerate dynamic programming algorithms in surfaces. We next present this concept, adapted to the terminology that we introduced above. A branch-decomposition (T, σ) of a Ψ -embedded t -boundaried graph $\mathbf{G}^A = (G^A, R^A, \lambda^A)$ with root $r \in V(T)$ is called a *surface-cut decomposition* if, for every edge e of T , there exists a surface-cut collection of nooses \mathcal{N}_e of the Ψ -embedded graph G^A , such that:

- The nooses in $\bigcup_{e \in E(T)} \mathcal{N}_e$ are pairwise confluent,
- $N_{e_r} = \mathbf{bd}(\Psi)$,
- for every $e \in E(T)$, it holds that $G_e^A = \Psi_e \cap G^A$, where Ψ_e is the generalized disk bounded by N_e such that $\Psi_e \subseteq \Psi_{e_r}$,

Notice that each G_e^A is a Ψ_e -embedded graph and that all above conditions reduce to a sphere-cut decomposition in case $a = 0$ and \mathbf{G} is a Δ -embedded t -boundaried graph. We define $\mathbf{G}_e = (G_e, R_e, \lambda_e)$ as a t_e -boundaried graph where G_e is obtained by G_e^A after adding to it the vertices in A and the edges of G that connect vertices in A with vertices in G_e^A , $R_e = R(G_e^A) \cup A$, and $\lambda_e = \rho|_{V(G_e)}$. Moreover, we define the *complement* of \mathbf{G}_e as $\tilde{\mathbf{G}}_e = (\tilde{G}_e, R_e, \lambda_e)$, where $\tilde{G}_e = G_e \setminus (V(G_e) \setminus R_e)$. Notice that for every $e \in E(T)$, $(\mathbf{G}_e, A) \in \mathcal{E}_{\Psi_e,a}^{(t_e)}$ and $(\tilde{\mathbf{G}}_e, A) \in \mathcal{E}_{\tilde{\Psi}_e,a}^{(t_e)}$. Keep in mind that, for every $e \in E(T)$, (\mathbf{G}_e, A) and $(\tilde{\mathbf{G}}_e, A)$ are complementary members of $\mathcal{E}_g^{(t)}$ and that $\mathbf{G}_e \oplus \tilde{\mathbf{G}}_e = G$.

It was proved in [44] that there is an algorithm that, given a polyhedrally Ψ -embedded t -boundaried graph \mathbf{G}^A where $\text{bw}(\mathbf{G}^A) \leq w$, it returns a surface-cut decomposition of \mathbf{G}^A of width $\mathcal{O}(w)$, in $2^{\mathcal{O}(w)} \cdot n^3$ steps. By adding the apex set A to all the t_e -boundaried graphs \mathbf{G}_e^A corresponding to this decomposition, we obtain the decomposition on which the algorithm of Theorem 3.3 now applies. It is important to keep in mind that now we are working with members of $\mathcal{E}_g^{(t)}$, therefore we need to adapt the equivalence relation $\equiv^{\mathcal{F},t}$ for $\mathcal{E}_g^{(t)}$. Then we use the almost-embeddability property in order to prove that the number of equivalence classes – and therefore the tables of the dynamic programming – is bounded by a function that is single-exponential in t_e .

We define the equivalence relation $\equiv_{\Psi,a}^{(\mathcal{F},t)}$ on $\mathcal{E}_{\Psi,a}^{(t)}$ as follows:

Given $(\mathbf{G}_1, A_1), (\mathbf{G}_2, A_2) \in \mathcal{E}_{\Psi,a}^{(t)}$, we say that $(\mathbf{G}_1, A_1) \equiv_{\Psi,a}^{(\mathcal{F},t)} (\mathbf{G}_2, A_2)$ if

$$\forall \mathbf{G} \in \mathcal{C}(\mathbf{G}, A) \quad \mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{G}_1 \iff \mathcal{F} \preceq_{\text{tm}} \mathbf{G} \oplus \mathbf{G}_2.$$

We also say that $(\mathbf{G}_1, A_1) \equiv_g^{(\mathcal{F},t)} (\mathbf{G}_2, A_2)$ if both $(\mathbf{G}_1, A_1), (\mathbf{G}_2, A_2)$ are a -almost Ψ -embedded t -boundaried graphs and $(\mathbf{G}_1, A_1) \equiv_{\Psi,a}^{(\mathcal{F},t)} (\mathbf{G}_2, A_2)$. Again using Theorem 4.5, it is easy to derive that the number of equivalence classes of $\equiv_g^{(\mathcal{F},t)}$ depends only on g, t , and $d = \text{size}(\mathcal{F})$. It is now easy to see that Lemma 9.3 can be extended to a set of representatives $\mathcal{R}_g^{(\mathcal{F},t)}$ of $\equiv_g^{(\mathcal{F},t)}$, i.e., that $|\mathcal{R}_g^{(\mathcal{F},t)}| = 2^{\mathcal{O}_{g+d}(t)}$ and that $\mathcal{R}_g^{(\mathcal{F},t)}$ can be constructed in $2^{\mathcal{O}_{g+d}(t)}$ steps. Indeed, this is a direct consequence of the fact that an analogue of Lemma 9.2 can be proved using that $|\mathcal{L}_{\Psi,a}^{(t)}| = 2^{\mathcal{O}_g(t)}$ (for $a = \mathcal{O}(g)$) and the fact that there are $2^{\mathcal{O}_g(t)}$ Σ_g -embedded graphs on t vertices (this follows from the bounds on planar graphs by a standard planarization argument cutting along non-contractible cycles). Now, by replacing $\mathcal{R}^{(\mathcal{F},t)}$ by $\mathcal{R}_g^{(\mathcal{F},t)}$ in the tables of the dynamic programming algorithm of Theorem 3.3, we can derive that, for Σ_g -embedded graphs, the algorithm runs in $2^{\mathcal{O}_{d+g}(\text{tw})} \cdot n$ steps, as claimed. \square

Again, by Lemma 3.7 and Observation 1 we obtain the following counterpart of Theorem 10.1 for the minor version.

THEOREM 10.2. *If \mathcal{F} is a proper collection containing a planar graph, where $d = \text{size}(\mathcal{F})$, then there exists an algorithm that solves \mathcal{F} -M-DELETION on Σ_g -embedded graphs in $2^{\mathcal{O}_{d+g}(\text{tw})} \cdot n^3$ steps.*

11. Conclusions and further research. We presented parameterized algorithms for \mathcal{F} -M-DELETION and \mathcal{F} -TM-DELETION taking as parameter the treewidth of the input graph. These algorithms are complemented by single-exponential algorithms and lower bounds presented in [5, 6].

The ultimate goal is to establish the tight complexity of \mathcal{F} -M-DELETION and \mathcal{F} -TM-DELETION for all collections \mathcal{F} . Recently, the authors made a significant step in this direction [4], by providing an algorithm to solve \mathcal{F} -M-DELETION in time $\mathcal{O}^*(2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})})$ for every collection \mathcal{F} . This algorithm uses, as a black box, the algorithm of Section 8 (as stated in Theorem 8.1), as well as other ingredients such as Bidimensionality [21], the irrelevant vertex technique [43, 42], and recent results about rerouting paths on flat structures [26]. Note that this result vastly generalizes the ones of Jansen et al. [30] and Kociumaka and Pilipczuk [34], running in time $\mathcal{O}^*(2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})})$, for the problems of deleting a minimum number of vertices to obtain a planar graph and a graph of Euler genus at most g , respectively. Combined with the lower bounds presented in [6], the algorithm in [4] settles completely the complexity

of \mathcal{F} -M-DELETION when \mathcal{F} consists of a single connected graph. Determining the tight complexity, conceivably either $\mathcal{O}^*(2^{\mathcal{O}(\text{tw})})$ or $\mathcal{O}^*(2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})})$, when \mathcal{F} contains more than one graph, or when the graph(s) in \mathcal{F} may be disconnected, remains still open.

On the other hand, we still do not have such a general algorithm running in time $\mathcal{O}^*(2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})})$ for the \mathcal{F} -TM-DELETION problem, the fastest one being the algorithm of Theorem 3.1. In particular, we do not know whether there exists some \mathcal{F} for which there is a lower bound of, say, $2^{o(\text{tw}^2)} \cdot n^{\mathcal{O}(1)}$ for \mathcal{F} -TM-DELETION.

We presented single-exponential algorithms for \mathcal{F} -DELETION when the input graph is planar or, more generally, embedded in a fixed surface. In both cases, the key tool is a special type of branch decomposition with nice topological properties. It seems plausible that this result could be extended to input graphs excluding a fixed graph H as a minor, by using the so-called *H-minor-free cut decompositions* introduced by Rué et al. [45].

In the last years, the \mathcal{F} -M-DELETION problem has been extensively studied in the literature taking as the parameter the size of the solution [32, 22, 31, 33, 30, 46]. In all these papers, FPT-algorithms parameterized by treewidth play a fundamental role. The results that we presented in this paper have already been used in [4], which in turn has been strongly used in the algorithms in [46].

Our results have also interesting consequences by applying the Bidimensionality framework [21, 15, 16], as we proceed to discuss. Let \mathcal{F} be a collection containing a planar graph and consider the \mathcal{F} -M-DELETION problem parameterized by the solution size k , restricted to input graphs G that exclude some fixed graph H as a minor. The linearity in terms of treewidth of the size of a largest grid in an H -minor-free graph [17] implies that positive instances of \mathcal{F} -M-DELETION have treewidth $\mathcal{O}_{\mathcal{F},H}(\sqrt{k})$. Indeed, otherwise, for any solution $S \subseteq V(G)$ of size at most k , $G \setminus S$ would contain as a minor a large enough grid, as a function of \mathcal{F} , which would contain a planar graph in \mathcal{F} as a minor, contradicting the fact that S is a solution. Therefore, when \mathcal{F} contains a planar graph, Theorem 3.4 yields an algorithm to solve \mathcal{F} -M-DELETION in time $2^{\mathcal{O}_{\mathcal{F},H}(\sqrt{k} \cdot \log k)} \cdot n$ when the input graph is H -minor-free, and, by Theorem 10.2, in time $2^{\mathcal{O}_{\mathcal{F},g}(\sqrt{k})} \cdot n$ when the input graph has genus at most g . Plausibly, using the results in [45], the running time $2^{\mathcal{O}_{\mathcal{F},g}(\sqrt{k})} \cdot n$ could be achieved also for H -minor-free graphs. To be best of our knowledge, subexponential algorithms for \mathcal{F} -M-DELETION on these classes of graphs were not known before.

Acknowledgements. We would like to thank the referees of the two conference versions containing some of the results of this article, and those of the current version, for helpful remarks that improved the presentation of the manuscript, in particular for pointing out the fact that our results yield subexponential algorithms parameterized by the size of the solution.

REFERENCES

- [1] I. ADLER, F. DORN, F. V. FOMIN, I. SAU, AND D. M. THILIKOS, *Faster parameterized algorithms for minor containment*, Theoretical Computer Science, 412 (2011), pp. 7018–7028.
- [2] J. BASTE, M. NOY, AND I. SAU, *On the number of labeled graphs of bounded treewidth*, European Journal of Combinatorics, 71 (2018), pp. 12–21, <https://doi.org/https://doi.org/10.1016/j.ejc.2018.02.030>.
- [3] J. BASTE, I. SAU, AND D. M. THILIKOS, *Optimal algorithms for hitting (topological) minors on graphs of bounded treewidth*, in Proc. of the 12th International Symposium on Parameterized and Exact Computation (IPEC), vol. 89 of LIPIcs, 2017, pp. 4:1–4:12.

- [4] J. BASTE, I. SAU, AND D. M. THILIKOS, *A complexity dichotomy for hitting connected minors on bounded treewidth graphs: the chair and the banner draw the boundary*, in Proc. of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA), 2020, pp. 951–970, <https://doi.org/10.1137/1.9781611975994.57>.
- [5] J. BASTE, I. SAU, AND D. M. THILIKOS, *Hitting minors on bounded treewidth graphs. II. Single-exponential algorithms*, Theoretical Computer Science, 814 (2020), pp. 135–152, <https://doi.org/10.1016/j.tcs.2020.01.026>, <https://doi.org/10.1016/j.tcs.2020.01.026>.
- [6] J. BASTE, I. SAU, AND D. M. THILIKOS, *Hitting minors on bounded treewidth graphs. III. Lower bounds*, Journal of Computer and System Sciences, 109 (2020), pp. 56–77, <https://doi.org/10.1016/j.jcss.2019.11.002>, <https://doi.org/10.1016/j.jcss.2019.11.002>.
- [7] H. L. BODLAENDER, M. CYGAN, S. KRATSCHE, AND J. NEDERLOF, *Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth*, Information and Computation, 243 (2015), pp. 86–111.
- [8] 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 (2016), pp. 317–378.
- [9] H. L. BODLAENDER, F. V. FOMIN, D. LOKSHTANOV, E. PENNINKX, S. SAURABH, AND D. M. THILIKOS, *(Meta) Kernelization*, Journal of the ACM, 63 (2016), pp. 44:1–44:69.
- [10] H. L. BODLAENDER AND B. VAN ANTWERPEN-DE FLUITER, *Reduction algorithms for graphs of small treewidth*, Information and Computation, 167 (2001), pp. 86–119, <https://doi.org/10.1006/inco.2000.2958>.
- [11] N. BONICHON, C. GAVOILLE, N. HANUSSE, D. POULALHON, AND G. SCHAEFFER, *Planar graphs, via well-orderly maps and trees*, Graphs and Combinatorics, 22 (2006), pp. 185–202.
- [12] B. COURCELLE, *The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs*, Information and Computation, 85 (1990), pp. 12–75, [http://dx.doi.org/10.1016/0890-5401\(90\)90043-H](http://dx.doi.org/10.1016/0890-5401(90)90043-H).
- [13] M. CYGAN, F. V. FOMIN, L. KOWALIK, D. LOKSHTANOV, D. MARX, M. PILIPCZUK, M. PILIPCZUK, AND S. SAURABH, *Parameterized Algorithms*, Springer, 2015.
- [14] 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), 2011, pp. 150–159.
- [15] E. DEMAINE AND M. HAJIAGHAYI, *The Bidimensionality Theory and Its Algorithmic Applications*, The Computer Journal, 51 (2007), pp. 292–302.
- [16] E. D. DEMAINE, F. V. FOMIN, M. HAJIAGHAYI, AND D. M. THILIKOS, *Subexponential parameterized algorithms on graphs of bounded genus and H -minor-free graphs*, Journal of the ACM, 52 (2005), pp. 866–893.
- [17] E. D. DEMAINE AND M. T. HAJIAGHAYI, *Graphs excluding a fixed minor have grids as large as treewidth, with combinatorial and algorithmic applications through bidimensionality*, in Proc. of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2005, pp. 682–689, <http://dl.acm.org/citation.cfm?id=1070432.1070528>.
- [18] R. DIESTEL, *Graph Theory*, vol. 173, Springer-Verlag, 4th edition, 2010.
- [19] F. DORN, E. PENNINKX, H. L. BODLAENDER, AND F. V. FOMIN, *Efficient Exact Algorithms on Planar Graphs: Exploiting Sphere Cut Decompositions*, Algorithmica, 58 (2010), pp. 790–810.
- [20] R. G. DOWNEY AND M. R. FELLOWS, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013.
- [21] F. V. FOMIN, E. D. DEMAINE, M. T. HAJIAGHAYI, AND D. M. THILIKOS, *Bidimensionality*, in Encyclopedia of Algorithms, 2016, pp. 203–207, https://doi.org/10.1007/978-1-4939-2864-4_47.
- [22] 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), 2012, pp. 470–479.
- [23] 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 (2016), pp. 29:1–29:60.
- [24] 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), 2010, pp. 503–510. Full version available at *CoRR*, abs/1606.05689, 2016.
- [25] V. GARNERO, C. PAUL, I. SAU, AND D. M. THILIKOS, *Explicit linear kernels via dynamic programming*, SIAM Journal on Discrete Mathematics, 29 (2015), pp. 1864–1894.
- [26] P. A. GOLOVACH, G. STAMOULIS, AND D. M. THILIKOS, *Hitting Topological Minor Models*

- in Planar Graphs is Fixed Parameter Tractable*, in Proc. of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA), 2020, pp. 931–950, <https://doi.org/10.1137/1.9781611975994.56>.
- [27] M. GROHE, K. KAWARABAYASHI, D. MARX, AND P. WOLLAN, *Finding topological subgraphs is fixed-parameter tractable*, in Proc. of the 43rd ACM Symposium on Theory of Computing (STOC), 2011, pp. 479–488, <https://doi.org/10.1145/1993636.1993700>.
- [28] Q. GU AND H. TAMAKI, *Optimal branch-decomposition of planar graphs in $O(n^3)$ time*, ACM Transactions on Algorithms, 4 (2008), pp. 30:1–30:13.
- [29] R. IMPAGLIAZZO, R. PATURI, AND F. ZANE, *Which problems have strongly exponential complexity?*, Journal of Computer and System Sciences, 63 (2001), pp. 512–530.
- [30] 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), 2014, pp. 1802–1811.
- [31] G. JORET, C. PAUL, I. SAU, S. SAURABH, AND S. THOMASSÉ, *Hitting and harvesting pumpkins*, SIAM Journal on Discrete Mathematics, 28 (2014), pp. 1363–1390.
- [32] 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 (2016), pp. 21:1–21:41.
- [33] E. J. KIM, C. PAUL, AND G. PHILIP, *A single-exponential FPT algorithm for the K_4 -minor cover problem*, Journal of Computer and System Sciences, 81 (2015), pp. 186–207.
- [34] T. KOCIUMAKA AND M. PILIPCZUK, *Deleting vertices to graphs of bounded genus*, Algorithmica, 81 (2019), pp. 3655–3691, <https://doi.org/10.1007/s00453-019-00592-7>.
- [35] J. KOMLÓS AND E. SZEMERÉDI, *Topological cliques in graphs II*, Combinatorics, Probability & Computing, 5 (1996), pp. 79–90.
- [36] J. M. LEWIS AND M. YANNAKAKIS, *The node-deletion problem for hereditary properties is NP-complete*, Journal of Computer and System Sciences, 20 (1980), pp. 219–230.
- [37] D. LOKSHTANOV, D. MARX, AND S. SAURABH, *Slightly superexponential parameterized problems*, SIAM Journal on Computing, 47 (2018), pp. 675–702, <https://doi.org/10.1137/16M1104834>.
- [38] B. MOHAR AND C. THOMASSEN, *Graphs on surfaces*, Johns Hopkins University Press, 2001.
- [39] M. PILIPCZUK, *A tight lower bound for Vertex Planarization on graphs of bounded treewidth*, Discrete Applied Mathematics, 231 (2017), pp. 211–216.
- [40] N. ROBERTSON AND P. D. SEYMOUR, *Graph minors. V. Excluding a planar graph*, Journal of Combinatorial Theory, Series B, 41 (1986), pp. 92–114.
- [41] N. ROBERTSON AND P. D. SEYMOUR, *Graph minors. X. Obstructions to tree decomposition*, Journal of Combinatorial Theory, Series B, 52 (1991), pp. 153–190.
- [42] N. ROBERTSON AND P. D. SEYMOUR, *Graph Minors. XIII. The Disjoint Paths Problem*, Journal of Combinatorial Theory, Series B, 63 (1995), pp. 65–110, <https://doi.org/10.1006/jctb.1995.1006>.
- [43] N. ROBERTSON AND P. D. SEYMOUR, *Graph Minors. XXII. Irrelevant vertices in linkage problems*, Journal of Combinatorial Theory, Series B, 102 (2012), pp. 530–563, <https://doi.org/10.1016/j.jctb.2007.12.007>.
- [44] J. RUÉ, I. SAU, AND D. M. THILIKOS, *Dynamic programming for graphs on surfaces*, ACM Transactions on Algorithms, 10 (2014), pp. 8:1–8:26.
- [45] J. RUÉ, I. SAU, AND D. M. THILIKOS, *Dynamic programming for H -minor-free graphs*, in Proc. of the 18th Annual International Conference on Computing and Combinatorics (COCOON), vol. 7434 of LNCS, 2012, pp. 86–97.
- [46] I. SAU, G. STAMOULIS, AND D. M. THILIKOS, *An FPT-algorithm for recognizing k -apices of minor-closed graph classes*, CoRR, abs/2004.12692 (2020), <https://arxiv.org/abs/2004.12692>. To appear in Proc. of the 47th International Colloquium on Automata, Languages and Programming (ICALP 2020).
- [47] P. D. SEYMOUR AND R. THOMAS, *Call routing and the ratcatcher*, Combinatorica, 14 (1994), pp. 217–241.
- [48] W. T. TUTTE, *A census of planar triangulations*, Canadian Journal of Mathematics, 14 (1962), pp. 21–38.
- [49] H. WHITNEY, *A set of topological invariants for graphs*, American Journal of Mathematics, 55 (1933), pp. 231–235.