



HAL
open science

A Complexity Dichotomy for Hitting Small Planar Minors Parameterized by Treewidth

Julien Baste, Ignasi Sau, Dimitrios M. Thilikos

► **To cite this version:**

Julien Baste, Ignasi Sau, Dimitrios M. Thilikos. A Complexity Dichotomy for Hitting Small Planar Minors Parameterized by Treewidth. IPEC 2018 - 13th International Symposium on Parameterized and Exact Computation, Aug 2018, Helsinki, Finland. pp.2:1–2:13, 10.4230/LIPIcs.IPEC.2018.2.lirmm-02342806

HAL Id: lirmm-02342806

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-02342806>

Submitted on 1 Nov 2019

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.



Distributed under a Creative Commons Attribution 4.0 International License

A Complexity Dichotomy for Hitting Small Planar Minors Parameterized by Treewidth

Julien Baste


Sorbonne Université, Laboratoire d'Informatique de Paris 6, LIP6, Paris, France

julien.baste@uni-ulm.de

Ignasi Sau

LIRMM, CNRS, Université de Montpellier, Montpellier, France

ignasi.sau@lirmm.fr


 <https://orcid.org/0000-0002-8981-9287>

Dimitrios M. Thilikos

LIRMM, CNRS, Université de Montpellier, Montpellier, France, and

Department of Mathematics, National and Kapodistrian University of Athens, Greece

sedthilk@thilikos.info

 <https://orcid.org/0000-0003-0470-1800>

Abstract

For a fixed graph H , we are interested in the parameterized complexity of the following problem, called $\{H\}$ -M-DELETION, parameterized by the treewidth tw of the input graph: given an n -vertex graph G and an integer k , decide whether there exists $S \subseteq V(G)$ with $|S| \leq k$ such that $G \setminus S$ does not contain H as a minor. In previous work [IPEC, 2017] we proved that if H is planar and connected, then the problem cannot be solved in time $2^{o(\text{tw})} \cdot n^{O(1)}$ under the ETH, and can be solved in time $2^{O(\text{tw} \cdot \log \text{tw})} \cdot n^{O(1)}$. In this article we manage to classify the optimal asymptotic complexity of $\{H\}$ -M-DELETION when H is a connected planar graph on at most 5 vertices. Out of the 29 possibilities (discarding the trivial case $H = K_1$), we prove that 9 of them are solvable in time $2^{\Theta(\text{tw})} \cdot n^{O(1)}$, and that the other 20 ones are solvable in time $2^{\Theta(\text{tw} \cdot \log \text{tw})} \cdot n^{O(1)}$. Namely, we prove that K_4 and the diamond are the only graphs on at most 4 vertices for which the problem is solvable in time $2^{\Theta(\text{tw} \cdot \log \text{tw})} \cdot n^{O(1)}$, and that the chair and the banner are the only graphs on 5 vertices for which the problem is solvable in time $2^{\Theta(\text{tw})} \cdot n^{O(1)}$. For the version of the problem where H is forbidden as a *topological* minor, the case $H = K_{1,4}$ can be solved in time $2^{\Theta(\text{tw})} \cdot n^{O(1)}$. This exhibits, to the best of our knowledge, the first difference between the computational complexity of both problems.

2012 ACM Subject Classification Mathematics of computing \rightarrow Graph algorithms, Theory of computation \rightarrow Parameterized complexity and exact algorithms

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

Digital Object Identifier 10.4230/LIPIcs.IPEC.2018.2

Related Version All the results and proofs of this article, as well as those of [1] and some other ones, are permanently available at <https://arxiv.org/abs/1704.07284>.

Funding Work supported by projects DEMOGRAPH (ANR-16-CE40-0028) and ESIGMA (ANR-17-CE23-0010).

Acknowledgements We would like to thank Eun Jung Kim and Édouard Bonnet for insightful discussions on the topic of this paper.



© Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos;
licensed under Creative Commons License CC-BY

13th International Symposium on Parameterized and Exact Computation (IPEC 2018).

Editors: Christophe Paul and Michal Pilipczuk; Article No. 2; pp. 2:1–2:13

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Let H be a fixed graph. In the $\{H\}$ -M-DELETION (resp. $\{H\}$ -TM-DELETION) problem, we are given an n -vertex 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 H as a minor (resp. topological minor). These problems belong to the more general category of *graph modification problems*. The cases where H is planar and connected are already quite general, as the cases $H = K_2$ and $H = K_3$ correspond to VERTEX COVER and FEEDBACK VERTEX SET, respectively. We are interested in the parameterized complexity of $\{H\}$ -M-DELETION and $\{H\}$ -TM-DELETION taking as the parameter the treewidth of G , denoted by tw .

Determining the optimal asymptotic complexity of $\{H\}$ -M-DELETION parameterized by treewidth has been an active area in the parameterized complexity community during the last years. As relevant examples, VERTEX COVER is easily solvable in time $2^{\mathcal{O}(\text{tw})} \cdot n^{\mathcal{O}(1)}$, 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)¹ fails [10]. For FEEDBACK VERTEX SET, the existence of a single-exponential algorithm remained open for a while, until Cygan et al. [6] presented the *Cut&Count* technique. See also [2, 9, 11, 15].

We recently studied these problems in [1] and proved², among other results, that if H is planar and connected, then the problems cannot be solved in time $2^{o(\text{tw})} \cdot n^{\mathcal{O}(1)}$ under the ETH, and can be solved in time $2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$ (for $\{H\}$ -TM-DELETION, we additionally need H to have maximum degree at most 3). We also presented a dichotomy when $H = C_i$ is a cycle on i vertices, by proving that both problems (which are clearly equivalent for subcubic graphs) can be solved in single-exponential time if and only if $i \leq 4$. We aimed at a similar dichotomy when $H = P_i$ is a path on i vertices, but we left open the case $H = P_5$.

In this article we obtain the following results, the lower bounds holding under the ETH:

1. When $H = K_{1,i}$, the star with i leaves, $\{K_{1,i}\}$ -M-DELETION is solvable in time $2^{\Theta(\text{tw})} \cdot n^{\mathcal{O}(1)}$ for $i \leq 3$, and in time $2^{\Theta(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$ for $i \geq 4$. On the other hand, $\{K_{1,i}\}$ -TM-DELETION can be solved in time $2^{\Theta(\text{tw})} \cdot n^{\mathcal{O}(1)}$ for every $i \geq 1$. To the best of our knowledge, this is the first example of a graph H for which the complexity of $\{H\}$ -M-DELETION and $\{H\}$ -TM-DELETION differ.
2. When $H = \theta_i$, the multigraph consisting of two vertices and $i \geq 1$ parallel edges, both problems can be solved in time $2^{\Theta(\text{tw})} \cdot n^{\mathcal{O}(1)}$ for $i \leq 2$, and in time $2^{\Theta(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$ for $i \geq 3$. The same dichotomy occurs when $H = K_{2,i}$.
3. We classify the optimal asymptotic complexity of $\{H\}$ -M-DELETION when H is a connected planar graph on at most 5 vertices. Out of the 29 possibilities (discarding the trivial case $H = \{K_1\}$), we prove that 9 of them are solvable in time $2^{\Theta(\text{tw})} \cdot n^{\mathcal{O}(1)}$, and that the other 20 ones are solvable in time $2^{\Theta(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$; a summary is shown in Figure 1. Note that K_4 and the *diamond* are the only graphs on at most 4 vertices for which the problem is solvable in time $2^{\Theta(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$, and that the *chair* and the *banner* are the only graphs on 5 vertices for which the problem is solvable in time $2^{\Theta(\text{tw})} \cdot n^{\mathcal{O}(1)}$. In particular, this settles the complexity of $\{P_5\}$ -M-DELETION, which we left open in [1]. All the lower bounds also hold for $\{H\}$ -TM-DELETION, with the only difference that the case $H = K_{1,4}$ can be solved in time $2^{\Theta(\text{tw})} \cdot n^{\mathcal{O}(1)}$, as mentioned in item 1 above.

¹ The ETH states that 3-SAT on n variables cannot be solved in time $2^{o(n)}$; see [10] for more details.

² In [1] we considered the more general case where all the graphs in a fixed finite family \mathcal{F} are forbidden as (topological) minors. For simplicity, we only consider here the case where \mathcal{F} contains a single graph.

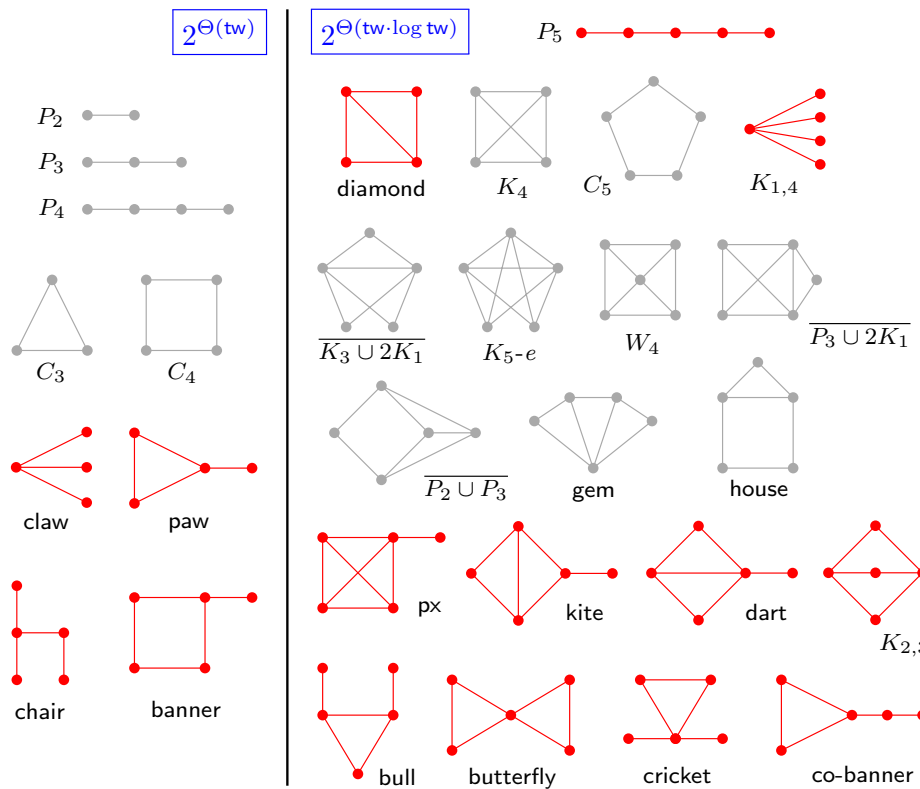


Figure 1 Classification of the complexity of $\{H\}$ -M-DELETION for all connected simple planar graphs H with $|V(H)| \leq 5$ and $|E(H)| \geq 1$: for the 9 graphs on the left (resp. 20 graphs on the right), the problem is solvable in time $2^{\Theta(tw)} \cdot n^{\mathcal{O}(1)}$ (resp. $2^{\Theta(tw \cdot \log tw)} \cdot n^{\mathcal{O}(1)}$). For $\{H\}$ -TM-DELETION, $K_{1,4}$ should be on the left. The results of this article correspond to the red (darker) graphs; the other ones were either already known (namely, P_2 , P_3 , and C_3) or obtained in [1].

Let us discuss about the techniques that we used to obtain the above results. The single-exponential algorithms are ad hoc, some being easier than others. All of them exploit a structural characterization of the graphs that exclude that particular graph as a (topological) minor; cf. for instance Lemma 1. Intuitively, the “complexity” of this characterization is what determines the difficulty of the corresponding dynamic programming algorithm, and is also what makes the difference between being solvable in single-exponential time or not.

More precisely, the algorithms for $\{K_{1,s}\}$ -TM-DELETION are simple and use standard dynamic programming techniques on graphs of bounded treewidth. The algorithm for $\{\text{paw}\}$ -TM-DELETION is more involved and uses the rank-based approach introduced by Bodlaender et al. [2], similarly to the algorithm for $\{C_4\}$ -TM-DELETION that we presented in [1]. Finally, the algorithms for $\{\text{chair}\}$ -TM-DELETION and $\{\text{banner}\}$ -TM-DELETION are a combination of some of the algorithms given here and in [1], the latter one using the rank-based approach.

The superexponential lower bounds of this article are inspired by a reduction of Bonnet et al. [4]. Namely, we prove subexponential lower bounds for P_5 , $K_{1,i}$ with $i \geq 4$ for the minor version, $K_{2,i}$ and θ_i for $i \geq 3$ (note that if G is a simple graph, $\{\theta_3\}$ -DELETION is equivalent to $\{\text{diamond}\}$ -DELETION), and the following graphs depicted in Figure 1: the px, the kite, the dart, the bull, the butterfly, the cricket, and the co-banner. All these reductions are based on a general construction (cf. Section 4.1), and then we need particular small gadgets to deal with each of the graphs. On the other hand, one can easily check that all the other graphs

on the right hand side of Figure 1, namely K_4 , C_5 , $\overline{K_3 \cup 2K_1}$, K_{5-e} , W_4 , $\overline{P_3 \cup 2K_1}$, $\overline{P_2 \cup P_3}$, the gem, and the house, satisfy the general condition given in [1, Theorem 20], and therefore the superexponential lower bound follows for both problems and each of these graphs. This completes the dichotomy for all connected (simple) planar graphs on at most 5 vertices.

The remainder of this article is organized as follows. In Section 2 we provide some preliminaries. The single-exponential algorithms are presented in Section 3 and the superexponential lower bounds in Section 4. We conclude the article in Section 5. Due to space constraints, the proofs of the results marked with ‘(★)’ can be found in the full version of this article.

2 Preliminaries

We use standard graph-theoretic notation, and we refer to [7] for any undefined term and the notions of minor and topological minor. We also refer to [8, 5] for the basic definitions of parameterized complexity, tree decompositions, and treewidth. We need to introduce nice tree decompositions, which make the presentation of the algorithms much simpler.

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

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

For each $t \in V(T)$, we denote by V_t the set $V(G_t)$. Given a tree decomposition, it is possible to transform it in polynomial time to a *nice* new one of the same width [12]. Moreover, by Bodlaender et al. [3] we can find in time $2^{\mathcal{O}(\text{tw})} \cdot n$ a tree decomposition of width $\mathcal{O}(\text{tw})$ of any graph G . Hence, since in this article we focus on single-exponential algorithms, we may assume that a nice tree decomposition of width $w = \mathcal{O}(\text{tw})$ is given with the input.

If a graph G contains a graph H as a minor (resp. topological minor), we denote it by $H \preceq_m G$ (resp. $H \preceq_{\text{tm}} G$). For a fixed graph H and a graph G , we define the parameter $\mathbf{m}_{\{H\}}(G)$ (resp. $\mathbf{tm}_{\{H\}}(G)$) as the minimum size of a set $S \subseteq V(G)$ such that $H \not\preceq_m G \setminus S$ (resp. $H \not\preceq_{\text{tm}} G \setminus S$).

3 Single-exponential algorithms

In this section we present single-exponential algorithms for hitting particular graphs. Since the algorithms when $H = K_{1,s}$ use standard dynamic programming techniques, they have been moved to the full version. In what follows we present a single-exponential algorithm for {paw}-TM-DELETION. For completeness, the basic ingredients and notations of the rank-based approach of Bodlaender et al. [2] are given in the full version. The algorithms for the chair and the banner are also given in the full version.

We start with a simple structural characterization of the simple graphs that exclude the paw as a topological minor; recall the paw graph in Figure 1.

► **Lemma 1.** *A simple graph G satisfies $\text{paw} \not\leq_{\text{tm}} G$ if and only if each connected component of G is either a cycle or a tree.*

Proof. It is easy to see that neither a cycle nor a tree contain the **paw** as a topological minor. Let G be a graph such that $\text{paw} \not\leq_{\text{tm}} G$. Let us assume w.l.o.g. that G is connected. If G does not contain a cycle, then it is a tree. Otherwise, let C be a chordless cycle in G . If G contains a vertex v that is not in C , then, as G is connected, there exists a path from v to C containing at least 2 vertices. This is not possible, as it would imply that G contains the **paw** as a topological minor. As C is chordless and G is simple, we obtain that G is exactly the cycle C , and the lemma follows. ◀

We present an algorithm that solves the decision version of $\{\text{PAW}\}$ -TM-DELETION. As the algorithm that we presented for $\{C_4\}$ -TM-DELETION in [1], this algorithm is based on the one given in [2, Section 3.5] for FEEDBACK VERTEX SET. Let G be a graph and k be an integer. The idea of the following algorithm is to partition $V(G)$ into three sets. The first one will be the solution set S , the second one will be a set F of vertices that induces a forest, and the third one will be a set C of vertices that induces a collection of cycles. If we can partition our graph into three such sets (S, F, C) such that there is no edge between a vertex of F and a vertex of C and such that $|S| \leq k$, then, using Lemma 1, we know that $\text{tm}_{\{\text{paw}\}}(G) \leq k$. On the other hand, if such a partition does not exist, we know that $\text{tm}_{\{\text{paw}\}}(G) > k$. The main idea of this algorithm is to combine classical dynamic programming techniques in order to verify that C induces a collection of cycles, and the rank-based approach in order to verify that F induces a forest.

As for $\{C_4\}$ -TM-DELETION (see [1]), we define a new graph $G_0 = (V(G) \cup \{v_0\}, E(G) \cup E_0)$, where v_0 is a new vertex and $E_0 = \{\{v_0, v\} \mid v \in V(G)\}$. For each subgraph H of G_0 , for each $Z_1 \subseteq V(H)$, and for each $Y \subseteq E_0 \cap E(H[Z_1])$, we denote by $H\langle Z_1, Y \rangle$ the graph $(Z_1, Y \cup E(H[Z_1 \setminus \{v_0\}]))$.

Given a nice tree decomposition of G of width w , we define a nice tree decomposition $((T, \mathcal{X}), r, \mathcal{G})$ of G_0 of width $w + 1$ such that the only empty bags are the root and the leaves and for each $t \in T$, if $X_t \neq \emptyset$, then $v_0 \in X_t$. Note that this can be done in linear time. For each bag t , each integers i, j , and ℓ , each function $\mathbf{s} : X_t \rightarrow \{0, 1, 2_0, 2_1, 2_2\}$, each function $\mathbf{s}_0 : \{v_0\} \times \mathbf{s}^{-1}(1) \rightarrow \{0, 1\}$, and each partition $p \in \Pi(\mathbf{s}^{-1}(1))$, we define:

$$\begin{aligned} \mathcal{E}_t(p, \mathbf{s}, \mathbf{s}_0, i, j, \ell) = & \{(Z_1, Z_2, Y) \mid (Z_1, Z_2, Y) \in 2^{V_t} \times 2^{V_t} \times 2^{E_0 \cap E(G_t)}, Z_1 \cap Z_2 = \emptyset, \\ & |Z_1| = i, |Z_2| = \ell, |E(G_t[Z_1 \setminus \{v_0\}]) \cup Y| = j, \\ & \forall e \in E_0 \cap E_t, \mathbf{s}_0(e) = 1 \Leftrightarrow e \in Y, \\ & \forall v \in Z_2 \cap X_t, \mathbf{s}(v) = 2_z \text{ with } z = \text{deg}_{G_t[Z_2]}(v), \\ & \forall v \in Z_2 \setminus X_t, \text{deg}_{G_t[Z_2]}(v) = 2, \\ & Z_1 \cap X_t = \mathbf{s}^{-1}(1), v_0 \in X_t \Rightarrow \mathbf{s}(v_0) = 1, \\ & \forall u \in Z_1 \setminus X_t : \text{either } t \text{ is the root or} \\ & \quad \exists u' \in \mathbf{s}^{-1}(1) : u \text{ and } u' \text{ are connected in } G_t\langle Z_1, Y \rangle, \\ & \forall v_1, v_2 \in \mathbf{s}^{-1}(1) : p \sqsubseteq V_t[\{v_1, v_2\}] \Leftrightarrow v_1 \text{ and } v_2 \text{ are con-} \\ & \quad \text{nected in } G_t\langle Z_1, Y \rangle, \\ & \forall (u, v) \in (Z_1 \setminus \{v_0\}) \times Z_2, \{u, v\} \notin E(G_t)\} \end{aligned}$$

$$\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, i, j, \ell) = \{p \mid p \in \Pi(\mathbf{s}^{-1}(1)), \mathcal{E}_t(p, \mathbf{s}, \mathbf{s}_0, i, j, \ell) \neq \emptyset\}.$$

In the definition of \mathcal{E}_t , the sets Z_1 (resp. Z_2) correspond to the set F (resp. C) restricted to G_t . The vertex v_0 and the set Y exist to ensure that F will be connected.

By Lemma 1, we have that the given instance of $\{\text{PAW}\}$ -TM-DELETION is a YES-instance if and only if for some i and ℓ , $i + \ell \geq |V(G) \cup \{v_0\}| - k$ and $\mathcal{A}_r(\emptyset, i, i - 1, \ell) \neq \emptyset$. For each $t \in V(T)$, we assume that we have already computed $\mathcal{A}_{t'}$ for every children t' of t , and we proceed to the computation of \mathcal{A}_t . As usual, we distinguish several cases depending on the type of node t .

Leaf. By definition of \mathcal{A}_t , we have $\mathcal{A}_t(\emptyset, \emptyset, 0, 0, 0) = \{\emptyset\}$.

Introduce vertex. Let v be the insertion vertex of X_t , let t' be the child of t , let $\mathbf{s} : X_t \rightarrow \{0, 1, 2_0, 2_1, 2_2\}$, $\mathbf{s}_0 : \{v_0\} \times \mathbf{s}^{-1}(1) \rightarrow \{0, 1\}$, and let $H = G_t \langle \mathbf{s}^{-1}(1), \mathbf{s}_0^{-1}(1) \rangle$.

- If $v = v_0$ and $\mathbf{s}(v_0) \in \{0, 2_0, 2_1, 2_2\}$, then by definition of \mathcal{A}_t we have that $\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, i, j, \ell) = \emptyset$.
- Otherwise, if $v = v_0$, then by construction of the nice tree decomposition, we know that t' is a leaf of T and so $\mathbf{s} = \{(v_0, 1)\}$, $j = \ell = i - 1 = 0$ and $\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, i, j, \ell) = \text{ins}(\{v_0\}, \mathcal{A}_{t'}(\emptyset, \emptyset, 0, 0, 0))$.
- Otherwise, if $\mathbf{s}(v) = 0$, then, by definition of \mathcal{A}_t , it holds that $\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, i, j, \ell) = \mathcal{A}_{t'}(\mathbf{s}|_{X_{t'}}, \mathbf{s}|_{E_{t'}}, i, j, \ell)$.
- Otherwise, if $\mathbf{s}(v) = 2_z$, $z \in \{0, 1, 2\}$, then let $Z'_2 = N_{G_t[X_t]}(v) \setminus \mathbf{s}^{-1}(0)$. If $Z'_2 \not\subseteq \mathbf{s}^{-1}(\{2_1, 2_2\})$ or $|Z'_2| \neq z$, then $\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, i, j, \ell) = \emptyset$. Otherwise $Z'_2 \subseteq \mathbf{s}^{-1}(\{2_1, 2_2\})$ and $|Z'_2| = z$, and with $\mathbf{s}' : X_{t'} \rightarrow \{0, 1, 2_0, 2_1, 2_2\}$ defined such that $\forall v' \in X_{t'} \setminus Z'_2$, $\mathbf{s}'(v') = \mathbf{s}(v')$ and for each $v' \in Z'_2$ such that $\mathbf{s}(v') = 2_{z'}$, $z' \in \{1, 2\}$, $\mathbf{s}'(v') = 2_{z'-1}$. It holds that $\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, i, j, \ell) = \mathcal{A}_{t'}(\mathbf{s}', \mathbf{s}_0, i, j, \ell - 1)$.
- Otherwise, we know that $v \neq v_0$, $\mathbf{s}(v) = 1$, and $v_0 \in N_{G[\mathbf{s}^{-1}(1)]}(v)$. First, if $N_{G_t[X_t]}(v) \setminus \mathbf{s}^{-1}(0) \not\subseteq \mathbf{s}^{-1}(1)$, then $\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, i, j, \ell) = \emptyset$. Indeed, this implies that the cycle part and the forest part are connected. As $\mathbf{s}(v) = 1$, we have to insert v in the forest part and we have to make sure that all vertices of $N_H[v]$ are in the same connected component of H . The only remaining choice is to insert the edge $\{v, v_0\}$ or not. Again, this is handled by the function \mathbf{s}_0 . By adding v , we add one vertex and $|N_H(v)|$ edges in the forest part. Therefore, we have that

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

Forget vertex. Let v be the forget vertex of X_t , let t' be the child of t , and let $\mathbf{s} : X_t \rightarrow \{0, 1, 2_0, 2_1, 2_2\}$. As a vertex from the collection of cycles can be removed only if it has exactly two neighbors, we obtain that

$$\begin{aligned} \mathcal{A}_t(\mathbf{s}, i, j, \ell) &= \mathcal{A}_{t'}(\mathbf{s} \cup \{(v, 0)\}, \mathbf{s}_0, i, j, \ell) \\ &\Downarrow \text{proj}(\{v\}, \mathcal{A}_{t'}(\mathbf{s} \cup \{(v, 1)\}, \mathbf{s}_0 \cup \{(\{v_0, v\}, 0)\}, i, j, \ell)) \\ &\Downarrow \text{proj}(\{v\}, \mathcal{A}_{t'}(\mathbf{s} \cup \{(v, 1)\}, \mathbf{s}_0 \cup \{(\{v_0, v\}, 1)\}, i, j, \ell)) \\ &\Downarrow \mathcal{A}_{t'}(\mathbf{s} \cup \{(v, 2_2)\}, \mathbf{s}_0, i, j, \ell). \end{aligned}$$

Join. Let t' and t'' be the two children of t , let $\mathbf{s} : X_t \rightarrow \{0, 1, 2_0, 2_1, 2_2\}$, $\mathbf{s}_0 : \{v_0\} \times \mathbf{s}^{-1}(1) \rightarrow \{0, 1\}$, and let $H = G_t \langle \mathbf{s}^{-1}(1), \mathbf{s}_0^{-1}(1) \rangle$. Given three functions $\mathbf{s}^*, \mathbf{s}', \mathbf{s}'' : X_t \rightarrow \{0, 1, 2_0, 2_1, 2_2\}$, we say that $\mathbf{s}^* = \mathbf{s}' \oplus \mathbf{s}''$ if for each $v \in \mathbf{s}^{-1}(\{0, 1\})$, $\mathbf{s}^*(v) = \mathbf{s}'(v) = \mathbf{s}''(v)$, and for each $v \in X_t$ such that $\mathbf{s}^*(v) = 2_z$, $z \in \{0, 1, 2\}$, there exist $z', z'' \in \{0, 1, 2\}$ such that $\mathbf{s}'(v) = 2_{z'}$, $\mathbf{s}''(v) = 2_{z''}$, and $z = z' + z'' - \text{deg}_{G_t[X_t \setminus \mathbf{s}^{-1}(0)]}(v)$.

We join every compatible entries $A_{t'}(\mathbf{s}', \mathbf{s}'_0, i', j', \ell')$ and $A_{t''}(\mathbf{s}'', \mathbf{s}''_0, i'', j'', \ell'')$. For two such entries being compatible, we need $\mathbf{s}' \oplus \mathbf{s}''$ to be defined and $\mathbf{s}'_0 = \mathbf{s}''_0$. We obtain that

$$\mathcal{A}_t(\mathbf{s}, \mathbf{s}_0, i, j, \ell) = \bigsqcup_{\substack{\mathbf{s}', \mathbf{s}'' : X_t \rightarrow \{0, 1, 2_0, 2_1, 2_2\}, \\ \mathbf{s} = \mathbf{s}'_1 \oplus \mathbf{s}''_2 \\ i' + i'' = i + |V(H)| \\ j' + j'' = j + |E(H)| \\ \ell' + \ell'' = \ell + |\mathbf{s}^{-1}(\{2_0, 2_1, 2_2\})|}} \text{join}(A_{t'}(\mathbf{s}', \mathbf{s}_0, i', j', \ell'), A_{t''}(\mathbf{s}'', \mathbf{s}_0, \mathbf{r}'', i'', j'', \ell'')).$$

► **Theorem 2.** $\{\text{PAW}\}$ -TM-DELETION can be solved in time $2^{\mathcal{O}(\text{tw})} \cdot n^7$.

Proof. The algorithm works in the following way. For each node $t \in V(T)$ and for each entry M of its table, instead of storing $\mathcal{A}_t(M)$, we store $\mathcal{A}'_t(M) = \text{reduce}(\mathcal{A}_t(M))$ by using [2, Theorem 3.7]. As each of the operations we use preserves representation by [2, Lemma 3.6], we obtain that for each node $t \in V(T)$ and for each possible entry M , $\mathcal{A}'_t(M)$ represents $\mathcal{A}_t(M)$. In particular, we have that $\mathcal{A}'_r(M) = \text{reduce}(\mathcal{A}_r(M))$ for each possible entry M . Using the definition of \mathcal{A}_r and Lemma 1, we have that $\text{tm}_{\{\text{paw}\}}(G) \leq k$ if and only if for some i and ℓ , $i + \ell \geq |V(G) \cup \{v_0\}| - k$ and $\mathcal{A}'_r(\emptyset, i, i - 1, \ell) \neq \emptyset$.

We now focus on the running time of the algorithm. The size of the intermediate sets of weighted partitions for a leaf node and for an introduce vertex node, are upper-bounded by $2^{|\mathbf{s}^{-1}(1)|}$. For a forget vertex node, we take the union of four sets of size $2^{|\mathbf{s}^{-1}(1)|}$, so the intermediate sets of weighted partitions have size at most $4 \cdot 2^{|\mathbf{s}^{-1}(1)|}$. For a join node, as in the big union operation we take into consideration at most $5^{|X_t|}$ possible functions \mathbf{s}' , as many functions \mathbf{s}'' , at most $n + |\mathbf{s}^{-1}(1)|$ choices for i' and i'' , at most $n + |\mathbf{s}^{-1}(1)|$ choices for j' and j'' (as we can always assume, during the algorithm, that H is a forest), and at most $n + |\mathbf{s}^{-1}(\{2_0, 2_1, 2_2\})|$ choices for ℓ' and ℓ'' , we obtain that the intermediate sets of weighted partitions have size at most $25^{|X_t|} \cdot (n + |\mathbf{s}^{-1}(1)|)^2 \cdot (n + |\mathbf{s}^{-1}(\{2_0, 2_1, 2_2\})|) \cdot 4^{|\mathbf{s}^{-1}(1)|}$. We obtain that the intermediate sets of weighted partitions have size at most $(n + |X_t|)^3 \cdot 100^{|X_t|}$. Moreover, for each node $t \in V(T)$, the function reduce will be called as many times as the number of possible entries, i.e., at most $2^{\mathcal{O}(w)} \cdot n^3$ times. Thus, using [2, Theorem 3.7], \mathcal{A}'_t can be computed in time $2^{\mathcal{O}(w)} \cdot n^6$. The theorem follows by taking into account the linear number of nodes in a nice tree decomposition. ◀

4 Superexponential lower bounds

Let $\mathcal{Q} = \{P_5\} \cup \{K_{2,s} \mid s \geq 3\} \cup \{\theta_s \mid s \geq 3\}$ and let $\mathcal{R} = \{K_{1,s} \mid s \geq 4\}$. In this section, we provide superexponential lower bounds for $\{H\}$ -M-DELETION when $H \in \mathcal{Q} \cup \mathcal{R}$, for $\{H\}$ -TM-DELETION when $H \in \mathcal{Q}$, and for both problems for some particular graphs on 5 vertices depicted in Figure 1, namely the cricket, the px, the butterfly, the co-banner, the bull, the kite, and the dart. As mentioned before, the reductions of this section are strongly inspired by the ideas of Bonnet et al. [4]. More precisely, we will prove the following theorems.

► **Theorem 3.** Let $H \in \mathcal{Q} \cup \mathcal{R}$. Unless the ETH fails, $\{H\}$ -M-DELETION cannot be solved in time $2^{\mathcal{O}(\text{tw} \log \text{tw})} \cdot n^{\mathcal{O}(1)}$.

► **Theorem 4.** Let $H \in \mathcal{Q}$. Unless the ETH fails, $\{H\}$ -TM-DELETION cannot be solved in time $2^{\mathcal{O}(\text{tw} \log \text{tw})} \cdot n^{\mathcal{O}(1)}$.

► **Theorem 5.** Let $H \in \{\{\text{cricket}\}, \{\text{px}\}, \{\text{butterfly}\}, \{\text{co-banner}\}, \{\text{bull}\}, \{\text{kite}\}, \{\text{dart}\}\}$. Unless the ETH fails, neither $\{H\}$ -M-DELETION nor $\{H\}$ -TM-DELETION can be solved in time $2^{\mathcal{O}(\text{tw} \log \text{tw})} \cdot n^{\mathcal{O}(1)}$.

In the following we focus on the proof of Theorem 3 and Theorem 5. Theorem 4 can be proved by using the same reductions as for Theorem 3 by just replacing “minor” by “topological minor” and “-M-DELETION” by “-TM-DELETION”. Note that this holds when $H \in \mathcal{Q}$ but not when $H \in \mathcal{R}$.

We first provide in Section 4.1 a general framework that will be used for every $H \in \mathcal{Q} \cup \mathcal{R}$ and the graphs H listed in Theorem 5, and then we explain how to modify this framework for each specific H . Namely, in Section 4.2 we deal with P_5 and in Section 4.3 with the stars. The other cases can be found in the full version of this article. All these proofs for particular graphs are quite similar and follow the same structure, but we need different gadgets and slight changes in the analysis to deal with each of the graphs H .

4.1 The general construction

In order to prove Theorems 3 and 5, we present several reductions from $k \times k$ PERMUTATION INDEPENDENT SET, introduced by Lokshantov et al. [14].

$k \times k$ PERMUTATION INDEPENDENT SET

Input: An integer k and a graph G with vertex set $[1, k] \times [1, k]$.

Parameter: k .

Output: Is there an independent set of size k in G with exactly one element from each row and one element from each column?

► **Theorem 6** (Lokshantov et al. [14]). *The $k \times k$ PERMUTATION INDEPENDENT SET problem cannot be solved in time $2^{o(k \log k)}$ unless the ETH fails.*

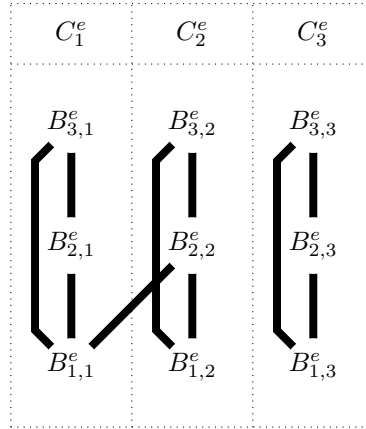
The general construction that we proceed to present only depends on the number of vertices of H . Let $H \in \mathcal{Q} \cup \mathcal{R} \cup \{\{\text{cricket}\}, \{\text{px}\}, \{\text{butterfly}\}, \{\text{co-banner}\}, \{\text{bull}\}, \{\text{kite}\}, \{\text{dart}\}\}$ and let (G, k) be an instance of $k \times k$ PERMUTATION INDEPENDENT SET. As we are asking for an independent set that contains exactly one vertex in each row, we will assume w.l.o.g. that, for each $(i, j), (i, j')$ in $V(G)$, $\{(i, j), (i, j')\} \in E(G)$. Let $n = |V(G)|$ and $m = |E(G)|$. We proceed to construct a graph F that displays the encoding of the m subgraphs of G consisting of exactly one edge. This is done in such a way that each edge is encoded exactly once. Moreover, these encodings are arranged in a cyclic way separated by gadgets ensuring the consistency of the selected solution.

Namely, we first define the graph $K = K_{h-1}$ where $h = |V(H)|$. For each $e \in E(G)$, and each $(i, j) \in [1, k]^2$, we define the graph $B_{i,j}^e$ to be the disjoint union of two copies of K and two new vertices $a_{i,j}^e$ and $b_{i,j}^e$. Informally, every graph $B_{i,j}^e$, $e \in E(G)$, plays the same role and corresponds to the vertex $(i, j) \in V(G)$. For each $e \in E(G)$ and each $j \in [1, k]$, we define the graph C_j^e obtained from the disjoint union of every $B_{i,j}^e$, $i \in [1, k]$, such that two graphs $B_{i_1,j}^e$ and $B_{i_2,j}^e$, $i_1 \neq i_2$, are complete to each other, i.e., for every $v_1 \in V(B_{i_1,j}^e)$ and $v_2 \in V(B_{i_2,j}^e)$, then $\{v_1, v_2\} \in E(C_j^e)$. Informally, every graph C_j^e , $e \in E(G)$, plays the same role and corresponds to the column j of G .

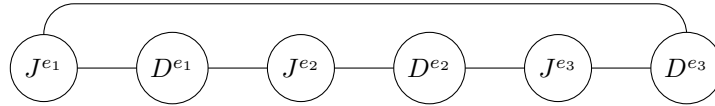
For every $e \in E(G)$, we also define the graph D^e obtained from the disjoint union of every C_j^e , $j \in [1, k]$, by adding, if $e = \{(i, j), (i', j')\}$, every edge $\{v_1, v_2\}$ such that $v_1 \in V(B_{i,j}^e)$ and $v_2 \in V(B_{i',j'}^e)$. The graph D^e is depicted in Figure 2. Informally, the graph D^e , $e \in E(G)$, encodes the edge e of the graph G .

For every $e \in E(G)$, we also define J^e such that $V(J^e) = \{c_j^e \mid j \in [1, k]\} \cup \{r_i^e \mid i \in [1, k]\}$ is a set of new vertices and $E(J^e) = \emptyset$. The graphs J^e , $e \in E(G)$, are the separators that will ensure the consistency of the selected solution.

Finally, the graph F is obtained from the disjoint union of every D^e , $e \in E(G)$, and every J^e , $e \in E(G)$, by adding the following edges, for a given fixed cyclic permutation



■ **Figure 2** The graph D^e for $e = \{(1, 1), (2, 2)\} \in E(G)$ where $k = 3$. A bold edge means that two graphs B are connected in a complete bipartite way.



■ **Figure 3** The shape of the framework graph F assuming that $k = 3$, G contains only the three edges e_1, e_2 , and e_3 , and σ is the cyclic permutation (e_1, e_2, e_3) .

σ of the elements of $E(G)$: for each $e \in E(G)$ and each $i, j \in [1, k]$, we add the edges $\{a_{i,j}^e, r_i^e\}$, $\{r_i^e, b_{i,j}^{\sigma^{-1}(e)}\}$, $\{b_{i,j}^{\sigma^{-1}(e)}, c_j^e\}$, and $\{c_j^e, a_{i,j}^e\}$ to $E(F)$. This concludes the definition of the framework graph F , which is depicted in Figure 3 (a similar figure appears in [4]). The pair $(F, \ell := 2h(k - 1)km)$ is called the H -framework of (G, k) . For convenience, we always assume that we know the permutation σ linked to the graph F .

Let us now discuss about the treewidth of F . First note that for each $e \in E(G)$, the set $V(J^e) \cup V(J^{\sigma(e)})$ disconnects the vertex set $V(D^e)$ from the remaining part of F . Moreover, if $e = \{(i, j), (i', j')\}$, then the bags $V(C_1^e) \cup V(B_{i,j}^e) \cup V(B_{i',j'}^e)$, $V(C_2^e) \cup V(B_{i,j}^e) \cup V(B_{i',j'}^e), \dots, V(C_k^e) \cup V(B_{i,j}^e) \cup V(B_{i',j'}^e)$ form a path decomposition of D^e of width $2h(k + 2) - 1$. Combining this decomposition with the circular shape of F and the fact that $V(J^e) \cup V(J^{\sigma(e)})$ disconnects the vertex set $V(D^e)$ from the remaining part of F , we obtain that the treewidth (in fact, also the pathwidth) of F is at most $6k + 2h(k + 2) - 1$, and therefore $\text{tw}(F) = \mathcal{O}(k)$.

For each graph H , we will consider (F, ℓ) , the H -framework of (G, k) , and create another pair (F_H, ℓ) , where F_H is a graph obtained starting from F by adding some vertices and edges. We will claim that there exists a solution of $k \times k$ PERMUTATION INDEPENDENT SET on (G, k) if and only if there exists a solution of $\{H\}$ -M-DELETION on (F_H, ℓ) . In order to do this, we will prove the two following properties for each graph H .

► **Property 1.** *Let S be a solution of $\{H\}$ -M-DELETION on (F_H, ℓ) . For every $e \in E(G)$ and $j \in [1, k]$ such that $|V(C_j^e) \setminus S|$ is maximized, there exists $i \in [1, k]$ such that $V(C_j^e) \setminus S \subseteq V(B_{i,j}^e)$.*

The above property states that for each column C_j^e , $j \in [1, k]$ and $e \in E(G)$, containing a minimum number of vertices of the solution, the remaining vertices all belong to the same row.

► **Property 2.** Let S be a solution of $\{H\}$ -M-DELETION on (F_H, ℓ) . For every $e \in E(G)$, and for every $i, j \in [1, k]$, if $b_{i,j}^e \notin S$, then for every $i' \in [1, k] \setminus \{i\}$, we have $a_{i',j}^{\sigma(e)} \in S$.

The above property states that the choices of the vertices $a_{i,j}^e, b_{i,j}^e$ are consistent through the whole framework graph F_H .

If we assume that Property 1 holds, we have the following lemma.

► **Lemma 7.** If Property 1 holds, then for every solution S of $\{H\}$ -M-DELETION on $(F_H, \ell = 2h(k-1)km)$, for every $e \in E(G)$, and for every $j \in [1, k]$, there exists $i \in [1, k]$ such that $V(C_j^e) \setminus S = V(B_{i,j}^e)$. Moreover, for every $e \in E(G)$, $V(J^e) \cap S = \emptyset$.

Proof. Assume that Property 1 holds and let S be a solution of $\{H\}$ -M-DELETION on $(F_H, 2h(k-1)km)$. By Property 1, we know that for every $e \in E(G)$, and for every $j \in [1, k]$, $|V(C_j^e) \cap S| \geq 2h(k-1)$. As there are exactly m edges and k columns, the budget is tight and we obtain that $|V(C_j^e) \cap S| = 2h(k-1)$. This implies that $|V(C_j^e) \setminus S| = 2h$, corresponding to the size of a set $B_{i,j}^e$ for some $i \in [1, k]$. The lemma follows. ◀

For each specific graph H , in order to prove Property 1 and Property 2, we will first prove Property 1 and then use Lemma 7 in order to prove Property 2.

► **Lemma 8** (\star). If Property 1 and Property 2 hold and there exists a solution S of $\{H\}$ -M-DELETION on $(F_H, \ell = 2h(k-1)km)$, then, for any $e \in E(G)$, the set $T^e = \{(i, j) \mid V(B_{i,j}^e) \cap S = \emptyset\}$ is a solution of $k \times k$ PERMUTATION INDEPENDENT SET on (G, k) .

Using the same argumentation, we obtain the same results for the topological minor version.

► **Property 3.** Let S be a solution of $\{H\}$ -TM-DELETION on (F_H, ℓ) . For every $e \in E(G)$ and $j \in [1, k]$ such that $|V(C_j^e) \setminus S|$ is maximized, there exists $i \in [1, k]$ such that $V(C_j^e) \setminus S \subseteq V(B_{i,j}^e)$.

► **Property 4.** Let S be a solution of $\{H\}$ -TM-DELETION on (F_H, ℓ) . For every $e \in E(G)$, and for every $i, j \in [1, k]$, if $b_{i,j}^e \notin S$, then for every $i' \in [1, k] \setminus \{i\}$, we have $a_{i',j}^{\sigma(e)} \in S$.

► **Lemma 9.** If Property 3 holds, then for every solution S of $\{H\}$ -TM-DELETION on $(F_H, \ell = 2h(k-1)km)$, for every $e \in E(G)$, and for every $j \in [1, k]$, there exists $i \in [1, k]$ such that $V(C_j^e) \setminus S = V(B_{i,j}^e)$. Moreover, for every $e \in E(G)$, $V(J^e) \cap S = \emptyset$.

► **Lemma 10.** If Property 3 and Property 4 hold and there exists a solution S of $\{H\}$ -TM-DELETION on $(F_H, \ell = 2h(k-1)km)$, then, for any $e \in E(G)$, the set $T^e = \{(i, j) \mid V(B_{i,j}^e) \cap S = \emptyset\}$ is a solution of $k \times k$ PERMUTATION INDEPENDENT SET on (G, k) .

Given a solution T of $k \times k$ PERMUTATION INDEPENDENT SET on (G, k) , we define $S_T = \{v \in V(F_H) \mid v \in B_{i,j}^e : e \in E(G), (i, j) \in [1, k]^2 \setminus T\}$. Note that $|S_T| = 2h(k-1)km$.

4.2 The reduction for P_5

We are ready to present the hardness reduction when $H = P_5$.

► **Theorem 11.** $\{P_5\}$ -M-DELETION cannot be solved in time $2^{o(\text{tw} \log \text{tw})} \cdot n^{\mathcal{O}(1)}$ unless the ETH fails.

Proof. Let $H = P_5$. Let (G, k) be an instance of $k \times k$ PERMUTATION INDEPENDENT SET and let (F, ℓ) be the H -framework of (G, k) , as defined in Section 4.1. Note that $\ell = 10(k-1)km$. In this theorem, we define $F_{P_5} = F$ without any modification.

Let T be a solution of $k \times k$ PERMUTATION INDEPENDENT SET on (G, k) . One can check that every connected component of $F \setminus S_T$ is of size 4. Indeed, the connected components of $F \setminus S_T$ are either the copies of the graph K , which is of size 4, or the subgraph induced by the edges $\{a_{i,j}^e, r_i^e\}$, $\{r_i^e, b_{i,j}^{\sigma^{-1}(e)}\}$, $\{b_{i,j}^{\sigma^{-1}(e)}, c_j^e\}$, and $\{c_j^e, a_{i,j}^e\}$, for every $(i, j) \in T$. Thus $F \setminus S_T$ does not contain any P_5 as a minor and S_T is a solution of $\{P_5\}$ -M-DELETION of size $10(k-1)km$.

Assume now that S is a solution of $\{P_5\}$ -M-DELETION on G of size $10(k-1)km$. We first prove that Property 1 holds. Let $e \in E(G)$ and $j \in [1, k]$ that maximize the size of $|V(C_j^e) \setminus S|$. By the pigeonhole principle, $|V(C_j^e) \setminus S| \geq 10$. Let U_M be a set $V(B_{i,j}^e) \setminus S$, $i \in [1, k]$, with the maximum number of elements, and let $U_A = V(C_j^e) \setminus (S \cup U_M)$. If $|U_M| = 1$, then, as $|V(C_j^e) \setminus S| \geq 10$, we obtain that K_{10} is a subgraph of C_j^e , contradicting the definition of S . If $|U_M| \geq 2$ and $|U_A| \geq 3$ or if $|U_M| \geq 3$ and $|U_A| \geq 2$, then C_j^e contains a $K_{2,3}$ as a subgraph, also contradicting the definition of S . Finally if $|U_A| = 1$, we have that $|U_M| = 9$ and so, $C_j^e[U_M]$ contains a K_4 that, combined with the element of U_A , produces a K_5 that is forbidden by the definition of S . Thus $|U_A| = 0$ and Property 1 holds.

Let $e \in E(G)$ and let $i, j \in [1, k]$ such that $b_{i,j}^e \notin S$. Let $i' \in [1, k]$ such that $i \neq i'$. If $a_{i',j}^{\sigma(e)} \notin S$, then, as by Lemma 7 $S \cap V(J^{\sigma(e)}) = \emptyset$, we have that the path $r_{i'}^{\sigma(e)}, a_{i',j}^{\sigma(e)}, c_j^{\sigma(e)}, b_{i,j}^e, r_i^e$ is a subgraph of $F \setminus S$. Since, by definition of S , $F \setminus S$ does not contain P_5 as a minor, we have that $a_{i',j}^{\sigma(e)} \in S$. Thus Property 2 holds and the theorem follows. \blacktriangleleft

4.3 The reduction for $K_{1,s}$

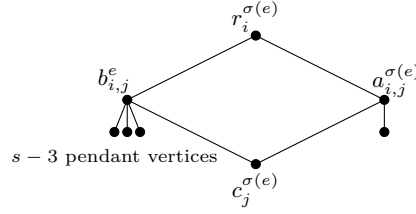
The next theorem should be compared to the result in the full version stating that there exist single-exponential algorithms for hitting $K_{1,s}$ as a *topological* minor for every $s \geq 1$, while in Theorem 12 we prove that it is not the case for hitting $K_{1,s}$ as a minor, for every $s \geq 4$. It should be noted that the bound on s of Theorem 12 is tight, as if $s \leq 3$, then $\{K_{1,s}\}$ -M-DELETION is exactly $\{K_{1,s}\}$ -TM-DELETION, and therefore it can be solved in single-exponential time; see the full version for the details.

► **Theorem 12.** *Given $s \geq 4$, $\{K_{1,s}\}$ -M-DELETION cannot be solved in time $2^{o(\text{tw} \log \text{tw})} \cdot n^{\mathcal{O}(1)}$ unless the ETH fails.*

Proof. Let $s \geq 4$ and let $H = K_{1,s}$. Let (G, k) be an instance of $k \times k$ PERMUTATION INDEPENDENT SET and let (F, ℓ) be the H -framework of (G, k) , as defined in Section 4.1. Note that $\ell = 2(s+1)(k-1)km$. We construct the graph F_H from F by adding a pendant vertex to every vertex $a_{i,j}^e$, $e \in E(G)$, $i, j \in [1, k]$, and by adding $s-3$ pendant vertices to every vertex $b_{i,j}^e$, $e \in E(G)$, $i, j \in [1, k]$.

Let T be a solution of $k \times k$ PERMUTATION INDEPENDENT SET on (G, k) . Then every connected component of $F_H \setminus S_T$ is either a copy of the graph K , which is of size s , or the subgraph induced by $a_{i,j}^e, r_i^e, b_{i,j}^{\sigma^{-1}(e)}, c_j^e$, and the vertices that are pendant to $a_{i,j}^e$ and $b_{i,j}^{\sigma^{-1}(e)}$, for every $(i, j) \in T$. This latter subgraph, depicted in Figure 4, does not contain $K_{1,s}$ as a minor. Thus $F_H \setminus S_T$ does not contain any $K_{1,s}$ as a minor and S_T is a solution of $\{K_{1,s}\}$ -M-DELETION of size $2(s+1)(k-1)km$.

Assume now that S is a solution of $\{K_{1,s}\}$ -M-DELETION on G of size $2(s+1)(k-1)km$. We first prove that Property 1 holds. Let $e \in E(G)$ and $j \in [1, k]$ that maximize the size of



■ **Figure 4** A connected component of $F_H \setminus S$ that is not a copy of K , with $s = 6$.

$|V(C_j^e) \setminus S|$. By the pigeonhole principle, $|V(C_j^e) \setminus S| \geq 2(s+1)$. Let U_M be a set $V(B_{i,j}^e) \setminus S$, $i \in [1, k]$, with the maximum number of elements, and let $U_A = V(C_j^e) \setminus (S \cup U_M)$. If $1 \leq |U_M| < s$, then $|U_A| \geq s$ and $K_{1,s}$ is a subgraph of C_j^e , contradicting the definition of S . If $|U_M| \geq s$ and $|U_A| \geq 1$, then again $K_{1,s}$ is a subgraph of C_j^e , contradicting again the definition of S . Thus $|U_A| = 0$ and Property 1 holds.

Let $e \in E(G)$ and let $i, j \in [1, k]$ such that $b_{i,j}^e \notin S$. Let $i' \in [1, k]$ such that $i \neq i'$. If $a_{i',j}^{\sigma(e)} \notin S$, then, as by Lemma 7 it holds that $S \cap V(J^{\sigma(e)}) = \emptyset$, we have that the path $r_{i'}^{\sigma(e)}, a_{i',j}^{\sigma(e)}, c_j^{\sigma(e)}, b_{i,j}^e, r_i^{\sigma(e)}$ combined with the vertex pendant to $a_{i',j}^{\sigma(e)}$ and the $s - 3$ vertices pendant to $b_{i,j}^e$ is a subgraph of $F_H \setminus S$ and $K_{1,s}$ is a minor of it. As, by definition of S , $F_H \setminus S$ does not contain $K_{1,s}$ as a minor, we have that $a_{i',j}^{\sigma(e)} \in S$. Thus Property 2 holds and the theorem follows. ◀

5 Conclusions and further research

The ultimate goal in this line of research is to establish the *tight* complexity of $\{H\}$ -M-DELETION and $\{H\}$ -TM-DELETION for any graph H , but we are still very far from it. In particular, we do not know whether there exists some H for which a double-exponential lower bound can be proved. Very recently, Kociumaka and Pilipczuk [13] studied the problem of deleting a minimum number of vertices to obtain a graph of Euler genus at most g , and presented an algorithm running in time $2^{\mathcal{O}_g(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$. Generalizing their technique to H -minor-free graphs (which would correspond to the general $\{H\}$ -M-DELETION problem) seems quite challenging, as this would involve a huge amount of technical details.

We managed to classify the complexity of $\{H\}$ -M-DELETION when H is a connected planar graph on at most 5 vertices (cf. Figure 1). While we consider this dichotomy a significant result, most of the algorithms and the reductions are ad hoc, and therefore our approach does not seem to be easily applicable for dealing with larger graphs H . The case where H is planar and connected is already very interesting, since the results in [1] imply that $\{H\}$ -M-DELETION cannot be solved in time $2^{\mathcal{O}(\text{tw})} \cdot n^{\mathcal{O}(1)}$ under the ETH and can be solved in time $2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$. Thus, it makes sense to guess that, in this case, the complexity of $\{H\}$ -M-DELETION is either $2^{\mathcal{O}(\text{tw})} \cdot n^{\mathcal{O}(1)}$ or $2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$, as it happens if $|V(H)| \leq 5$. We conjecture that for every connected simple planar graph H with $|V(H)| \geq 6$, the latter case holds. In fact, we conjecture the following property, which is easily seen to imply the previous conjecture: if H and H' are graphs such that $H \preceq_m H'$ and $\{H\}$ -M-DELETION is not solvable under the ETH in time $f(\text{tw}) \cdot n^{\mathcal{O}(1)}$ for some function f , then $\{H'\}$ -M-DELETION is not solvable under the ETH in time $f(\text{tw}) \cdot n^{\mathcal{O}(1)}$ either. We think that the equivalent property for the topological minor version also holds. Note that for establishing a dichotomy for $\{H\}$ -TM-DELETION when H is a connected planar graph on at most 5 vertices, it remains to obtain algorithms in time $2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$ for the graphs in Figure 1 that have maximum degree 4, like the gem or the dart, as for those graphs [1, Theorem 6] cannot be applied.

Finally, note that the only connected (simple) graph on at most 5 vertices missing in Figure 1 is K_5 . We think that, using techniques similar as those developed in [11], $\{K_5\}$ -M-DELETION is solvable in time $2^{\mathcal{O}(\text{tw} \cdot \log \text{tw})} \cdot n^{\mathcal{O}(1)}$, which would be tight.

References

- 1 Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Optimal Algorithms for Hitting (Topological) Minors on Graphs of Bounded Treewidth. In *Proc. of the 12th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 89 of *LIPICs*, pages 4:1–4:12, 2017.
- 2 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86–111, 2015.
- 3 Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshantov, and Michal Pilipczuk. A $c^k n$ 5-Approximation Algorithm for Treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016.
- 4 Édouard Bonnet, Nick Brettell, O-joung Kwon, and Dániel Marx. Generalized Feedback Vertex Set Problems on Bounded-Treewidth Graphs: Chordality Is the Key to Single-Exponential Parameterized Algorithms. In *Proc. of the 12th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 89 of *LIPICs*, pages 7:1–7:13, 2017.
- 5 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 6 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving Connectivity Problems Parameterized by Treewidth in Single Exponential Time. In *Proc. of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 150–159, 2011.
- 7 Reinhard Diestel. *Graph Theory*, volume 173. Springer-Verlag, 4th edition, 2010.
- 8 R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- 9 Fedor V. Fomin, Daniel Lokshantov, Fahad Panolan, and Saket Saurabh. Efficient Computation of Representative Families with Applications in Parameterized and Exact Algorithms. *Journal of the ACM*, 63(4):29:1–29:60, 2016.
- 10 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.
- 11 Bart M. P. Jansen, Daniel Lokshantov, and Saket Saurabh. A Near-Optimal Planarization Algorithm. In *Proc. of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1802–1811, 2014.
- 12 T. Kloks. *Treewidth. Computations and Approximations*. Springer-Verlag LNCS, 1994.
- 13 Tomasz Kociumaka and Marcin Pilipczuk. Deleting vertices to graphs of bounded genus. *CoRR*, abs/1706.04065, 2017. [arXiv:1706.04065](https://arxiv.org/abs/1706.04065).
- 14 Daniel Lokshantov, Dániel Marx, and Saket Saurabh. Slightly Superexponential Parameterized Problems. In *Proc. of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 760–776, 2011.
- 15 Marcin Pilipczuk. A tight lower bound for Vertex Planarization on graphs of bounded treewidth. *Discrete Applied Mathematics*, 231:211–216, 2017.