A Complexity Dichotomy for Hitting Small Planar Minors Parameterized by Treewidth
Julien Baste, Ignasi Sau Valls, Dimitrios M. Thilikos

To cite this version:

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

Dimitrios M. Thilikos
LIRMM, CNRS, Université de Montpellier, Montpellier, France, and
Department of Mathematics, National and Kapodistrian University of Athens, Greece
sedthilk@thilikos.info

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 \( \text{tw} \) 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 \( 2^{O(\text{tw} \log \text{tw})} \cdot n^{O(1)} \). 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 \( 2^{\Theta(\text{tw} \log \text{tw})} \cdot n^{O(1)} \). 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 \( 2^{\Theta(\text{tw} \log \text{tw})} \cdot n^{O(1)} \), 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 → Graph algorithms, Theory of computation → 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.
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 \text{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 $2^{O(\text{tw})} \cdot n^{O(1)}$, called single-exponential, by standard dynamic-programming techniques, and no algorithm with running time $2^{o(\text{tw})} \cdot n^{O(1)}$ 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-Decide technique. See also $[2, 9, 11, 15]$.

We recently studied these problems in $[1]$ and proved$^2$, among other results, that if $H$ is planar and connected, then the problems cannot be solved in time $2^{o(\text{tw})} \cdot n^{O(1)}$ under the ETH, and can be solved in time $2^{O(\text{tw} \cdot \log \text{tw})} \cdot n^{O(1)}$ (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, \{\begin{small}$K_{1,i}$\end{small}\}-M-Deletion is solvable in time $2^{O(\text{tw})} \cdot n^{O(1)}$ for $i \leq 3$, and in time $2^{O(\text{tw} \cdot \log \text{tw})} \cdot n^{O(1)}$ for $i \geq 4$. On the other hand, \{\begin{small}$K_{1,i}$\end{small}\}-TM-Deletion can be solved in time $2^{O(\text{tw})} \cdot n^{O(1)}$ 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 \{\begin{small}$H$\end{small}\}-M-Deletion and \{\begin{small}$H$\end{small}\}-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 $2^{O(\text{tw})} \cdot n^{O(1)}$ for $i \leq 2$, and in time $2^{O(\text{tw} \cdot \log \text{tw})} \cdot n^{O(1)}$ for $i \geq 3$. The same dichotomy occurs when $H = K_{2,i}$.

3. We classify the optimal asymptotic complexity of \{\begin{small}$H$\end{small}\}-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 $2^{O(\text{tw})} \cdot n^{O(1)}$, and that the other 20 ones are solvable in time $2^{O(\text{tw} \cdot \log \text{tw})} \cdot n^{O(1)}$; 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 $2^{O(\text{tw} \cdot \log \text{tw})} \cdot n^{O(1)}$, and that the chair and the banner are the only graphs on 5 vertices for which the problem is solvable in time $2^{O(\text{tw})} \cdot n^{O(1)}$. In particular, this settles the complexity of \{\begin{small}$P_5$\end{small}\}-M-Deletion, which we left open in item 1.

All the lower bounds also hold for \{\begin{small}$H$\end{small}\}-TM-Deletion, with the only difference that the case $H = K_{1,4}$ can be solved in time $2^{O(\text{tw})} \cdot n^{O(1)}$, 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.
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 $\theta_i$ 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

---

**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 $2^{\Theta(tw)} \cdot n^{O(1)}$ (resp. $2^{\Theta(tw \cdot \log tw)} \cdot n^{O(1)}$). 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].
on the right hand side of Figure 1, namely $K_4$, $C_5$, $K_3 \cup 2K_1$, $K_5$-e, $W_4$, $P_3 \cup 2K_1$, $P_2 \cup P_3$, 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 $\mathcal{G} = \{G_t | t \in V(T)\}$ 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 insert vertex node and the vertex $v_{\text{insert}}$ 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 $v_{\text{forget}}$ 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 $2^{O(tw)} \cdot n$ a tree decomposition of width $O(tw)$ 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 $w = O(tw)$ 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. $H \preceq_{tm} G$). 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. $H \not\preceq_{tm} G \setminus S$).

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 \texttt{paw} $\leq \texttt{tm}$ $G$ 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 \texttt{paw} as a topological minor. Let $G$ be a graph such that \texttt{paw} $\leq \texttt{tm}$ $G$. 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 \texttt{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 \texttt{(paw)-TM-Deletion}. As the algorithm that we presented for \{\texttt{C}_1\}-\texttt{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 $\texttt{tm}_{\texttt{paw}}(G) \leq k$.

On the other hand, if such a partition does not exist, we know that $\texttt{tm}_{\texttt{paw}}(G) > k$. The main idea of this algorithm is to combine classical dynamic programming techniques in order to verify that $F$ induces a forest. As for \{\texttt{C}_1\}-\texttt{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 $H/Z_1, Y$ 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 $(T, X, r, G)$ 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 $\ell$, each function $s : X_t \to \{0, 1, 2, 3, 4\}$, each function $s_0 : \{v_0\} \times s^{-1}(1) \to \{0, 1\}$, and each partition $p \in \Pi(s^{-1}(1))$, we define:

$$
\mathcal{E}_t(p, s, s_0, i, j, \ell) = \{(Z_1, Z_2, Y) \mid (Z_1, Z_2, Y) \in 2^{V_t} \times 2^{V_t} \times 2^{E_0 \cup E(G_t)}, Z_1 \cap Z_2 = \emptyset, \ Z_1 = i, \ Z_2 = \ell, \ |E(G_t[Z_1 \setminus \{v_0\}]) \cup Y| = j, \ \forall e \in E_0 \cap E_t, s_0(e) = 1 \Leftrightarrow e \in Y, \ \forall v \in Z_2 \cap X_t, s(v) = 2 \Leftrightarrow s = \deg_{G_t[Z_2]}(v), \ \forall v \in Z_2 \setminus X_t, \deg_{G_t[Z_2]}(v) = 2, \ \forall u \in Z_1 \setminus X_t : \text{either } t \text{ is the root or } \exists u' \in s^{-1}(1) : u \text{ and } u' \text{ are connected in } G_t[Z_1, Y], \ \forall v_1, v_2 \in s^{-1}(1) : p \subseteq V_t[[v_1, v_2]] \Leftrightarrow v_1 \text{ and } v_2 \text{ are connected in } G_t[Z_1, Y], \ \forall (u, v) \in (Z_1 \setminus \{v_0\}) \times Z_2, \ u, v \notin E(G_t))
$$

$$
\mathcal{A}_t(s, s_0, i, j, \ell) = \{p \mid p \in \Pi(s^{-1}(1)), \mathcal{E}_t(p, s, s_0, i, j, \ell) \neq \emptyset\}.
$$
In the definition of $E_t$, 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 $\ell$, $i + \ell \geq |V(G)| \cup \{v_0\} - k$ and $A_t(\emptyset, i, i - 1, \ell) \neq \emptyset$. For each $t \in V(T)$, we assume that we have already computed $A_t$ for every children $t'$ of $t$, and we proceed to the computation of $A_t$. As usual, we distinguish several cases depending on the type of node $t$.

**Leaf.** By definition of $A_t$, we have $A_t(\emptyset, \emptyset, 0, 0, 0) = \emptyset$.

**Introduce vertex.** Let $v$ be the insertion vertex of $X_t$, let $t'$ be the child of $t$, let $s : X_t \rightarrow \{0, 1, 2, 0, 1, 2\}$, $s_0 : \{v_0\} \times s^{-1}(1) \rightarrow \{0, 1\}$, and let $H = G_t(s^{-1}(1), s_0^{-1}(1))$.

- If $v = v_0$ and $s(v_0) \in \{0, 2, 1, 2\}$, then by definition of $A_t$ we have that $A_t(s, s_0, i, j, \ell) = \emptyset$.
- Otherwise, if $v = v_0$, then by construction of the nice tree decomposition, we know that $t'$ is a leaf of $T$ and so $s = \{(v_0, 1)\}$, $j = \ell = i = 0$ and $A_t(s, s_0, i, j, \ell) = \text{ins}(\{v_0\}, A_t(\emptyset, \emptyset, 0, 0, 0))$.
- Otherwise, if $s(v) = 0$, then, by definition of $A_t$, it holds that $A_t(s, s_0, i, j, \ell) = A_t(s, s_0, i, j, \ell)$.
- Otherwise, if $s(v) = 2$, $z \in \{0, 1, 2\}$, then $Z_2' = N_{G_t}[X_t](v) \setminus s^{-1}(0)$. If $Z_2' \not\subset s^{-1}(\{2, 1, 2\})$ or $|Z_2'| \neq z$, then $A_t(s, s_0, i, j, \ell) = \emptyset$. Otherwise $Z_2' \subset s^{-1}(\{2, 1, 2\})$ and $|Z_2'| = z$, and with $s' : X_{v'} \rightarrow \{0, 1, 2, 0, 1, 2\}$ defined such that $\forall v' \in X_v, Z_2', s'(v') = s(v')$ and for each $v' \in Z_2'$ such that $s'(v') = 2$, $z' \in \{1, 2\}$, $s'(v') = 2z' - 1$. It holds that $A_t(s, s_0, i, j, \ell) = A_t(s', s_0, i, j, \ell - 1)$.
- Otherwise, we know that $v \neq v_0$, $s(v) = 1$, and $v_0 \in N_{G_t}[s^{-1}(1)](v)$. First, if $N_{G_t}[X_t](v) \setminus s^{-1}(0) \subset s^{-1}(1)$, then $A_t(s, s_0, i, j, \ell) = \emptyset$. Indeed, this implies that the cycle part and the forest part are connected. As $s(v) = 1$, 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 $s_0$. By adding $v$, we add one vertex and $|N_H(v)|$ edges in the forest part. Therefore, we have that $A_t(s, s_0, r, i, j, \ell) = \text{glue}(N_H[v], \text{ins}(\{v\}, A_t(s, s_0, r, i, j, \ell) - |N_H(v)|))$.

**Forget vertex.** Let $v$ be the forget vertex of $X_t$, let $t'$ be the child of $t$, and let $s : X_t \rightarrow \{0, 1, 2, 0, 1, 2\}$. As a vertex from the collection of cycles can be removed only if it has exactly two neighbors, we obtain that $A_t(s, i, j, \ell) = A_t(s, \emptyset \cup \{(v, 0)\}, s_0, i, j, \ell)$.

**Join.** Let $t'$ and $t''$ be the two children of $t$, let $s : X_t \rightarrow \{0, 1, 2, 0, 1, 2\}$, $s_0 : \{v_0\} \times s^{-1}(1) \rightarrow \{0, 1\}$, and let $H = G_t(s^{-1}(1), s_0^{-1}(1))$. Given three functions $s^*, s', s'' : X_t \rightarrow \{0, 1, 2, 0, 1, 2\}$, we say that $s^* = s' \oplus s''$ if for each $v \in s^{-1}(\{0, 1\})$, $s^*(v) = s'(v)$, and for each $v \in X_t$ such that $s'(v) = 2$, $z \in \{0, 1, 2\}$, there exist $z', z'' \in \{0, 1, 2\}$ such that $s''(v) = 2z'$, $s'(v) = 2z''$, and $z = z' + z'' - \deg_{G_t}(X_t)(s^{-1}(0))(v)$. 


We join every compatible entries \( A_r(s', s_0') = \bigcup \text{join}(A_r(s', s_0', i', j', \ell'), A_r(s'', s_0'', i'', j'', \ell'')) \).

\[ s', s'' : \forall i \in \{0, 1, 2, 3, 4\}, \]
\[ i + i' = i + |V(H)|, \]
\[ j + j'' = j + |E(H)|, \]
\[ \ell + \ell'' = \ell + |s\{(20, 21, 22)\}|. \]

\textbf{Theorem 2.} \textit{[paw]}-TM-Deletion can be solved in time \( 2^{O(tw)} \cdot n^7 \).

\textbf{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 \( A_r(M) \), we store \( A_r(M) = \text{reduce}(A_r(M)) \) 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 \( M, A_r(M) \) represents \( A_r(M) \). In particular, we have that \( A_r(M) = \text{reduce}(A_r(M)) \) for each possible entry \( M \). Using the definition of \( A_r \) and Lemma 1, we have that \( \text{tm}(\text{paw})(G) \leq k \) if and only if for some \( i \) and \( t, i + \ell \geq |V(G) \cup \{v_0\}| - k \) and \( A_r(\emptyset, i, i - 1, \ell) \neq \emptyset \).

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 \( 2^{2^{\mathcal{g}_1}}(1) \). For a forget vertex node, we take the union of four sets of size \( 2^{2^{\mathcal{g}_1}}(1) \), so the intermediate sets of weighted partitions have size at most \( 4 \cdot 2^{2^{\mathcal{g}_1}}(1) \). For a join node, as in the big union operation we take into consideration at most \( 5|X_1| \) possible functions \( s' \), as many functions \( s'' \), at most \( n + |s^{-1}(1)| \) choices for \( i' \) and \( i'' \), at most \( n + |s^{-1}(1)| \) choices for \( j' \) and \( j'' \) (as we can always assume, during the algorithm, that \( H \) is a forest), and at most \( n + |s^{-1}((20, 21, 22))| \) choices for \( \ell' \) and \( \ell'' \), we obtain that the intermediate sets of weighted partitions have size at most \( 25|X_2| \cdot (n + |s^{-1}(1)|)^2 \cdot (n + |s^{-1}((20, 21, 22))|) \cdot 4^{2^{\mathcal{g}_1}}(1) \). We obtain that the intermediate sets of weighted partitions have size at most \( n + |X_1|^3 \cdot 100|X_1|^3 \).

Moreover, for each node \( t \in V(T) \), the function \text{reduce} will be called as many times as the number of possible entries, i.e., at most \( 2^{|w|} \cdot n^3 \) times. Thus, using [2, Theorem 3.7], \( A_r \) can be computed in time \( 2^{O(w)} \cdot n^6 \). The theorem follows by taking into account the linear number of nodes in a nice tree decomposition.

\section{Superexponential lower bounds}

Let \( Q = \{P_3\} \cup \{K_{2,s} | s \geq 3\} \cup \{\theta_s | s \geq 3\} \) and let \( R = \{K_{1,s} | s \geq 4\} \). In this section, we provide superexponential lower bounds for \{H\}-M-Deletion when \( H \in Q \cup R \), for \{H\}-TM-Deletion when \( H \in Q \), 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.

\textbf{Theorem 3.} Let \( H \in Q \cup R \). Unless the ETH fails, \{H\}-M-Deletion cannot be solved in time \( 2^{o(tw \log tw)} \cdot n^{O(1)} \).

\textbf{Theorem 4.} Let \( H \in Q \). Unless the ETH fails, \{H\}-TM-Deletion cannot be solved in time \( 2^{o(tw \log tw)} \cdot n^{O(1)} \).

\textbf{Theorem 5.} Let \( H \in \{\text{cricket}, \{px\}, \{butterfly\}, \{co-banner\}, \{bull\}, \{kite\}, \{dart\}\} \). Unless the ETH fails, neither \( \{H\}\)-M-Deletion nor \{H\}-TM-Deletion can be solved in time \( 2^{o(tw \log tw)} \cdot n^{O(1)} \).
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 “topological minor” by “topological minor” and “-M-Deletion” by “-TM-Deletion”. Note that this holds when H ∈ Q but not when H ∈ R.

We first provide in Section 4.1 a general framework that will be used for every H ∈ Q ∪ 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 P3 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 Lokshtanov 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 (Lokshtanov 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 $H \in Q \cup R \cup \{\text{cricket}, \{\text{px}\}, \{\text{butterfly}\}, \{\text{co-banner}\}, \{\text{bull}\}, \{\text{kite}\}, \{\text{dart}\}\}$ 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 $i_1 \neq i_2$, if $v_1 \in V(B_{i_1,j}^e)$ 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...
and we obtain that the treewidth (in fact, also the pathwidth) of \( 2 \) width \( V \) row.

For each graph \( G,k \) ∈ \( \mathcal{F} \), where \( J \) is a graph obtained starting from \( H \) and \( F,\ell \). Combining this decomposition with the circular shape of \( \sigma \) 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 \( V(C_{1}^{e}) \cup V(B_{1,1}^{e}) \cup V(B_{1,2}^{e}) \cup V(B_{1,3}^{e}) \cup V(C_{2}^{e}) \cup V(B_{2,1}^{e}) \cup V(B_{2,2}^{e}) \cup V(B_{2,3}^{e}) \cup V(C_{3}^{e}) \cup V(B_{3,1}^{e}) \cup V(B_{3,2}^{e}) \cup V(B_{3,3}^{e}) \) 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 \( tw(F) = O(k) \).

For each graph \( H \), we will consider \( (F,\ell) \), the \( H \)-framework of \( (G,k) \), and create another pair \( (F_H,\ell) \), 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 \( (F_H,\ell) \). 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 \( (F_H,\ell) \). 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 \( V(C_{i}^{e}) \setminus S \subseteq V(B_{i,j}^{e}) \).

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 $(F_H, \ell)$. For every $e \in E(G)$, and for every $i, j \in [1, k]$, if $b^e_{i,j} \not\in S$, then for every $i' \in [1, k] \setminus \{i\}$, we have $a^{(e)}_{i',j} \in S$.

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 $\tau \in E(G)$, and for every $j \in [1, k]$, there exists $i \in [1, k]$ such that $V(C^\tau_j) \setminus S = V(B^\tau_{i,j})$. Moreover, for every $\tau \in E(G), V(J^\tau) \cap S = \emptyset$.

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 $v \in E(G)$, and for every $j \in [1, k]$, $|V(C^\tau_j) \setminus S| \geq 2h(k-1)$. As there are exactly $m$ edges and $k$ columns, the budget is tight and we obtain that $|V(C^\tau_j) \cap S| = 2h(k-1)$. This implies that $|V(C^\tau_j) \setminus S| = 2h$, corresponding to the size of a set $B^\tau_{i,j}$ 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 ($\ast$). 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 $\tau \in E(G)$, the set $T^\tau = \{(i, j) | V(B^\tau_{i,j}) \cap S = \emptyset\}$ 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 $(F_H, \ell)$. For every $\tau \in E(G)$ and $j \in [1, k]$ such that $|V(C^\tau_j) \setminus S|$ is maximized, there exists $i \in [1, k]$ such that $V(C^\tau_j) \setminus S \subseteq V(B^\tau_{i,j})$.

Property 4. Let $S$ be a solution of $\{H\}$-TM-Deletion on $(F_H, \ell)$. For every $\tau \in E(G)$, and for every $i, j \in [1, k]$, if $b^e_{i,j} \not\in S$, then for every $i' \in [1, k] \setminus \{i\}$, we have $a^{(e)}_{i',j} \in S$.

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 $\tau \in E(G)$, and for every $j \in [1, k]$, there exists $i \in [1, k]$ such that $V(C^\tau_j) \setminus S = V(B^\tau_{i,j})$. Moreover, for every $\tau \in E(G), V(J^\tau) \cap S = \emptyset$.

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 $\tau \in E(G)$, the set $T^\tau = \{(i, j) | V(B^\tau_{i,j}) \cap S = \emptyset\}$ 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 $S_T = \{v \in V(F_H) | v \in B^e_{i,j} : e \in E(G), (i, j) \in [1, k]^2 \setminus T\}$. 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 $2^{o(tw log tw)} \cdot n^{O(1)}$ unless the ETH fails.
The next theorem should be compared to the result in the full version stating that there exist single-exponential algorithms for hitting \( K_{a,K} \) as a 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.

\begin{theorem}
Given \( s \geq 4 \), \( \{ K_{1,s} \} \)-M-Deletion cannot be solved in time \( 2^{o(tw \log tw)} \cdot n^{O(1)} \) unless the ETH fails.
\end{theorem}

\begin{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 \( (F,\ell) \) 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 \( F_{\emptyset} = F \) 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 \( \{ a_{i,j},e_{i,j}\}, \{ r_i^e,r_i^{e-1(e)}\}, \{ b_i^{e-1(e)},c_i^e\} \), and \( \{ c_j,a_{i,j}\} \), 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 \( |V(C^*_j) \setminus S| \). By the pigeonhole principle, \( |V(C^*_j) \setminus S| \geq 10 \). Let \( U_M \) be a set \( V(B^e_{i,j}) \setminus S \), \( i \in [1,k] \), with the maximum number of elements, and let \( U_A = V(C^*_j) \setminus (S \cup U_M) \). If \( |U_M| = 1 \), then, as \( |V(C^*_j) \setminus S| \geq 10 \), we obtain that \( K_{10} \) is a subgraph of \( C^*_j \), 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 \( C^*_j \) contains a \( K_2 \), as a subgraph, also contradicting the definition of \( S \). Finally if \( |U_A| = 1 \), we have that \( |U_M| = 9 \) and so, \( C^*_j \subset U_M \) 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, by Lemma 7 \( S \cap V(J^{\sigma(e)}) = \emptyset \), we have that the path \( \Sigma_{i'}^{e-}(e_{i,j} \sigma(e)), a_{i,j}^{\sigma(e)}, c_{i,j}^e, b_{i,j}^{e-}, r_i^e \) 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.
\end{proof}

\subsection{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.

\begin{theorem}
Given \( s \geq 4 \), \( \{ K_{1,s} \} \)-M-Deletion cannot be solved in time \( 2^{o(tw \log tw)} \cdot n^{O(1)} \) unless the ETH fails.
\end{theorem}

\begin{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 \( (F,\ell) \) 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 \( a_{i,j}^e, r_i^e, b_i^{e-1(e)}, c_i^e \), and the vertices that are pendant to \( a_{i,j}^e \) and \( b_i^{e-1(e)} \), 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
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, Koichi 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 \(2^\Theta(\text{tw}) \cdot n^{O(1)}\). 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 \(2^\Theta(\text{tw}) \cdot n^{O(1)}\) under the ETH and can be solved in time \(2^\Theta(\text{tw} \log \text{tw}) \cdot n^{O(1)}\). Thus, it makes sense to guess that, in this case, the complexity of \{H\}-M-Deletion is either \(2^\Theta(\text{tw}) \cdot n^{O(1)}\) or \(2^\Theta(\text{tw} \log \text{tw}) \cdot n^{O(1)}\), 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 \(f(\text{tw}) \cdot n^{O(1)}\) for some function \(f\), then \{H'\}-M-Deletion is not solvable under the ETH in time \(f(\text{tw}) \cdot n^{O(1)}\) 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 \(2^{O(\text{tw} \log \text{tw})} \cdot n^{O(1)}\) 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.

![Figure 4](image-url) A connected component of \(F_H \setminus S\) that is not a copy of \(K\), with \(s = 6\).
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 $2^{O(tw \log tw)} \cdot n^{O(1)}$, which would be tight.

References