Hitting forbidden subgraphs in graphs of bounded treewidth⋆

Marek Cygan[1], Dániel Marx[2], Marcin Pilipczuk[3], and Michał Pilipczuk[3]

[1] Institute of Informatics, University of Warsaw, Poland
cygan@mimuw.edu.pl
[2] Institute for Computer Science and Control,
Hungarian Academy of Sciences (MTA SZTAKI), Hungary
dmarx@cs.bme.hu
[3] Department of Informatics, University of Bergen, Norway
(Marcin.Pilipczuk|Michal.Pilipczuk)@ii.uib.no

Abstract. We study the complexity of a generic hitting problem H-Subgraph Hitting, where given a fixed pattern graph H and an input graph G, we seek for the minimum size of a set $X \subseteq V(G)$ that hits all subgraphs of G isomorphic to H. In the colorful variant of the problem, each vertex of G is precolored with some color from $V(H)$ and we require to hit only H-subgraphs with matching colors. Standard techniques (e.g., Courcelle's theorem) show that, for every fixed H and the problem is fixed-parameter tractable parameterized by the treewidth of G; however, it is not clear how exactly the running time should depend on treewidth. For the colorful variant, we demonstrate matching upper and lower bounds showing that the dependence of the running time on treewidth of G is tightly governed by $\mu(H)$, the maximum size of a minimal vertex separator in H. That is, we show for every fixed H that, on a graph of treewidth t, the colorful problem can be solved in time $2^{\mathcal{O}(t^{\mu(H)})} \cdot |V(G)|$, but cannot be solved in time $2^{o(t^{\mu(H)})} \cdot |V(G)|^{O(1)}$, assuming the Exponential Time Hypothesis (ETH). Furthermore, we give some preliminary results showing that, in the absence of colors, the parameterized complexity landscape of H-Subgraph Hitting is much richer.

1 Introduction

The "optimality programme" is a thriving trend within parameterized complexity, which focuses on pursuing tight bounds on the time complexity of parameterized problems. Instead of just determining whether the problem is fixed-parameter tractable, that is, whether the problem with a certain parameter k can be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some computable function $f(k)$, the goal is to determine the best possible dependence $f(k)$ on the parameter k. For several problems,

⋆ The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 267959, and n. 280152, as well as OTKA grant NK10564 and Polish National Science Centre grant DEC-2012/05/D/ST6/03214.

matching upper and lower bounds have been obtained for the function $f(k)$. The lower bounds are under the complexity assumption Exponential Time Hypothesis (ETH), which roughly states than n-variable 3SAT cannot be solved in time $2^{o(n)}$; see, e.g., the survey of Lokshtanov et al. [11].

One area where this line of research was particularly successful is the study of fixed-parameter algorithms parameterized by the treewidth of the input graph and understanding how the running time has to depend on the treewidth. Classic results on model checking monadic second-order logic on graphs of bounded treewidth, such as Courcelle's Theorem, provide a unified and generic way of proving fixed-parameter tractability of most of the tractable cases of this parameterization [1, 5]. While these results show that certain problems are solvable in time $f(t) \cdot n$ on graphs of treewidth t for some function f, the exact function $f(t)$ resulting from this approach is usually hard to determine and far from optimal. To get reasonable upper bounds on $f(t)$, one typically resorts to constructing a dynamic programming algorithm, which often is straightforward, but tedious.

The question whether the straightforward dynamic programming algorithms for bounded treewidth graphs are optimal received particular attention in 2011. On the hardness side, Lokshtanov, Marx and Saurabh proved that many natural algorithms are probably optimal [10, 12]. In particular, they showed that there are problems for which the $2^{\mathcal{O}(t \log t)} n$ time algorithms are best possible, assuming ETH. On the algorithmic side, Cygan et al. [6] presented a new technique, called *Cut&Count*, that improved the running time of the previously known (natural) algorithms for many connectivity problems. For example, previously only $2^{\mathcal{O}(t \log t)} \cdot n^{\mathcal{O}(1)}$ algorithms were known for Hamiltonian Cycle and Feedback Vertex Set, which was improved to $2^{\mathcal{O}(t)} \cdot n^{\mathcal{O}(1)}$ by Cut&Count. These results indicated that not only proving tight bounds for algorithms on tree decompositions is within our reach, but such a research may lead to surprising algorithmic developments. Further work includes derandomization of Cut&Count in [3, 8], an attempt to provide a meta-theorem to describe problems solvable in single-exponential time [13], and a new algorithm for Planarization [9].

We continue here this line of research by investigating a family of subgraph-hitting problems parameterized by treewidth and find surprisingly tight bounds for a number of problems. An interesting conceptual message of our results is that, for every integer $c \geq 1$, there are fairly natural problems where the best possible dependence on treewidth is of the form $2^{\mathcal{O}(t^c)}$.

Studied problems and motivation. In our paper we focus on the following generic H-Subgraph Hitting problem: for a pattern graph H and an input graph G, what is the minimum size of a set $X \subseteq V(G)$ that hits all subgraphs of G that are isomorphic to H? (Henceforth we call them *H-subgraphs* for brevity.) This problem generalizes a few ones studied in the literature, for example Vertex Cover (for $H = P_2$), where a tight $2^t \cdot t^{\mathcal{O}(1)} \cdot |V(G)|$ time bound is known [10], or finding largest induced subgraph of maximum degree at most Δ (for $H = K_{1,\Delta+1}$), which is $W[1]$-hard for treewidth parameter if Δ is a part of the input [2], but, to the best of our knowledge, no detailed study of treewidth parameterization for constant Δ has been done before. We also study the following *colorful* variant

Colorful H-Subgraph Hitting, where the input graph G is additionally equipped with a coloring $\sigma : V(G) \to V(H)$, and we are only interested in hitting H-subgraphs whose all vertices match their colors.

A direct source of motivation for our study is the work of Pilipczuk [13], which attempted to describe graph problems admitting fixed-parameter algorithms with running time of the form $2^{\mathcal{O}(t)} \cdot |V(G)|^{\mathcal{O}(1)}$, where t is the treewidth of G. The proposed description is a logical formalism where one can quantify existence of some vertex/edge sets, whose properties can be verified "locally" by requesting satisfaction of a formula of modal logic in every vertex. In particular, Pilipczuk argued that the language for expressing local properties needs to be somehow modal, as it cannot be able to discover cycles in a constant-radius neighborhood of a vertex. This claim was supported by a lower bound: unless ETH fails, for any constant $\ell \geq 5$, the problem of finding the minimum size of a set that hits all the cycles C_ℓ in a graph of treewidth t cannot be solved in time $2^{o(t^2)} \cdot |V(G)|^{\mathcal{O}(1)}$. Motivated by this result, we think that it is natural to investigate the complexity of hitting subgraphs for more general patterns H, instead of just cycles.

We may see the colorful variant as an intermediate step towards full understanding of the complexity of H-Subgraph Hitting, but it is also an interesting problem on its own. It often turns out that the colorful variants of problems are easier to investigate, while their study reveals useful insights; a remarkable example is the kernelization lower bound for Set Cover and related problems [7]. In our case, if we allow colors, a major combinatorial difficulty vanishes: when the algorithm keeps track of different parts of the pattern H that appear in the graph G, and combines a few parts into a larger one, the coloring σ ensures that the parts are vertex-disjoint. Hence, the colorful variant is easier to study, whereas at the same time it reveals interesting insight into the standard variant.

Our results and techniques. In the case of Colorful H-Subgraph Hitting, we obtain a tight bounds for the complexity of the treewidth parameterization. First, note that, in the presence of colors, one actually can solve Colorful H-Subgraph Hitting for each connected component of H independently; hence, we may focus only on connected patterns H. Second, we observe that there are two special cases. If H is a path then Colorful H-Subgraph Hitting reduces to a maximum flow/minimum cut problem, and hence is polynomial-time solvable. If H is a clique, then any H-subgraph of G needs to be contained in a single bag of any tree decomposition, and there is a simple $2^{\mathcal{O}(t)}|V(G)|$-time algorithm, where t is the treewidth of G. Finally, for the remaining cases we show that the dependence on treewidth is tightly connected to the value of $\mu(H)$, the maximum size of a minimal vertex separator in H (a separator S is minimal if there are two vertices x, y such that S is an xy-separator, but no proper subset of S is). We prove the following matching upper and lower bounds.

Theorem 1. *A* Colorful H-Subgraph Hitting *instance* (G, σ) *can be solved in time* $2^{\mathcal{O}(t^{\mu(H)})}|V(G)|$ *in the case when* H *is connected and is not a clique, where* t *is the treewidth of* G*.*

Theorem 2. *Let H be a graph that contains a connected component that is neither a path nor a clique. Then, unless ETH fails, there does not exist an algorithm that, given a* Colorful H-Subgraph Hitting *instance* (G, σ) *and a tree decomposition of G of width t, resolves (G, σ) in time $2^{o(t^{\mu(H)})}|V(G)|^{\mathcal{O}(1)}$.*

In all the theorems in this work we treat H as a fixed graph of constant size, and hence the factors hidden in the $\mathcal{O}$-notation may depend on the size of H.

In the absence of colors, we give preliminary results showing that the parameterized complexity of the treewidth parameterization of H-Subgraph Hitting is more involved than the one of the colorful counterpart. In this setting, we are able to relate the dependence on treewidth only to a larger parameter of the graph H. Let $\mu^\star(H)$ be the maximum size of $N_H(A)$, where A iterates over connected subsets of $V(H)$ such that $N_H(N_H[A]) \neq \emptyset$, i.e., $N_H[A]$ is not a whole connected component of H. Observe that $\mu(H) \leq \mu^\star(H)$ for any H. First, we were able to construct a counterpart of Theorem 1 only with the exponent $\mu^\star(H)$.

Theorem 3. *Assume H contains a connected component that is not a clique. Then, given a graph G of treewidth t, one can solve* H-Subgraph Hitting *on G in time $2^{\mathcal{O}(t^{\mu^\star(H)} \log t)}|V(G)|$.*

We remark that for Colorful H-Subgraph Hitting, an algorithm with running time $2^{\mathcal{O}(t^{\mu^\star(H)})}|V(G)|$ (as opposed to $\mu(H)$ in the exponent in Theorem 1) is rather straightforward: in the state of dynamic programming one needs to remember, for every subset X of the bag of size at most $\mu^\star(G)$, all forgotten connected parts of H that are attached to X and not hit by the constructed solution. To decrease the exponent to $\mu(H)$, we introduce a "prediction-like" definition of a state of the dynamic programming, leading to highly involved proof of correctness. For the problem without colors, however, even an algorithm with the exponent $\mu^\star(H)$ (Theorem 3) is far from trivial. We cannot limit ourselves to keeping track of forgotten connected parts of the graph H independently of each other, since in the absence of colors these parts may not be vertex-disjoint and, hence, we would not be able to reason about their union in latter bags of the tree decomposition. To cope with this issue, we show that the set of forgotten (not necessarily connected) parts of the graph H that are subgraphs of G can be represented as a *witness graph* with $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges. As there are only $2^{\mathcal{O}(t^{\mu^\star(H)} \log t)}$ possible graphs of this size, the running time bound follows.

We also observe that the bound of $\mathcal{O}(t^{\mu^\star(H)})$ on the size of a witness graph is not tight for many patterns H. For example, if H is a path, then we are able to find a witness graph with $\mathcal{O}(t)$ vertices and edges, and the algorithm of Theorem 3 runs in $2^{\mathcal{O}(t \log t)}|V(G)|$ time.

From the lower bound perspective, we were not able to prove an analog of Theorem 2 in the absence of colors. However, there is a good reason for that: we show that for any fixed $h \geq 2$ and $H = K_{2,h}$, the H-Subgraph Hitting problem is solvable in time $2^{\mathcal{O}(t^2 \log t)}|V(G)|$ for a graph G of treewidth t. This should be put in contrast with $\mu^\star(K_{2,h}) = \mu(K_{2,h}) = h$. Moreover, the lower bound of $2^{o(t^h)}$ can be proven if we break the symmetry of $K_{2,h}$ by attaching a triangle to each of the two degree-h vertices of $K_{2,h}$. This indicates that the optimal dependency on t in an algorithm for H-Subgraph Hitting may heavily rely on the symmetries of H, and may be more difficult to pinpoint.

2 Preliminaries

Graph notation. In most cases, we use standard graph notation. A *t-boundaried graph* is a graph G with a prescribed (possibly empty) *boundary* $\partial G \subseteq V(G)$ with $|\partial G| \leq t$, and an injective function $\lambda_G : \partial G \to \{1, 2, \ldots, t\}$. For a vertex $v \in \partial G$ the value $\lambda_G(v)$ is called the *label* of v.

A *colored graph* is a graph G with a function $\sigma : V(G) \to \mathbb{L}$, where $\mathbb{L}$ is some finite set of colors. A graph G is H-colored, for some other graph H, if $\mathbb{L} = V(H)$. We also say in this case that σ is an H-coloring of G.

A *homomorphism* of graphs H and G is a function $\pi : V(H) \to V(G)$ such that $ab \in E(H)$ implies $\pi(a)\pi(b) \in E(G)$. In the H-colored setting, i.e., when G is H-colored, we also require that $\sigma(\pi(a)) = a$ for any $a \in V(H)$ (every vertex of H is mapped onto appropriate color). The notion extends also to t-boundaried graphs: if both H and G are t-boundaried, we require that whenever $a \in \partial H$ then $\pi(a) \in \partial G$ and $\lambda_G(\pi(a)) = \lambda_H(a)$. Note, however, that we allow that a vertex of $\operatorname{int} H$ is mapped onto a vertex of ∂G.

An *H-subgraph of G* is any injective homomorphism $\pi : V(H) \to V(G)$. Recall that in the t-boundaried setting, we require that the labels are preserved, whereas in the colored setting, we require that the homomorphism respect colors. In the latter case, we call it a *σ-H-subgraph of G* for clarity.

We say that a set $X \subseteq V(G)$ *hits* a (σ-)H-subgraph π if $X \cap \pi(V(H)) \neq \emptyset$. The (Colorful) H-Subgraph Hitting problem asks for a minimum possible size of a set that hits all (σ-)H-subgraphs of G.

Tree decompositions. In this work, we view tree decompositions as rooted: in a tree decomposition $(\mathtt{T}, \beta)$, $\mathtt{T}$ is a rooted tree and $\beta(w)$ is a bag at node $w \in V(T)$. We moreover define $\gamma(w) = \bigcup_{w' \preceq w} \beta(w')$, where the union iterates over all descendants w' of w in $\mathtt{T}$, and $\alpha(w) = \gamma(w) \setminus \beta(w)$. In all our algorithms, by using a recent 5-approximation algorithm for treewidth [4], we assume that we are given a tree decomposition $(\mathtt{T}, \beta)$ where each bag is of size at most t; this linear shift in the value of t is irrelevant for the complexity bounds, but makes the notation much cleaner. Moreover, we assume that we are additionally equipped with a labeling $\Lambda : V(G) \to \{1, 2, \ldots, t\}$ that is injective on each bag $\beta(w)$; observe that it is straightforward to compute such a labeling in a top-down manner on $\mathtt{T}$. Consequently, we may treat each graph $G[\gamma(w)]$ as a t-boundaried graph with $\partial G[\gamma(w)] = \beta(w)$ and labeling $\Lambda|_{\beta(w)}$.

Important graph invariants, chunks and slices. For two vertices $a, b \in V(H)$, a set $S \subseteq V(H) \setminus \{a, b\}$ is an *ab-separator* if a and b are not in the same connected component of $H \setminus S$. The set S is additionally a *minimal ab-separator* if no proper subset of S is an ab-separator. A set S is a *minimal separator* if it is a

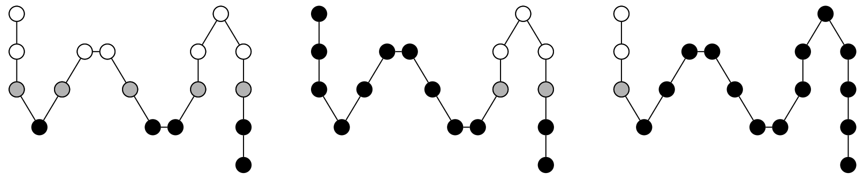

Fig. 1: White and gray vertices denote a slice (left), chunk (centre) and separator chunk (right) in a graph H being a path. The gray vertices belong to the boundary.

minimal ab-separator for some $a, b \in V(H)$. For a graph H, by $\mu(H)$ we denote the maximum size of a minimal separator in H.

For an induced subgraph $H' = H[D]$, $D \subseteq V(H)$, we define the boundary $\partial H' = N_H(V(H) \setminus D)$ and the interior $\operatorname{int} H' = D \setminus \partial H[D]$; thus $V(H') = \partial H' \uplus \operatorname{int} H'$. Observe that $N_H(\operatorname{int} H') \subseteq \partial H'$. An induced subgraph H' of H is a *slice* if $N_H(\operatorname{int} H') = \partial H'$, and a *chunk* if additionally $H[\operatorname{int} H']$ is connected. For a set $A \subseteq V(H)$, we use $\mathbf{p}[A]$ ($\mathbf{c}[A]$) to denote the unique slice (chunk) with interior A (if it exists). The intuition behind this definition is that, when we consider some bag $\beta(w)$ in a tree decomposition, a slice is a part of H that may already be present in $G[\gamma(w)]$ and we want to keep track of it. If a slice (chunk) $\mathbf{p}$ is additionally equipped with a injective labeling $\lambda_{\mathbf{p}} : \partial \mathbf{p} \to \{1, 2, \ldots, t\}$, then we call the resulting t-boundaried graph a *t-slice* (*t-chunk*, respectively).

By $\mu^\star(H)$ we denote the maximum size of $\partial \mathbf{c}$, where $\mathbf{c}$ iterates over all chunks of H. We remark here that both $\mu(H)$ and $\mu^\star(H)$ are positive only for graphs H that contain at least one connected component that is not a clique, as otherwise there are no chunks with nonempty boundary nor minimal separators in H.

Observe that if S is a minimal ab-separator in H, and A is the connected component of $H \setminus S$ that contains a, then $N_H(A) = S$ and $\mathbf{c}[A]$ is a chunk in H with boundary S. Consequently, $\mu(H) \leq \mu^\star(H)$ for any graph H. A chunk $\mathbf{c}$ for which $\partial \mathbf{c}$ is a minimal separator in H is henceforth called a *separator chunk*. See also Figure 1 for an illustration.

3 General algorithm for H-Subgraph Hitting

In this section we sketch an algorithm for H-Subgraph Hitting running in time $2^{\mathcal{O}(t^{\mu^\star(H)} \log t)} |V(G)|$, where t is the width of the tree decomposition we are working on. The general idea is the natural one: for each node w of the tree decomposition, for each set $\widehat{X} \subseteq \beta(w)$ and for each family $\mathbb{P}$ of t-slices, we would like to find the minimum size of a set $X \subseteq \alpha(w)$ such that, if we treat $G[\gamma(w)]$ as a t-boundaried graph with $\partial G[\gamma(w)] = \beta(w)$ and labeling $\Lambda|_{\beta(w)}$, then any slice that is a subgraph of $G[\gamma(w) \setminus (X \cup \widehat{X})]$ belongs to $\mathbb{P}$. However, as there can be as many as $t^{|H|}$ t-slices, we have too many choices for the family $\mathbb{P}$.

The essence of the proof, encapsulated in the next lemma, is to show that each "reasonable" choice of $\mathbb{P}$ can be encoded as a *witness graph* of essentially size $\mathcal{O}(t^{\mu^\star(H)})$. Such a claim would give a $2^{\mathcal{O}(t^{\mu^\star(H)} \log t)}$ bound on the number of possible witness graphs, and provide a good bound on the size of state space.

Lemma 4. *Assume H contains a connected component that is not a clique. Then, for any t-boundaried graph (G, λ) there exists a t-boundaried graph $(\widehat{G}, \lambda)$ that (a) is a subgraph of (G, λ), (b) $\partial G = \partial \widehat{G}$ and $G[\partial G] = \widehat{G}[\partial \widehat{G}]$, (c) $\widehat{G} \setminus E(\widehat{G}[\partial \widehat{G}])$ contains $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges, and, (d) for any t-slice $\mathbf{p}$ and any set $Y \subseteq V(G)$ such that $|Y| + |V(\mathbf{p})| \leq |V(H)|$, there exists a $\mathbf{p}$-subgraph in $(G \setminus Y, \lambda)$ if and only if there exists one in $(\widehat{G} \setminus Y, \lambda)$.*

Proof. We define $\widehat{G}$ by a recursive procedure. We start with $\widehat{G} = G[\partial G]$. Then, for every t-chunk $\mathbf{c} = (H', \lambda')$, we invoke a procedure $\texttt{enhance}(\mathbf{c}, \emptyset)$. The procedure $\texttt{enhance}(\mathbf{c}, X)$, for $X \subseteq V(G)$, first tries to find a $\mathbf{c}$-subgraph π in $(G \setminus X, \lambda)$. If there is none, the procedure terminates. Otherwise, it first adds all edges and vertices of $\pi(\mathbf{c})$ to $\widehat{G}$ that are not yet present there. Second, if $|X| < |V(H)|$, then it recursively invokes $\texttt{enhance}(\mathbf{c}, X \cup \{v\})$ for each $v \in \pi(\mathbf{c})$.

We first bound the size of the constructed graph $\widehat{G}$. There are at most $2^{|V(H)|} t^{\mu^\star(H)}$ choices for the chunk, since a chunk $\mathbf{c}$ is defined by its vertex set, and there are at most $t^{\mu^\star(H)}$ labellings of its boundary. The procedure $\texttt{enhance}(\mathbf{c}, X)$ at each step adds at most one copy of H to G, and branches into at most $|V(H)|$ directions. The depth of the recursion is bounded by $|V(H)|$. Hence, in total at most $2^{|V(H)|} t^{\mu^\star(H)} \cdot (|V(H)| + |E(H)|) \cdot |V(H)|^{|V(H)|}$ edges and vertices are added to $\widehat{G}$, except for the initial graph $G[\partial G]$.

It remains to argue that $\widehat{G}$ satisfies property (d). Clearly, since $(\widehat{G}, \lambda)$ is a subgraph of (G, λ), the implication in one direction is trivial. In the other direction, we start with the following claim.

Claim 5. *For any set $Z \subseteq V(G)$ of size at most $|V(H)|$, and for any t-chunk $\mathbf{c}$, if there exists a $\mathbf{c}$-subgraph in $(G \setminus Z, \lambda)$ then there exists also one in $(\widehat{G} \setminus Z, \lambda)$.*

Proof. Let π be a $\mathbf{c}$-subgraph in $(G \setminus Z, \lambda)$. Define $X_0 = \emptyset$. We will construct sets $X_0 \subsetneq X_1 \subsetneq \ldots$, where $X_i \subseteq Z$ for every i, and analyse the calls to the procedure $\texttt{enhance}(\mathbf{c}, X_i)$ in the process of constructing $\widehat{G}$.

Assume that $\texttt{enhance}(\mathbf{c}, X_i)$ has been invoked at some point during the construction; clearly this is true for $X_0 = \emptyset$. Since we assume $X_i \subseteq Z$, there exists a $\mathbf{c}$-subgraph in $(G \setminus X_i, \lambda)$ — π is one such example. Hence, $\texttt{enhance}(\mathbf{c}, X_i)$ has found a $\mathbf{c}$-subgraph π_i, and added its image to $\widehat{G}$. If π_i is a $\mathbf{c}$-subgraph also in $(\widehat{G} \setminus Z, \lambda)$, then we are done. Otherwise, there exists $v_i \in Z \setminus X_i$ that is also present in the image of π_i. In particular, since $|Z| \leq |V(H)|$, we have $|X_i| < |V(H)|$ and the call $\texttt{enhance}(\mathbf{c}, X_i \cup \{v_i\})$ has been invoked. We define $X_{i+1} := X_i \cup \{v_i\}$.

Since the sizes of sets X_i grow at each step, for some X_i, $i \leq |Z|$, we reach the conclusion that π_i is a $\mathbf{c}$-subgraph of $(\widehat{G} \setminus Z, \lambda)$, and the claim is proven. ⌟

Fix now a set $Y \subseteq V(G)$ and a t-slice $\mathbf{p}$ with labeling $\lambda_{\mathbf{p}}$ and with $|Y| + |V(\mathbf{p})| \leq |V(H)|$. Let π be a $\mathbf{p}$-subgraph of $(G \setminus Y, \lambda)$. Let $A_1, A_2, \ldots, A_r$ be the connected components of $H[\operatorname{int} \mathbf{p}]$. Define $H_i = N_H[A_i]$, and observe that each H_i is a chunk with $\partial H_i = N_H(A_i) \subseteq \partial \mathbf{p}$. We define $\lambda_i = \lambda_{\mathbf{p}}|_{\partial H_i}$ to obtain a t-chunk $\mathbf{c}_i = (H_i, \lambda_i)$. By the properties of a t-slice, each vertex of $\mathbf{p}$ is present in at least one graph $\mathbf{c}_i$, and vertices of $\partial \mathbf{p}$ may be present in more than one.

We now inductively define injective homomorphisms $\pi_0, \pi_1, \ldots, \pi_r$ such that of π_i maps the subgraph of $\mathbf{p}$ induced by $\partial\mathbf{p} \cup \bigcup_{j\leq i} A_j$ to $(\widehat{G} \setminus Y, \lambda)$, and does not use any vertex of $\bigcup_{j>i} \pi(A_j)$. Observe that π_r is a $\mathbf{p}$-subgraph of $(\widehat{G} \setminus Y, \lambda)$. Hence, this construction will conclude the proof of the lemma.

For base case, recall that $\pi(\partial\mathbf{p}) \subseteq \partial G = \partial\widehat{G}$ and define $\pi_0 = \pi|_{\partial\mathbf{p}}$. For the inductive case, assume that π_{i-1} has been constructed for some $1 \leq i \leq r$. Define

$$Z_i = Y \cup \pi(\partial\mathbf{p} \setminus \partial H_i) \cup \bigcup_{j<i} \pi_{i-1}(A_j) \cup \bigcup_{j>i} \pi(A_j).$$

Note that since π and π_{i-1} are injective and Y is disjoint with $\pi(\partial\mathbf{p})$, then we have that $Z_i \cap \pi(\partial\mathbf{p}) = \pi(\partial\mathbf{p} \setminus \partial H_i)$. This observation and the inductive assumption on π_{i-1} imply that the mapping $\pi|_{V(H_i)}$ does not use any vertex of Z_i. Thus, $\pi|_{V(H_i)}$ is a $\mathbf{c}_i$-subgraph in $(G \setminus Z_i, \lambda)$. Observe moreover that $|Z_i| \leq |Y| + |V(\mathbf{p})| \leq |V(H)|$. By Claim 5, there exists a $\mathbf{c}_i$-subgraph π'_i in $(\widehat{G} \setminus Z_i, \lambda)$. Observe that, since π'_i and π_{i-1} are required to preserve labellings on boundaries of their preimages, $\pi_i := \pi'_i \cup \pi_{i-1}$ is a function and a homomorphism. Moreover, by the definition of Z_i, π_i is injective and does not use any vertex of $\bigcup_{j>i} \pi(A_j)$. Hence, π_i satisfies all the required conditions, and the inductive construction is completed. This concludes the proof of the lemma. $\square$

Using Lemma 4, we now define states of dynamic programming algorithm on the input tree decomposition $(\mathtt{T}, \beta)$. For every node $w \in V(\mathtt{T})$, a *state* is a pair $\mathbf{s} = (\widehat{X}, \widehat{G})$ where $\widehat{X} \subseteq \beta(w)$ and $\widehat{G}$ is a graph with $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges such that $\beta(w) \setminus \widehat{X} \subseteq V(\widehat{G})$ and $\widehat{G}[\beta(w) \setminus \widehat{X}] = G[\beta(w) \setminus \widehat{X}]$. We treat $\widehat{G}$ as a t-boundaried graph with $\partial\widehat{G} = \beta(w) \setminus \widehat{X}$ and labeling $\Lambda|_{\beta(w)\setminus\widehat{X}}$. We say that a set $X \subseteq \alpha(w)$ is *feasible* for w and $\mathbf{s}$ if for every $Y \subseteq \beta(w) \setminus \widehat{X}$ and for every t-slice $\mathbf{p}$ such that $|Y| + |V(\mathbf{p})| \leq |V(H)|$, if there is a $\mathbf{p}$-subgraph in $(G[\gamma(w) \setminus (X \cup \widehat{X} \cup Y)], \Lambda|_{\beta(w)\setminus(\widehat{X}\cup Y)})$ then there is also one in $(\widehat{G} \setminus Y, \Lambda|_{\beta(w)\setminus(\widehat{X}\cup Y)})$. For every w and every state $\mathbf{s}$, we would like to compute $T[w, \mathbf{s}]$, the minimum possible size of a feasible set X. Note that the answer to the input H-Subgraph Hitting instance is the minimum value of $T[\mathtt{root}(\mathtt{T}), (\emptyset, \widehat{G})]$ where $\widehat{G}$ iterates over all graphs of with $\mathcal{O}(t^{\mu^\star(H)})$ vertices and edges that do not contain the t-slice $(H, \emptyset)$ as a subgraph. Hence, it remains to show how to compute the values $T[w, \mathbf{s}]$ in a bottom-up manner in the tree decomposition, which is relatively standard.

4 Discussion on special cases of H-Subgraph Hitting

As announced in the introduction, we now discuss a few special cases of H-Subgraph Hitting. First, let us consider H being a path, $H = P_h$ for some $h \geq 3$. Note that $\mu(P_h) = 1$, while $\mu^\star(P_h) = 2$ for $h \geq 5$. Observe that in the dynamic programming algorithm of the previous section we have that $G[\gamma(w) \setminus (X \cup X_w)]$ does not contain an H-subgraph and, hence, the witness graph obtained through Lemma 4 does not contain an H-subgraph as well. However, graphs excluding

P_h as a subgraph have very rigid structure: any their depth-first search tree has depth bounded by h. Using this insight, we can derive the following improvement of Lemma 4, that improves the running time of Theorem 3 to $2^{\mathcal{O}(t \log t)}|V(G)|$ for H being a path.[4]

Lemma 6. *Assume H is a path. Then, for any t-boundaried graph (G, λ) that does not contain an H-subgraph, there exists a witness graph as in Lemma 4 with $\mathcal{O}(t)$ vertices and edges.*

Second, let us consider $H = K_{2,h}$ (the complete biclique with 2 vertices on one side, and h on the other), for some $h \geq 2$. Observe that $\mu^\star(K_{2,h}) = \mu(K_{2,h}) = h$. On the other hand, we note the following.

Lemma 7. *Assume $H = K_{2,h}$ for some $h \geq 2$. If the witness graph given by Lemma 4 does not admit an H-subgraph, then it has $\mathcal{O}(t^2)$ vertices and edges.*

Proof. Since the constructed witness graph $\widehat{G}$ does not admit an H-subgraph, each two vertices $v_1, v_2 \in \partial\widehat{G}$ have less than h common neighbours in $\widehat{G}$, as otherwise there is a H-subgraph in $\widehat{G} \setminus \partial\widehat{G}$ on vertices v_1, v_2 and h vertices of $N_{\widehat{G}}(v_1) \cap N_{\widehat{G}}(v_2)$. Hence

$$\sum_{v \in V(\widehat{G}) \setminus \partial\widehat{G}} \binom{|N_{\widehat{G}}(v) \cap \partial\widehat{G}|}{2} \leq (h-1)\binom{|\partial\widehat{G}|}{2} \leq (h-1)\binom{t}{2}. \qquad (1)$$

Let $V(H) = \{a_1, a_2, b_1, b_2, \ldots, b_h\}$ where $A := \{a_1, a_2\}$ and $B := \{b_1, b_2, \ldots, b_h\}$ are bipartition classes of H. Note that there are only two types of proper chunks in H: $N_H[a_i]$, $i = 1, 2$ and $N_H[b_j]$, $1 \leq j \leq h$. Hence, one can easily verify that in the construction of the witness graph $\widehat{G}$ of Lemma 4 every vertex $v \in \widehat{G} \setminus \partial\widehat{G}$ has at least two neighbours in $\partial\widehat{G}$, and $\widehat{G} \setminus \partial\widehat{G}$ is edgeless. Then we have $|N_{\widehat{G}}(v) \cap \partial\widehat{G}| \leq 2\binom{|N_{\widehat{G}}(v) \cap \partial\widehat{G}|}{2}$ for each $v \in V(\widehat{G}) \setminus \partial\widehat{G}$. Consequently, by (1) there are at most $2(h-1)\binom{t}{2}$ edges of $\widehat{G}$ with exactly one endpoint in $\partial\widehat{G}$, whereas there are at most $\binom{t}{2}$ edges in $\widehat{G}[\partial\widehat{G}]$. The lemma follows. $\square$

Lemma 7 together with a dynamic programming as in Section 3 imply that $K_{2,h}$-Subgraph Hitting can be solved in $2^{\mathcal{O}(t^2 \log t)}|V(G)|$ time, in spite of the fact that $\mu^\star(K_{2,h}) = \mu(K_{2,h}) = h$.

We now show that a slight modification of $K_{2,h}$ enables us to prove a much higher lower bound. For this, let us consider a graph H_h for $h \geq 2$ defined as $K_{2,h}$ with triangles attached to both degree-h vertices. Note that $\mu(H_h) = \mu^\star(H_h) = h$. One may view H_h as $K_{2,h}$ with some symmetries broken, so that the proof of Lemma 7 does not extend to H_h. We observe that the lower bound proof of Theorem 2 works, with small modifications, also for the case of H_h-Subgraph Hitting.

Theorem 8. *Unless ETH fails, for every $h \geq 2$ there does not exist an algorithm that, given a H_h-Subgraph Hitting instance G and a tree decomposition of G of width t, resolves G in time $2^{o(t^h)}|V(G)|^{\mathcal{O}(1)}$.*

[4] The proofs Lemma 6 and Theorem 8 are deferred to the full version of the paper.

Furthermore, the proof of Theorem 8 does not need to assume that h is a constant. Thus, we obtain the following interesting double-exponential lower bound.

Corollary 9. *Unless ETH fails, there does not exist an algorithm that, given a graph G with a tree decomposition of width t, and an integer $h = \mathcal{O}(\log |V(G)|)$, finds in $2^{2^{o(t)}} |V(G)|^{\mathcal{O}(1)}$ time the minimum size of a set that hits all H_h-subgraphs of G.*

5 Overview of the proof for colorful variant

5.1 Proof sketch of Theorem 1

In this sketch, we focus on the definition of a state that will be used in the dynamic programming algorithm on the input tree decomposition $(\mathtt{T}, \beta)$. A *potential chunk* is a separator t-chunk $\mathbf{c}[A]$. A *state* at node $w \in V(\mathtt{T})$ is a pair $(\widehat{X}, \mathbb{C})$ where $\widehat{X} \subseteq \beta(w)$ and $\mathbb{C}$ is a family of potential chunks, where each chunk $\mathbf{c}$ in $\mathbb{C}$: (i) uses only labels of $\Lambda^{-1}(\beta(w) \setminus \widehat{X})$; and (ii) the mapping $\pi : \partial\mathbf{c} \to \beta(w) \setminus \widehat{X}$ that maps a vertex of $\partial\mathbf{c}$ to a vertex with the same label is a homomorphism from $H[\partial\mathbf{c}]$ to G (in particular, it respects colors). Observe that, as $|\partial\mathbf{c}| \leq \mu(H)$ for any separator chunk $\mathbf{c}$, there are $\mathcal{O}(t^{\mu(H)})$ possible separator t-chunks, and hence $2^{\mathcal{O}(t^{\mu(H)})}$ possible states for a fixed node w.

The intuitive idea behind a state is that, for node $w \in V(\mathtt{T})$ and state $(\widehat{X}, \mathbb{C})$, we investigate the possibility of the following: for a solution X we are looking for, it holds that $\widehat{X} = X \cap \beta(w)$ and the family $\mathbb{C}$ is exactly the set of possible separator chunks of H that are subgraphs of $G \setminus X$, where the subgraph relation is defined as on t-boundaried graphs and $G \setminus X$ is equipped with $\partial G \setminus X = \beta(w) \setminus X$ and labeling $\Lambda_w|_{\beta(w)\setminus X}$. The difficult part of the proof is to show that this information is sufficient, in particular, it suffices to keep track only of the separator chunks, and not all proper chunks of H. We emphasize here that the intended meaning of the set $\mathbb{C}$ is that it represents separator chunks present in the entire $G \setminus X$, not $G[\gamma(w)] \setminus X$. That is, to be able to limit ourselves only to separator chunks, we need to encode in the state some prediction for the future. This makes our dynamic programming algorithm rather non-standard.

Let us proceed to a more formal definition of the dynamic programming table. For a bag w and a state $\mathbf{s} = (\widehat{X}, \mathbb{C})$ at w we define the graph $G(w, \mathbf{s})$ as follows. We first take the graph $G[\gamma(w)] \setminus \widehat{X}$ and then, for each chunk $\mathbf{c} \in \mathbb{C}$ we add a disjoint copy of $\mathbf{c}$ to $G(w, \mathbf{s})$ and identify the pairs of vertices with the same label in $\partial\mathbf{c}$ and in $\beta(w)$. Note that $G[\gamma(w)] \setminus \widehat{X}$ is an induced subgraph of $G(w, \mathbf{s})$: by the properties of elements of $\mathbb{C}$, no new edge has been introduced between two vertices of $\beta(w)$. We make $G(w, \mathbf{s})$ a t-boundaried graph in a natural way: $\partial G(w, \mathbf{s}) = \beta(w) \setminus \widehat{X}$ with labeling $\Lambda|_{\partial G(w,\mathbf{s})}$. Here we exploit the crucial property of the colored version of the problem: no two isomorphic chunks glued to $\beta(w)$ can participate together in any σ-H-subgraph, since their vertices have the same colors. Therefore, attaching undeletable chunks of $\mathbb{C}$ explicitly to $\beta(w)$

is equivalent to just allowing these chunks to be present either in the future, or in the forgotten part of the graph.

For each bag w and for each state $\mathbf{s} = (\widehat{X}, \mathbb{C})$ we say that a set $X \subseteq \alpha(w)$ is *feasible* if $G(w, \mathbf{s}) \setminus X$ does not contain any σ-H-subgraph, and for any separator t-chunk $\mathbf{c}$ of H, if there is a $\mathbf{c}$-subgraph in $G(w, \mathbf{s}) \setminus X$ then $\mathbf{c} \in \mathbb{C}$. We would like to compute the value $T[w, \mathbf{s}]$ that equals the minimum size of a feasible set X, using the standard bottom-up dynamic programming. Observe that $T[\mathtt{root(T)}, \emptyset]$ is the minimum size of a solution for Colorful H-Subgraph Hitting.

5.2 Proof sketch of Theorem 2

The proof of this theorem is inspired by the approach used in [13] for the lower bound for C_ℓ-Subgraph Hitting.

Consider a minimal separator S in H such that $\mu := |S| = \mu(H)$ and A, B are two such distinct connected components of $H \setminus S$ with $N_H(A) = N_H(B) = S$.

With some simple preprocessing, we may assume we are given n-variable 3-CNF formula Φ where each variable appears exactly three times, at least once positively and at least once negatively. Let s be a smallest integer such that $s^\mu \geq 3n$; $s = \mathcal{O}(n^{1/\mu})$. We start by introducing a set M of $s\mu$ vertices $w_{i,c}$, $1 \leq i \leq s$, $c \in S$, with coloring $\sigma(w_{i,c}) = c$. The set M is the central part of the constructed graph G. In particular, each connected component of $G \setminus M$ will be of constant size, immediately implying that G has treewidth $\mathcal{O}(n^{1/\mu})$.

To each clause C of Φ, and to each literal l in C, assign a function $f_{C,l} : S \to \{1, 2, \ldots, s\}$ such that $f_{C,l} \neq f_{C',l'}$ for $(C, l) \neq (C', l')$. Observe that this is possible due to the assumption $s^\mu \geq 3n$ and the preprocessing step.

The main idea is as follows. For each clause C and literal l in C, we attach a copy of $H[N_H[A]]$ and a copy of $H[N_H[B]]$ to $\{w_{f_{C,l}(c),c} \mid c \in S\}$ in a natural way. For each variable x we use constant-size gadgets to we wire up all the copies of $H[N_H[A]]$ that correspond to an occurrence of that variable, so that with minimum budget we may hit all copies corresponding to positive occurrences of x or all copies corresponding to negative occurrences; this choice corresponds to the decision on the value of x. Similarly, for each clause C we use constant-size gadgets to wire up all the copies of $H[N_H[B]]$ that correspond to literals in C, so that with minimum budget we may hit all but one of these copies; this choice corresponds to the decision which literal of C is satisfied by an assignment. Finally, we attach to the construction a large number of copies of $H \setminus (A \cup B \cup S)$, so that a small solution needs to hit any σ-H-subgraph of (G, σ) in a vertex of $A \cup B$. The construction enforces that, whenever a clause C chooses a literal l to satisfy C, it leaves the corresponding copy of $H[B]$ not hit, forcing the solution to hit the corresponding copy of $H[A]$, and therefore forcing the correct assignment of the variable in l.

6 Conclusions and open problems

Our preliminary study of the treewidth parameterization of the H-Subgraph Hitting problem revealed that its parameterized complexity is highly involved.

Whereas for the more graspable colored version we obtained essentially tight bounds, a large gap between lower and upper bounds remains for the standard version. In particular, the following two questions arise: Can we improve the running time of Theorem 3 to factor $t^{\mu(H)}$ in the exponent? Is there any relatively general symmetry-breaking assumption on H that would allow us to show a $2^{o(t^{\mu(H)})}$ lower bound in the absence of colors?

In a broader view, let us remark that the complexity of the treewidth parameterization of *minor-hitting* problems is also currently highly unclear. Here, for a minor-closed graph class $\mathcal{G}$ and input graph G, we seek for the minimum size of a set $X \subseteq V(G)$ such that $G \setminus X \in \mathcal{G}$, or, equivalently, X hits all minimal forbidden minors of $\mathcal{G}$. A straightforward dynamic programming algorithm has double-exponential dependency on the width of the decomposition. However, it was recently shown that $\mathcal{G}$ being the class of planar graphs, a $2^{\mathcal{O}(t \log t)}|V(G)|$-time algorithm exists [9]. Can this result be generalized to more graph classes?

References

1. Arnborg, S., Lagergren, J., Seese, D.: Easy problems for tree-decomposable graphs. J. Algorithms 12(2), 308–340 (1991)
2. Betzler, N., Bredereck, R., Niedermeier, R., Uhlmann, J.: On bounded-degree vertex deletion parameterized by treewidth. Discrete Appl. Math. 160(1-2), 53–60 (2012)
3. Bodlaender, H.L., Cygan, M., Kratsch, S., Nederlof, J.: Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. In: ICALP (1). pp. 196–207 (2013)
4. Bodlaender, H.L., Drange, P.G., Dregi, M.S., Fomin, F.V., Lokshtanov, D., Pilipczuk, M.: An $\mathcal{O}(c^k n)$ 5-approximation algorithm for treewidth. In: FOCS. pp. 499–508 (2013)
5. Courcelle, B.: The monadic second-order logic of graphs I: Recognizable sets of finite graphs. Inf. Comput. 85, 12–75 (1990)
6. Cygan, M., Nederlof, J., Pilipczuk, M., Pilipczuk, M., van Rooij, J.M.M., Wojtaszczyk, J.O.: Solving connectivity problems parameterized by treewidth in single exponential time. In: FOCS. pp. 150–159 (2011)
7. Dom, M., Lokshtanov, D., Saurabh, S.: Incompressibility through colors and ids. In: ICALP (1). pp. 378–389 (2009)
8. Fomin, F.V., Lokshtanov, D., Saurabh, S.: Efficient computation of representative sets with applications in parameterized and exact algorithms. In: SODA. pp. 142–151 (2014)
9. Jansen, B.M.P., Lokshtanov, D., Saurabh, S.: A near-optimal planarization algorithm. In: SODA. pp. 1802–1811 (2014)
10. Lokshtanov, D., Marx, D., Saurabh, S.: Known algorithms on graphs on bounded treewidth are probably optimal. In: SODA. pp. 777–789 (2011)
11. Lokshtanov, D., Marx, D., Saurabh, S.: Lower bounds based on the exponential time hypothesis. Bulletin of the EATCS 105, 41–72 (2011)
12. Lokshtanov, D., Marx, D., Saurabh, S.: Slightly superexponential parameterized problems. In: SODA. pp. 760–776 (2011)
13. Pilipczuk, M.: Problems parameterized by treewidth tractable in single exponential time: A logical approach. In: MFCS. pp. 520–531 (2011)