

Turing kernelization for finding long paths in graph classes excluding a topological minor

Citation for published version (APA):

Jansen, B. M. P., Pilipczuk, M., & Wrochna, M. (2019). Turing kernelization for finding long paths in graph classes excluding a topological minor. *Algorithmica*, 81(10), 3936-3967. <https://doi.org/10.1007/s00453-019-00614-4>

DOI:

[10.1007/s00453-019-00614-4](https://doi.org/10.1007/s00453-019-00614-4)

Document status and date:

Published: 01/10/2019

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.



Turing Kernelization for Finding Long Paths in Graph Classes Excluding a Topological Minor

Bart M. P. Jansen¹ · Marcin Pilipczuk² · Marcin Wrochna² 

Received: 6 November 2017 / Accepted: 23 July 2019 / Published online: 3 August 2019
© The Author(s) 2019

Abstract

The notion of Turing kernelization investigates whether a polynomial-time algorithm can solve an NP-hard problem, when it is aided by an oracle that can be queried for the answers to bounded-size subproblems. One of the main open problems in this direction is whether k -PATH admits a polynomial Turing kernel: can a polynomial-time algorithm determine whether an undirected graph has a simple path of length k , using an oracle that answers queries of size $k^{\mathcal{O}(1)}$? We show this can be done when the input graph avoids a fixed graph H as a topological minor, thereby significantly generalizing an earlier result for bounded-degree and $K_{3,t}$ -minor-free graphs. Moreover, we show that k -PATH even admits a polynomial Turing kernel when the input graph is not H -topological-minor-free itself, but contains a known vertex modulator of size bounded polynomially in the parameter, whose deletion makes it so. To obtain our results, we build on the graph minors decomposition to show that any H -topological-minor-free graph that does not contain a k -path, has a separation that can safely be reduced after communication with the oracle.

Keywords Turing kernelization · k -path · Graph minors decomposition

This work was supported by the Netherlands Organization for Scientific Research (NWO) Veni Grant 639.021.437 “Frontiers in Parameterized Preprocessing” and Gravitation Grant 024.002.003 “Networks”. Marcin Pilipczuk is supported by the “Recent trends in kernelization: theory and experimental evaluation” Project, carried out within the Homing programme of the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund. Marcin Wrochna is supported by the National Science Centre of Poland Grant Number 2013/11/D/ST6/03073 and by the Foundation for Polish Science (FNP) via the START stipend programme.

✉ Marcin Wrochna
m.wrochna@mimuw.edu.pl

¹ Eindhoven University of Technology, Eindhoven, The Netherlands

² University of Warsaw, Warsaw, Poland

1 Introduction

Suppose that Alice is a polynomial-time agent faced with an input to an NP-hard problem that she wishes to solve exactly. To facilitate her in this process, she can ask questions to an all-knowing oracle. These will be answered truthfully and instantly, but the oracle is memory-less and will not take previous questions into account when answering the next one. How large do these questions have to be, to allow Alice to find the answer to her problem? Clearly, it suffices to send the entire input to the oracle, who determines the answer and sends it to Alice. Could there be a more clever strategy? Alice can attempt to isolate a small but meaningful question about the behavior of her input, such that after learning its answer, she can reduce her current input to a smaller input without changing the outcome. Iterating this process solves her problem: when it has become sufficiently small, it can be posed to the oracle in its entirety.

Such problem-solving strategies can be rigorously analyzed using the notion of *Turing kernelization* that originated in parameterized algorithmics. The parameter makes it possible to express how the size of the questions that Alice asks, depends on properties of the input that she is given. (See Sect. 3 for a formal definition.)

Understanding the power of Turing kernelization is one of the main open research horizons in parameterized algorithmics. There is a handful of problems for which a nontrivial Turing kernelization is known [1,3,4,6,16,20,22,23,29,31]. On the other hand, there is a hierarchy of parameterized complexity classes which are conjectured not to admit polynomial Turing kernels [19]. Arguably, the main open problem (cf. [4, 5,19]) in this direction is to determine whether the k -PATH problem (determine whether an undirected graph has a simple path of length k) has a polynomial Turing kernel. In earlier work [21], the first author showed that k -PATH indeed admits polynomial Turing kernels on several graph classes. In this work, we develop Turing kernels for k -PATH in a much more general setting.

Our results Our algorithmic contributions are twofold. First of all, we extend the Turing kernelization for k -PATH to much broader families of sparse graphs. Whereas the earlier work could only deal with $K_{3,t}$ -minor-free graphs, claw-free graphs, and bounded-degree graphs, we show that a Turing kernelization exists on H -minor-free graphs for all fixed graphs H . We even lift the kernelization to H -topological-minor-free graphs, thereby capturing a common generalization of the bounded-degree and $K_{3,t}$ -minor-free cases.

Theorem 1 *For every fixed graph H , the k -PATH problem, restricted to graphs excluding H as a topological minor, admits a polynomial Turing kernel. Furthermore, the kernel runs in time $k^{\mathcal{O}_H(1)} \cdot n^2m$ and invokes $k^{\mathcal{O}_H(1)} \cdot n$ calls to the oracle.¹*

Our second contribution is the following theorem. By a novel algorithmic approach, we obtain a Turing kernelization even when the input graph does not belong to the desired restricted graph class itself, but contains a small known vertex modulator whose deletion places the graph in such a graph class.

¹ $\mathcal{O}_H(\cdot)$ notation has the same meaning as $\mathcal{O}(\cdot)$, but indicates that the hidden constant depends on H .

Theorem 2 *For every fixed graph H , the k -PATH problem, on instances consisting of a graph G , integer k , and a modulator $M \subseteq V(G)$ such that $G - M$ is H -topological-minor-free, admits a polynomial Turing kernel, when parameterized by k and $|M|$.*

Techniques To explain our approach, we briefly recall the idea behind the Turing kernelization for k -PATH on planar graphs. The graph-theoretical core is a theorem by Chen and Yu [8] stating that every n -vertex triconnected planar graph contains a polynomially long cycle (of length $\Omega(n^\varepsilon)$ for some $\varepsilon > 0$). Hence given a triconnected subgraph and an integer k , we can immediately conclude that a k -path exists or continue knowing that the subgraph has size polynomially bounded in k . For a general planar graph, Tutte's decomposition into triconnected bags thus becomes a tree decomposition with bags of size polynomially bounded in k . Furthermore, in this decomposition the adhesions, i.e. intersections of two adjacent bags, have constant size (at most 2). This makes it possible to find a separation (A, B) of the graph of constant order $|A \cap B|$ and with A moderately large: polynomially bounded in k (so that $G[A]$ can be sent to the oracle) but large enough to contain an irrelevant vertex (one whose removal will not change the answer).

To find such an irrelevant vertex, we use the constant bound on the order of the separation to bound the number structurally different ways in which the k -path we seek could intersect A . For example, if the separator $A \cap B$ consists of two vertices u, v then a k -path in G may use A to form a long path connecting u and v . Alternatively, a k -path in G may use A to realize two disjoint paths, one ending in u and one in v , which are connected into a single path through B . Hence the way in which a k -path uses A can be described by how the various pieces of its intersection with A connect to $A \cap B$. Details are presented in following sections, but the idea is that for each such way, we ask the oracle whether sufficiently long paths in A realize it (using the fact that an oracle to an NP-complete problem can answer any problem in NP) and we ask for paths witnessing this (using self-reduction). This allows us to collect a subset of A polynomially bounded in k (independent of $|A|$) that witnesses every possible way, making all remaining vertices in A irrelevant. Hence by choosing A to be moderately large we can guarantee to always remove some vertices, until the whole graph is polynomially bounded.

Theorem 1 is established by lifting this approach to H -(topological)-minor-free graphs: we show that every H -free graph that does not have a k -path, has a tree decomposition of constant adhesion size and width $\text{poly}(k)$. This requires an adaptation of the decomposition theorems of Robertson and Seymour [28] (for minors) and Grohe and Marx [18] (for topological minors). These guarantee a tree decomposition of constant adhesion (depending only on the fixed graph H) with bags that are possibly large but nearly-embeddable or have bounded degree (for topological minors). As before, if these bags are also triconnected, known theorems [7,9,30] can be used to bound their size polynomially. However, several technical steps are needed to translate this into the desired result, due to details of the decomposition theorems (vortices, virtual edges, and the lack of a direct polynomial-time algorithm to compute the decomposition).

To prove Theorem 2, we introduce a new algorithmic tool for finding irrelevant vertices for the k -PATH problem in the presence of a modulator M in the input graph G .

Since Theorem 1 can be applied to find a k -path in $G - M$ if one exists, the challenge is to detect a k -path in G that jumps between M and $G - M$ several times. The absence of a k -path in $G - M$ implies it has a tree decomposition of width $\text{poly}(k)$ and constant adhesion size. Using Theorem 1 as a subroutine, we compute paths in $G - M$ between every two vertices in M , and pack them to obtain a vertex set X of size polynomial in $k + |M|$ with the following guarantee. If there is a k -path, then there is a *guarded* k -path in which each successive pair of vertices in M are connected by a subpath through $G - M$ that intersects X . Using the tree decomposition of $G - M$, the standard ancestor-marking technique allows us to identify a vertex subset C of $G - (M \cup X)$ that is adjacent to at most two adhesions and hence to constantly many vertices from X . Unless G is already small, we can find such a set C that is sufficiently large to be reducible but small enough that we may invoke the oracle for questions about it. We then show that the constant bound on the adhesions (independent of $|M|, k$) suffices to reduce the graph by finding an irrelevant vertex as before.

Organization After preliminaries in Sect. 2, we give a generic Turing-style reduction rule for k -PATH in Sect. 3. In Sect. 4 we show that an H -minor-free graph either has a k -path or a separation that is suitable for reduction. In Sect. 5 we extend this to topological minors. Finally, in Sect. 6 we present a Turing kernel applicable when the input graph has a small modulator to a suitable graph class.

2 Preliminaries

Notation All graphs we consider are finite, simple, and undirected. A *separation* of a graph G is a pair (A, B) , $A, B \subseteq V(G)$ such that $A \cup B = V(G)$ and there are no edges between $A \setminus B$ and $B \setminus A$. The *order* of the separation (A, B) is $|A \cap B|$. A graph is *triconnected* if it is connected and cannot be disconnected by deleting fewer than three vertices. When referring to the *size* of a graph in our statements, we mean the number of vertices.

A *tree decomposition* of a graph G is a pair (T, \mathcal{X}) where T is a rooted tree and \mathcal{X} is a function that assigns to every node $t \in V(T)$ a subset $\mathcal{X}(t)$ of $V(G)$ called a *bag* such that:

- $\bigcup_{t \in V(T)} \mathcal{X}(t) = V(G)$;
- for each edge $uv \in E(G)$, there is a node $t \in V(T)$ with $u, v \in \mathcal{X}(t)$;
- for each $v \in V(G)$, the nodes $\{t \mid v \in \mathcal{X}(t)\}$ induce a (connected) subtree of T .

The *width* of (T, \mathcal{X}) is $\max_{t \in V(T)} |\mathcal{X}(t)| - 1$. For an edge tt' of T , we call the set $\mathcal{X}(t) \cap \mathcal{X}(t')$ the *adhesion* of tt' . The *adhesion size* of (T, \mathcal{X}) is the maximum size of an adhesion. For a decomposition (T, \mathcal{X}) of G and a node $t \in V(T)$, the *torso*, denoted $\text{TORSO}(G, \mathcal{X}(t))$, is the graph obtained from $G[\mathcal{X}(t)]$ by adding an edge between each pair of vertices in $\mathcal{X}(t) \cap \mathcal{X}(t')$, for every neighbor t' of t in T (so each adhesion induces a clique in the torso). Added edges not present in G are called *virtual edges*. For a subtree $T' \subseteq T$ we write $\mathcal{X}(T')$ for the union $\bigcup_{t \in V(T')} \mathcal{X}(t)$ of bags in T' .

For an edge $t_1 t_2 \in E(T)$, let T_i be the connected component of $T - \{t_1 t_2\}$ that contains t_i . Let $V_i = \mathcal{X}(T_i)$. Observe that the properties of a tree decomposition imply that (V_1, V_2) is a separation with $V_1 \cap V_2 = \mathcal{X}(t_1) \cap \mathcal{X}(t_2)$.

A decomposition (T, \mathcal{X}) is *connected* if for every $t \in V(T)$ and its child t' , if $T_{t'}$ is the subtree of T rooted at t' , we have (i) that $G[\mathcal{X}(T_{t'}) \setminus \mathcal{X}(t)]$ is connected, and (ii) that $\mathcal{X}(T_{t'}) \setminus \mathcal{X}(t)$ has edges to every vertex of the adhesion $\mathcal{X}(t) \cap \mathcal{X}(t')$. It is straightforward to turn any decomposition into a connected one without increasing its width nor adhesion. For (i), as long as there exists a pair (t, t') violating the requirement, make a distinct copy $T_{t'}^C$ of $T_{t'}$ for each connected component C of $G[\mathcal{X}(T_{t'}) \setminus \mathcal{X}(t)]$, restrict the bags of $T_{t'}^C$ to the vertices of $C \cup \mathcal{X}(t)$ only, and attach $T_{t'}^C$ as a subtree with the root being a child of t . For (ii), while there is a vertex $v \in \mathcal{X}(t) \cap \mathcal{X}(t')$ that has no neighbors in $\mathcal{X}(T_{t'}) \setminus \mathcal{X}(t)$, simply remove v from all bags in $T_{t'}$. Connected tree decompositions are useful for our arguments, because they imply that the virtual edges of $\text{TORSO}(G, \mathcal{X}(t))$ can in fact be realized by paths in the graph whose internal vertices are represented in $T - \{t\}$.

We will also need the following non-standard complexity measure of a tree decomposition (T, \mathcal{X}) . For every $t \in V(T)$, the number of distinct adhesions $\mathcal{X}(t) \cap \mathcal{X}(t')$ for $t' \in N_T(t)$ is called the *adhesion degree* of t . The maximum adhesion degree over all nodes t is the *adhesion degree* of the decomposition (T, \mathcal{X}) . Observe that if a tree decomposition (T, \mathcal{X}) has width less than ℓ and adhesions of size at most h , then its adhesion degree is at most

$$\sum_{i=0}^h \binom{\ell}{i} \leq (1 + \ell)^h.$$

However, in sparse graph classes we can prove a much better bound on the adhesion degree due to linear bounds on the number of cliques in such graphs; cf. Lemma 17.

A *path decomposition* is a tree decomposition where T is a path; we will denote it simply as a sequence of bags $\mathcal{Z}_1, \dots, \mathcal{Z}_{|V(T)|}$.

3 Turing Kernels

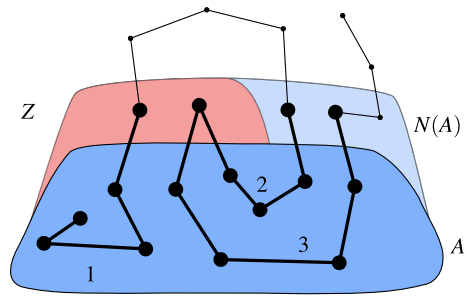
In this section we introduce a general toolbox and notation for proving our Turing kernel bounds.

3.1 Definitions and the Auxiliary Problem

For a parameterized problem Π and a computable function f , a *Turing kernel for Π of size f* is an algorithm that solves an input instance (x, k) of Π in polynomial time, given access to an oracle that solves instances (x', k') of Π with $|x'|, k' \leq f(k)$. A Turing kernel is a *polynomial* one if f is a polynomial.

If we are only interested in distinguishing between NP-complete problems admitting a polynomial Turing kernel from the ones that do not admit such a kernel, we can assume that the oracle solves an arbitrary problem in NP, not necessarily the k -

Fig. 1 A set A (blue) and a path with three A -traverses (bold). The path is guarded w.r.t. $Z \subseteq N(A)$ (the red set), since each A -traverse has an endpoint in it. (A path fully contained in A (with one A -traverse) or disjoint from A (with no A -traverses) would also be guarded) (Color figure online)



PATH problem. Indeed, note that by the definition of NP-completeness, an oracle to a problem in NP can be implemented with an oracle to k -PATH with only polynomial blow-up in the size of the passed instances.

In our work, it will be convenient to reduce to the AUXILIARY LINKAGE problem, defined as follows. The input consists of an undirected graph G' , an integer k' , a set of terminals $S \subseteq V(G')$, and a number of requests R_1, R_2, \dots, R_r ; a request is a set of at most two terminals. A path P_i in G is said to satisfy a request R_i if $V(P_i) \cap S = R_i$ and every vertex of $V(P_i) \cap S$ is an endpoint of P_i . With such an input, the AUXILIARY LINKAGE problem asks for a sequence of r paths P_1, P_2, \dots, P_r such that P_i satisfies R_i for every $1 \leq i \leq r$, $|\bigcup_{i=1}^r V(P_i)| = k'$, and every vertex of $V(G) \setminus S$ is contained in at most one path P_i (i.e., the paths P_i are vertex-disjoint, except that they may share an endpoint, but only if the requests ask them to do so).

We remark that AUXILIARY LINKAGE is a more general problem than k -PATH: an instance with $G' = G, k' = k, S = \emptyset, r = 1$, and $R_1 = \emptyset$ asks precisely for a k -path in G .

Clearly, the decision version of the AUXILIARY LINKAGE problem belongs to the class NP. By using its self-reducibility (cf. [21, Lemma 2]), we assume that the oracle returns a sequence of paths $(P_i)_{i=1}^r$ in case of a positive answer. That is, in all subsequent bounds on the number of AUXILIARY LINKAGE oracle calls, the bound adheres to the number of calls to an oracle that returns the actual paths P_i ; if one wants to use a decision oracle, one should increase the bound by the blow-up implied by the self-reducibility application (i.e., at most $|E(H)|$ for calls on a graph H).

3.2 Generic Reduction Rule

We now show a generic reduction rule for the k -PATH problem. We start with a few definitions.

Definition 3 For a graph G , a subset $A \subseteq V(G)$, and a simple path P in G , an A -traverse of P is a maximal subpath of P that contains at least one vertex of A and has all its internal vertices in A .

Note that if Q is an A -traverse of P , then every endpoint of Q is either an endpoint of P or lies in $N_G(A)$. See Fig. 1.

Definition 4 Let G be a graph, $A \subseteq V(G)$, and let k be an integer. A set $Z \subseteq N(A)$ is called a k -guard of A if the following implication holds: if G admits a k -path, then there exists a k -path P in G that is either contained in A or such that every A -traverse of P has at least one endpoint in Z .

Given a graph G , a set $A \subseteq V(G)$, and a k -guard $Z \subseteq N(A)$ of A , a k -path P satisfying properties as in the above definition is called *guarded* (w.r.t. k , A , and Z). If the integer k and the set A are clear from the context, we call such a set Z simply a guard.

Observe that $Z = N(A)$ is always a guard, but sometimes we will be able to find smaller ones. Of particular interest will be guards of constant size, as our kernel sizes will depend exponentially on the guard size. To describe our single reduction rule, we show how solutions to AUXILIARY LINKAGE can be used to preserve the existence of guarded k -paths.

Assume we are given a graph G , a set $A \subseteq V(G)$, an integer k , and a k -guard $Z \subseteq N(A)$ of A . Let $h = |Z|$ and $\ell = |N(A)|$. Furthermore, assume that G admits a k -path, and let P be a guarded one w.r.t. A and Z . Let (Q_1, Q_2, \dots, Q_r) be the A -traverses of P , let $R_i = V(Q_i) \setminus A = V(Q_i) \cap N(A)$ for $1 \leq i \leq r$, let $G' = G[N[A]]$, $S = N(A)$, and let $k' = |\bigcup_{i=1}^r V(Q_i)|$. Observe that (Q_1, Q_2, \dots, Q_r) is a feasible solution to the AUXILIARY LINKAGE instance $\mathcal{S}_P := (G[N[A]], k', S, (R_i)_{i=1}^r)$; the instance \mathcal{S}_P is henceforth called *induced by P and A* . Furthermore, it is easy to see that if $(Q'_1, Q'_2, \dots, Q'_r)$ is a different feasible solution to \mathcal{S}_P , then a path P' obtained from P by replacing every subpath Q_i with Q'_i is also a guarded k -path in G .

The crucial observation is that a small guard limits the number r of A -traverses.

Lemma 5 *The number of traverses of a guarded k -path P is bounded by $\max(1, 2|Z|)$.*

Proof Let r be the number of A -traverses. Every vertex of Z can be an endpoint of at most two traverses. If $r > 1$, then none of the traverses Q_i are contained in $G[A]$, and thus every traverse has at least one endpoint in the guard Z . \square

Lemma 5 in turn limits the number of possible instances \mathcal{S} that can be induced by a guarded k -path, for a fixed set A and guard Z . Note that we have $0 \leq k' \leq k$ and $0 \leq r \leq \max(1, 2|Z|)$. Furthermore, unless $r = 1$ and $R_1 = \emptyset$, we have $R_i \subseteq N(A)$, $|R_i| \in \{1, 2\}$, and every set R_i needs to have at least one element of Z ; there are at most $|Z| + |Z| \cdot |N(A)| = h(\ell + 1)$ choices for such a set R_i . Consequently, the number of possibilities for the instance \mathcal{S} is at most

$$(k + 1) \cdot \left(1 + \sum_{r=0}^{2h} h^r (\ell + 1)^r \right) \leq (k + 1) \cdot (h(\ell + 1))^{2h+1} =: \mathfrak{p}(k, \ell, h). \quad (1)$$

Reduction rule If $|A| > k \cdot \mathfrak{p}(k, \ell, h)$, then we can apply the following reduction rule. For each AUXILIARY LINKAGE instance \mathcal{S} out of at most $\mathfrak{p}(k, \ell, h)$ reasonable instances for A -traverses of a guarded k -path in G , we invoke an oracle on the instance \mathcal{S} , and mark the vertices of the solution if the oracle finds one. The whole process will mark at most $k \cdot \mathfrak{p}(k, \ell, h) < |A|$ vertices, thus at least one vertex of $|A|$ will remain unmarked. We delete any such vertices.

The observation that on a guarded k -path P one can replace a solution to the instance \mathcal{I}_P induced by P and A by a different solution ensures the safeness of this reduction. Finally, note that the reduction invokes at most $\mathfrak{p}(k, \ell, h)$ calls to the oracle; each call operates on a subgraph of the graph $G[N[A]]$ with $k' \leq k$ and $r \leq 2|Z|$.

We shall apply the Reduction Rule for a medium-sized set A and a guard set Z of constant size formed from adhesions of a tree decomposition. For most of the paper we will use $Z = N(A)$ with $\ell = h = |Z|$ a constant (depending on the excluded (topological) minor, in the results of Sects. 4 and 5). Only in Sect. 6, when dealing with a modulator M such that $G - M$ has an appropriate structure, it will be important to consider $N(A)$ potentially containing all of M , with a guard set Z of constant size disjoint from M .

3.3 Separation Oracles

The natural way of using our reduction rule is to find in a graph a large (but not too large) part of the graph with a small (preferably, constant) boundary. Let us first make an abstract definition of an algorithm finding such a separation.

Definition 6 For a graph class \mathcal{G} , a constant h , and a computable coordinate-wise nondecreasing function $q : \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$, an algorithm \mathcal{S} is called a $(h, q, T_{\mathcal{S}})$ -separation oracle if, given a graph $G \in \mathcal{G}$ and integers k and p , in time $T_{\mathcal{S}}(|G|, k, p)$ it finds a separation (A, B) in G of order at most h with $p < |A| \leq q(k, p)$, or correctly concludes that G contains a k -path.

For all considered graph classes, we will be able to provide a separation oracle with q being a polynomial. This, in turn, allows the following generic Turing kernel.

Lemma 7 Let \mathcal{S} be a $(h, q, T_{\mathcal{S}})$ -separation oracle for a hereditary graph class \mathcal{G} . Take $\hat{h} := (2h)^{4h+3}$. Then, the k -PATH problem restricted to graphs from \mathcal{G} can be solved:

- in time $\mathcal{O}(T_{\mathcal{S}}(|G|, k, k^2\hat{h}) \cdot |V(G)| + k\hat{h} \cdot |V(G)| \cdot |E(G)|)$,
- using at most $k\hat{h} \cdot |V(G)|$ calls to AUXILIARY LINKAGE
- each call on an induced subgraph of the input graph of size at most $q(k, k^2\hat{h})$.

Proof Let

$$p = k \cdot \mathfrak{p}(k, h, h) + h \leq k(k + 1)(h(h + 1))^{2h+1} + h \leq 2k^2(2h)^{4h+2} \leq k^2\hat{h}.$$

As long as $|V(G)| > p$, we proceed as follows. Invoke algorithm \mathcal{S} on G . If \mathcal{S} claims that G admits a k -PATH, we simply output the answer yes. Otherwise, let (A', B') be the separation output by \mathcal{S} . Apply the Reduction Rule for k , $A := A' \setminus B'$, and $Z = N(A) \subseteq A' \cap B'$. Note that as $|Z| \leq h$, the Reduction Rule deletes at least one vertex of A . Furthermore, the Reduction Rule invokes at most

$$\mathfrak{p}(k, h, h) \leq k\hat{h}$$

calls to the oracle, each call on an induced subgraph of G of size at most

$$|A'| \leq q(k, p) = q(k, k^2 \hat{h}).$$

Once we obtain $|V(G)| \leq p$, we solve the instance using a single call to AUXILIARY LINKAGE with $k' = k$, $r = 1$, and $R_1 = \emptyset$. The bounds follow, as there are at most $|V(G)|$ applications of the Reduction Rule, and each call to the oracle takes $\mathcal{O}(|E(G)|)$ time to prepare the instance and parse the output. \square

Note that for any graph class where separations as in Definition 6 exist, there exists a trivial separation oracle which finds them, running in time $n^{h+\mathcal{O}(1)}$: one iterates over every candidate for $A \cap B$ and, for fixed set $A \cap B$, a straightforward knapsack-type dynamic programming algorithm checks if one can assemble $A \setminus B$ of the desired size from the connected components of $G - (A \cap B)$.

However, this running time bound is unsatisfactory, as it greatly exceeds the number of used oracle calls. For all considered graph classes we prove a much stronger property than just merely the prerequisites of Lemma 7, in particular providing a more efficient separation oracle. We provide necessary definitions in the next section.

3.4 Decomposable Graph Classes

The following definition captures the key concept of this section.

Definition 8 For a constant h and a computable nondecreasing function $w: \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$, a graph class \mathcal{G} is called (w, h) -decomposable if for every positive integer k and every $G \in \mathcal{G}$ that does not admit a k -path, the graph G admits a tree decomposition of width less than $w(k)$ and adhesions of size at most h .

A standard argument shows that in a decomposable graph class, given the decomposition with appropriate parameters, it is easy to provide a separation oracle.

Lemma 9 Assume we are given a graph G and a tree decomposition (T, \mathcal{X}) of G of width less than w , adhesion size at most h , and adhesion degree at most $a \geq 2$. Then, given an integer p such that $|V(G)| > p$, one can in time $h^{\mathcal{O}(1)} \cdot (|V(G)| + |E(G)| + |V(T)| + \sum_{t \in V(T)} |\mathcal{X}(t)|)$ find a separation (A, B) of order at most h such that

$$p < |A| \leq w + p \cdot a.$$

Proof Root the tree T in an arbitrary node, and for $t \in V(T)$ let T_t be the subtree of T rooted in t . Let t_0 be a lowest node of T such that $|\mathcal{X}(V(T_{t_0}))| > p$; such a node can be computed in linear time in the size of G and (T, \mathcal{X}) .

Group the children t' of t_0 according to their adhesions $\mathcal{X}(t') \cap \mathcal{X}(t_0)$. Due to the bound on the adhesion degree, there are at most a groups. For every adhesion S , let $X_{t_0,S}$ be the set of the children of t_0 with $S = \mathcal{X}(t') \cap \mathcal{X}(t_0)$. Define

$$V_S = \bigcup_{t' \in X_{t_0,S}} \mathcal{X}(T_{t'}).$$

We consider now two cases. First, assume that $|V_S| \leq p$ for every adhesion S . Then, by the adhesion degree bound, we have

$$|\mathcal{X}(T_{t_0})| \leq |\mathcal{X}(t_0)| + ap \leq w + ap.$$

Consequently, we can return the separation (A, B) with $A = \mathcal{X}(T_{t_0})$ and $B = \mathcal{X}(T - V(T_{t_0}))$.

In the other case, there exists an adhesion S with $|V_S| > p$. We greedily take a minimal subset $Y_{t_0,S} \subseteq X_{t_0,S}$ such that

$$V'_S := \bigcup_{t' \in Y_{t_0,S}} \mathcal{X}(T_{t'})$$

is of size greater than p . By the minimality of t_0 , for every $t' \in X_{t_0,S}$ we have $|\mathcal{X}(T_{t'})| \leq p$ and, consequently $|V'_S| \leq 2p$. Thus, we can return the separation (A, B) for $A = V'_S$ and $B = N_G[V(G) \setminus V'_S]$, as then $A \cap B \subseteq S$. \square

To apply the above lemma for a (w, h) -decomposable class, we need to compute a decomposition approximating these parameters, which takes the remainder of this section. A critical insight is that the decomposition used by Cygan et al. [10] to solve the MINIMUM BISECTION problem provides this. The decomposition is based on the notion of an *unbreakable* vertex set X . Intuitively, a set $X \subseteq V(G)$ is (q, h) -unbreakable in G if it is impossible to break X into two pieces of size more than q each, by removing only h vertices. The formal statement in terms of separations is as follows.

Definition 10 A vertex set $X \subseteq V(G)$ of a graph G is called (q, h) -unbreakable if every separation (A, B) of order at most h satisfies $|(A \setminus B) \cap X| \leq q$ or $|(B \setminus A) \cap X| \leq q$.

Let us now recall the main technical result of [10].

Theorem 11 [10] *There is an algorithm that, given a graph G and integer h , runs in time $2^{\mathcal{O}(h^2)}|V(G)|^2|E(G)|$ and outputs a connected tree decomposition (T, \mathcal{Y}) of G such that:*

1. *for each $t \in V(T)$, the bag $\mathcal{Y}(t)$ is $(2^{\mathcal{O}(h)}, h)$ -unbreakable in G , and*
2. *for each $tt' \in E(T)$ the adhesion $\mathcal{Y}(t) \cap \mathcal{Y}(t')$ has at most $2^{\mathcal{O}(h)}$ vertices and is $(2h, h)$ -unbreakable in G .*

Lemma 12 *Let G be a graph and suppose there exists a tree decomposition (T, \mathcal{X}) of G of width less than w , adhesion size h , and adhesion degree a . Let (T', \mathcal{Y}) be a tree decomposition of G and $t \in V(T')$ such that the bag $\mathcal{Y}(t)$ is $(2^{\mathcal{O}(h)}, h)$ -unbreakable in G . Then $|\mathcal{Y}(t)| \leq w + a \cdot 2^{\mathcal{O}(h)}$.*

Proof Consider such a bag $Y = \mathcal{Y}(t)$ of the second decomposition, or in fact any $(f(h), h)$ -unbreakable set $Y \subseteq V(G)$, where $f(h) = 2^{\mathcal{O}(h)}$. We orient every edge of T toward Y in the following sense. For an edge $tt' \in E(T)$ of the first decomposition,

removing it partitions T into subtrees $T_{t't}$ and $T_{tt'}$ containing t' and t , respectively. The induced separation $(\mathcal{X}(T_{t't}), \mathcal{X}(T_{tt'}))$ on G has order at most h , so one of the sides, say $\mathcal{X}(T_{t't}) \setminus \mathcal{X}(t)$, contains at most $f(h)$ vertices of Y , by definition of unbreakability. We orient the edge tt' away from t' (the ‘small’ side). After orienting each edge of T , since T is acyclic, we can find a node t_0 of T such that all incident edges point to it. That is, for every neighbor t' of t_0 , we have $|Y \cap \mathcal{X}(T_{t't_0}) \setminus \mathcal{X}(t_0)| \leq f(h)$. We now want to bound $|Y \setminus \mathcal{X}(t_0)|$.

While t_0 can have many neighbors, we can group those neighbors t' based on the adhesion $\mathcal{X}(t') \cap \mathcal{X}(t_0)$ to which they correspond. By the adhesion degree bound, there are at most a such groups. If for any such group $U \subseteq N(t_0)$, the union $\bigcup_{t' \in U} \mathcal{X}(T_{t't_0}) \setminus \mathcal{X}(t_0)$ contained more than $3 \cdot f(h)$ vertices of Y , then the group can be partitioned into two parts with more than $f(h)$ vertices of Y each. This would give a separation of order h of G with too many vertices of Y on both sides, contradicting its unbreakability. Therefore, there are at most a groups, each containing at most $3 \cdot f(h)$ vertices of Y , thus the size of Y is bounded by $|\mathcal{X}(t_0)| + a \cdot 3 \cdot f(h)$. \square

By definition, a graph G in a (w, h) -decomposable graph class that admits no k -path has a decomposition (T, \mathcal{X}) of width less than $w(k)$ and adhesion size h . The trivial bound on its adhesion degree is $a := \sum_{i \leq h} \binom{w(k)}{i} \leq (w(k) + 1)^h$. Given G and h , the algorithm of Theorem 11 yields an ‘unbreakable’ decomposition (T', \mathcal{X}') of adhesion size $2^{\mathcal{O}(h)}$. By the above lemma, its width is bounded by $w(k) + a \cdot 2^{\mathcal{O}(h)} \leq (w(k) + 1)^{\mathcal{O}(h)}$.

Corollary 13 *Let \mathcal{G} be a (w, h) -decomposable graph class. Then, for every $G \in \mathcal{G}$ and every integer k , one can in $2^{\mathcal{O}(h^2)} |V(G)|^2 |E(G)|$ time either correctly conclude that G admits a k -path, or find a tree decomposition of G of width $(w(k) + 1)^{\mathcal{O}(h)}$ and adhesion size $2^{\mathcal{O}(h)}$.*

To conclude our Turing kernel, we can bound the adhesion degree of the computed decomposition again trivially by $(w(k) + 1)^{2^{\mathcal{O}(h)}}$. We use this decomposition to find separations of any induced subgraphs of G using the algorithm of Lemma 9 in time $2^{\mathcal{O}(h)}$ times linear in the size of G and the computed decomposition. This gives a (h, q, T) -separation oracle with $q(k, p) = p \cdot (w(k) + 1)^{2^{\mathcal{O}(h)}}$ and $T(n, k, p) = 2^{\mathcal{O}(h)} \cdot n \cdot (w(k) + 1)^{\mathcal{O}(h)}$, for any (w, h) -decomposable graph class. By plugging it into Lemma 7 (which additionally requires the class to be hereditary), we obtain the following.

Corollary 14 *Let \mathcal{G} be a hereditary (w, h) -decomposable graph class. Then the k -PATH problem restricted to graphs from \mathcal{G} can be solved in time $2^{2^{\mathcal{O}(h)}} |V(G)|^2 |E(G)|$ using $2^{2^{\mathcal{O}(h)}} kn$ calls to AUXILIARY LINKAGE on induced subgraphs of the input graph of size $k^2(1 + w(k))^{2^{\mathcal{O}(h)}}$.*

In the next section, we prove that H -minor-free graphs are $(\mathcal{O}_H(k^{22}), \mathcal{O}_H(1))$ -decomposable by analyzing the Global Structure Theorem of H -minor-free graphs due to Robertson and Seymour. A subsequent section provides an analogous result for graphs excluding a fixed topological minor. In both cases we also get better bounds

on the adhesion degree of the approximate decomposition outputted by Theorem 11, improving the bounds in the final kernel.

We would like to remark that we do not want to claim in this paper the idea that, in the context of H -(topological)-minor-free graphs, the decomposition of Theorem 11 should be related to the decomposition of the Global Structure Theorem via an argument as in the proof of Lemma 12. In particular, this observation appeared previously in a work of the second author with Daniel Lokshtanov, Michał Pilipczuk, and Saket Saurabh [24].

4 Excluding a Minor

In this section we tackle proper minor-closed graph classes, that is, we prove Theorem 1 for graph classes excluding a fixed minor, by proving the following.

Theorem 15 *For every graph H , the k -PATH problem restricted to H -minor-free graphs can be solved in time $\mathcal{O}_H(n^2m)$ using $\mathcal{O}_H(kn)$ calls to AUXILIARY LINKAGE on instances being induced subgraphs of the input graph of size $\mathcal{O}_H(k^{24})$.*

Our main technical result is the following:

Theorem 16 *For every graph H , the class of H -minor-free graphs is (w, h) -decomposable for $w(k) = \mathcal{O}_H(k^{22})$ and $h = \mathcal{O}_H(1)$.*

By plugging the above into Corollary 14, we obtain the desired polynomial Turing kernel, but with worse bounds than promised by Theorem 15. To obtain better bounds, we need to recall the folklore bound on the adhesion degree in sparse graph classes; for completeness, we provide a full proof in “Appendix A.1”.

Lemma 17 *Let G be a graph not containing H as a topological minor, and let (T, \mathcal{X}) be a connected tree decomposition of G of width less than ℓ and adhesion size h . Then the adhesion degree of (T, \mathcal{X}) is bounded by $f(h, H) \cdot \ell$ for some integer $f(h, H)$ depending only on h and H .*

This way, we conclude that H -minor-free graphs without k -paths have tree decompositions of width $\mathcal{O}_H(k^{22})$, adhesion size $\mathcal{O}_H(1)$ and adhesion degree $\mathcal{O}_H(k^{22})$. We can use the algorithm of Theorem 11 for h to find (by Lemma 12) an approximate decomposition of width $\mathcal{O}_H(k^{22})$, adhesion size $\mathcal{O}_H(1)$, and thus, again using Lemma 17, of adhesion degree $\mathcal{O}_H(k^{22})$. Theorem 15 follows from Lemma 7 if we find separations using the algorithm of Lemma 9 applied to this decomposition.

Thus, it remains to prove Theorem 16. For the proof, we use the graph minors structure theorem, decomposing an H -minor-free graph G into parts ‘nearly embeddable’ in surfaces (precise definitions are given in the next subsection). By carefully analyzing details of the structure, we either find a large triconnected embedded part, which must contain a long path by the following theorem of Chen et al. [9], or we tighten the graph structure to give a tree decomposition where all parts are small (polynomial in k) and adhesions (‘boundaries’) between them are of constant size.

Theorem 18 [9] *There is a constant $\varepsilon > 0$ such that for every integer t , every triconnected graph on $n \geq 3$ vertices embeddable in a surface of (Euler) genus g contains a cycle of length at least $n^\varepsilon / 2^{(2g+3)^2}$.*

We note that Chen et al. phrase the theorem (more generally) for $K_{3,t}$ -minor-free graphs, but a folklore edge-counting argument shows that graphs embeddable on a surface of genus g are $K_{3,t}$ -minor-free for $t = 2g + 3$ (see e.g. [2]).

Two intertwined problems that arise with this approach is that torsos of decompositions are not necessarily triconnected, and long paths in them do not necessarily imply long paths in the original graph, because of virtual edges added in torsos. Torsos can be made triconnected if their near-embeddings include cycles or paths around each vortex, but these may use virtual edges in essential ways. On the other hand, the decomposition can be modified so that virtual edges can be replaced with paths in the original graph, but this requires changes that remove virtual edges, hence potentially removing paths around vortices and destroying triconnectedness.

Because of that, we need to go a little deeper and use a local, strong version of the structure theorem from Graph Minors XVII [28]. For the same reason we cannot use existing algorithms for finding the graph minors decompositions. Instead, we only prove the *existence* of a tree decomposition of bounded size adhesions, small width, and with nearly embeddable bags.

Global and Local Graph Minor Structure Theorems

The major result proved by Robertson and Seymour in their *Graph Minors* series of papers is the Global Structure Theorem, which describes the structure of large graphs excluding a fixed graph as a minor. Before we summon the formal statement, let us recall a basic overview. We refer the reader to Diestel’s book [11] for basic definitions (Section 12.4 in [11]) and a description of how surfaces (orientable or not) are classified by their genus (Appendix B in [11]).

The Global Structure Theorem states that for every graph H , there is a constant $\alpha = \alpha(H)$ such that every H -minor free graph G admits a tree decomposition whose adhesions have size at most α , and whose torsos are possibly large, but “ α -nearly embeddable”. Recall that a torso in a decomposition (T, \mathcal{X}) is a graph obtained from a bag $\mathcal{X}(t)$ (for some $t \in V(T)$) by taking $G[\mathcal{X}(t)]$ and turning each adhesion $\mathcal{X}(t) \cap \mathcal{X}(t')$ (for $t' \in N_T(t)$) into a clique (adding a *virtual edge* between every two vertices in an adhesion, if they were not already connected). Informally, “ α -nearly-embeddable” means that the torso can be obtained as follows:

- draw a graph on a surface of constant (i.e. at most α) genus (called the *embedded part*),
- for a constant number of faces in the drawing, add a graph of constant pathwidth “inside the face” (called the *vortices*),
- add a constant number of vertices with arbitrary connections (called the *apices*).

The different constant bounds are essentially unrelated, but we prefer to use only one constant α for all of them, since we are not interested in the exact dependency on H . Let us now formally define near-embeddability and the graph minors α -decomposition.

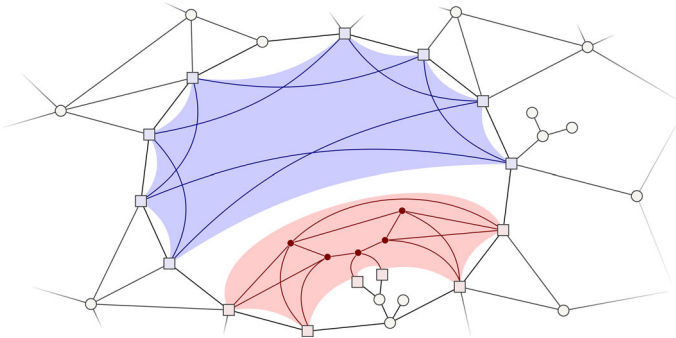


Fig. 2 A fragment of a nearly embedded graph with two vortices: top (blue) and bottom (red). Society vertices are squares. Note that a ‘disk’ on a surface means a subset homeomorphic to a closed disk in the plane (Color figure online)

Definition 19 For an integer α , an α -near embedding of a graph G consists of:

- (i) a set $A \subseteq V(G)$ of at most α vertices (called the *apex set*);
- (ii) a family $G_0 \cup G_1 \cup \dots \cup G_{\alpha'} = G \setminus A$ of edge-disjoint subgraphs of $G \setminus A$, where:
 - G_0 is called the *embedded part*,
 - $\mathcal{V} = \{G_1, \dots, G_{\alpha'}\}$ for some $\alpha' \leq \alpha$ are called (*large*) *vortices*,
 - the intersection $\Omega_i := V(G_0) \cap V(G_i)$ is called the *society* of vortex G_i ,
 - vortices are pairwise vertex-disjoint,
- (iii) an embedding of G_0 in a surface of genus at most α such that for $i \in [\alpha']$, the society Ω_i is embedded on the boundary of a disk whose interior is empty (i.e., does not intersect the embedding nor other disks), called the disk *accommodating* G_i ;
- (iv) a linear ordering $w_1, \dots, w_{|\Omega_i|}$ of each vortex society Ω_i , corresponding to its natural ordering around its disk (for some choice of direction and starting point);
- (v) for each large vortex $G_i \in \mathcal{V}$, a path decomposition $\mathcal{L}_1, \dots, \mathcal{L}_{|\Omega_i|}$ of G_i of width at most α such that $w_j \in \mathcal{L}_j$, for $j \in [|\Omega_i|]$.

We denote such an α -near embedding as (A, G_0, \mathcal{V}) , with the embedding and path decompositions only implicit in the notation.

See Fig. 2 for an illustration.

Definition 20 A (*graph minors*) α -decomposition of a graph G consists of:

- a rooted tree decomposition (T, \mathcal{X}) of G of adhesion size at most α ;
- for each $t \in V(T)$, an α -near embedding $(A_t, G_{t0}, \mathcal{V}_t)$ of $\text{TORSO}(G, \mathcal{X}(t))$.

Theorem 21 (Global Structure Theorem) *For every graph H , there is a constant $\alpha = \alpha(H)$ such that every H -minor-free graph admits an α -decomposition.*

In this section we shall prove the following variant of the Global Structure Theorem, which in particular implies Theorem 16, concluding our main result for minor-closed classes.

Theorem 22 For every graph H , there is a constant $\alpha = \alpha(H)$ such that the following holds, for any integer k : any graph G excluding H as a minor and without a k -path has an α -decomposition of width at most $\alpha \cdot k^{22}$.

Unfortunately we do not see a way to obtain this variant directly, just by adapting the original Global Structure Theorem. However, we can adapt a local version of it. Indeed, the proof of the Global Structure Theorem by Robertson and Seymour proceeds by analyzing only a single part of the graph (in fact the whole graph need not be H -minor-free), improving its structure in a number of technical steps to obtain the so-called Local Structure Theorem. Most of the analysis is hence done locally and it is only at the end that one proceeds with a fairly general inductive argument to conclude the Global Structure Theorem.

Informally, the Local Structure Theorem, instead of saying that the whole graph has a tree decomposition into nearly-embeddable torsos, only states that a single subgraph can be nearly-embedded so that its boundary (vertices with neighbors outside of this subgraph) is contained in vortices or triangular faces (called *small vortices*). Proving that this holds for every ‘part’ of the graph allows to inductively show that the remaining graph behind the vortices is also structured, which is how the tree decomposition of the Global Structure Theorem is obtained. Formalizing what is meant by ‘every part’ is a bit more technical (the somewhat nebulous notion of tangles is used for that).

This will also be our approach: we will improve the Local Structure Theorem to conclude that either the graph has a k -path, or that the embedded part is polynomially bounded. The general inductive argument to deduce the Global Structure Theorem will then work unchanged and the improvements will carry through, concluding Theorem 22 (essentially, each bag of the Global Structure Theorem’s decomposition comes directly from some nearly-embedded part from the Local Structure Theorem, so the bound on the embedded part will imply the bound on each of the final bags).

An α -near local embedding of a graph G is defined similarly to an α -near embedding, but we allow an arbitrary number of ‘small vortices’ \mathcal{W} and we allow the path decompositions of all vortices to have only bounded adhesion size instead of bounded width; thus arbitrarily complicated graphs can hide behind vortices. Formally:

Definition 23 For an integer α , an α -near local embedding of graph G consists of:

- (i) a set $A \subseteq V(G)$ of at most α vertices (called the *apex set*);
- (ii) a family $G_0 \cup G_1 \cup \dots \cup G_n = G \setminus A$ of edge-disjoint subgraphs of $G \setminus A$, where:
 - G_0 is called the *embedded part*,
 - $\mathcal{V} = \{G_1, \dots, G_{\alpha'}\}$ for some $\alpha' \leq \alpha$ are called *large vortices*,
 - $\mathcal{W} = \{G_{\alpha'+1}, \dots, G_n\}$ are called *small vortices*,
 - vortices intersect only in G_0 : $V(G_i) \cap V(G_j) \subseteq V(G_0)$ for $i \neq j \in [n]$,
 - the intersection $\Omega_i := V(G_0) \cap V(G_i)$ is called the *society* of vortex G_i ,
 - large vortices are pairwise vertex-disjoint,
 - small vortices have societies of size ≤ 3 ;
- (iii) an embedding of G_0 in a surface of genus at most α such that for $i \in [n]$, the society Ω_i is embedded on the boundary of a disk whose interior is empty (i.e., does not intersect the embedding nor other disks), called the disk *accommodating* G_i ;

- (iv) a linear ordering $w_1, \dots, w_{|\Omega_i|}$ of each society Ω_i , corresponding to its natural ordering around its disk (for some choice of direction and starting point);
- (v) for each large vortex $G_i \in \mathcal{V}$, a path decomposition $\mathcal{Z}_1, \dots, \mathcal{Z}_{|\Omega_i|}$ of G_i of adhesion size $\leq \alpha$ such that $\mathcal{Z}_j \cap \Omega_i = \{w_{j-1}, w_j\}$ for $1 < j \leq |\Omega_i|$ and $\mathcal{Z}_1 \cap \Omega_i = \{w_1\}$.

We denote such an α -near local embedding as $(A, G_0, \mathcal{V}, \mathcal{W})$, with the embedding and path decompositions only implicit in the notation.

We now recall the definition of tangles. Intuitively, a tangle defines what we’re looking at by pointing every small-order separation “towards it”. For an integer θ , a *tangle of order θ* in a graph G is a family \mathcal{T} of separations of order at most θ such that:

- for each separation (X, Y) of order at most θ in G , exactly one of (X, Y) and (Y, X) belongs to \mathcal{T} . If $(X, Y) \in \mathcal{T}$ we call X the *small side* of the separation and Y the *large side*.
- For every three separations $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3)$ in \mathcal{T} , we have $G[X_1] \cup G[X_2] \cup G[X_3] \neq G$. That is, three small sides cannot cover the whole graph.

Informally, the Local Structure Theorem states for every graph G and every tangle \mathcal{T} that does not “point towards” an H -minor in G , the graph G has an α -near local embedding such that the tangle “points towards” the embedded part, away from vortices. The formal statement requires the following definitions.

Definition 24 Let G and H be graphs. An *H-minor model* in G is a family of *branch sets* $(B_h)_{h \in V(H)}$ inducing vertex-disjoint, connected subgraphs in G such that for each edge $hh' \in E(H)$, there is an edge between B_h and $B_{h'}$ in G .

A tangle \mathcal{T} in G *controls an H-minor* if there is an H -minor model in G such that no branch set is fully contained in a small side of a separation in \mathcal{T} .

An α -near local embedding $(A, G_0, \mathcal{V}, \mathcal{W})$ of G is said to *respect* a tangle \mathcal{T} if for every large side Y of a separation in $(X, Y) \in \mathcal{T}$, $Y \setminus A$ is not contained in a small vortex $W \in \mathcal{W}$ or in a bag of the decomposition of a large vortex $V \in \mathcal{V}$.

Theorem 25 (Local Structure Theorem, [14,28]) *For every graph H there exist integers α, θ such that: for every graph G and every tangle \mathcal{T} in G of order $\geq \theta$ that controls no H -minor, there is an α -near local embedding of G which respects \mathcal{T} .*

The statement comes from [14] (Theorem 1). It follows from [28], though that paper uses different notation and is written in a way that requires the reader to understand all the paper to use its conclusions. We refer the reader to [14] for a detailed explanations of how [28] translates to this more modern language. We note that while [14] assumes that the whole graph G is H -minor-free, the original statement in [28] only requires that \mathcal{T} controls no H -minor; we will need this stronger version when dealing with topological minors.

Improving the Structure Theorems

Our aim is to improve the Local Structure Theorem 25 to say that the embedded part either contains a k -path or has size polynomially bounded in k . This would follow

immediately from Theorem 18 if the embedded part was triconnected. The remainder of this section hence focuses on working around the technical issue that we can only ensure this by involving vortices.

Indeed, if the embedded part is not triconnected, there is a separation (X, Y) of order at most 2: we would then like to limit the near-embedding to only Y , say (if Y is the large side of the separation (X, Y) in the tangle) and hide X in a new small vortex. We would then repeat this until the embedded part is triconnected. This simple idea is essentially applied as one of the improvement steps in [28]. However, a separation of the embedded part may not extend to a separation of the whole graph, as it could require to cut through a vortex. Because of that, triconnectivity can only be shown for an auxiliary graph defined from the embedded part by adding a few ‘virtual’ edges and vertices to make the society of each vortex triconnected (formal definitions follow below). Still, these virtual edges and vertices can be added to the embedding, so Theorem 18 applies.

Therefore, if the embedded part is large, we will find a long path in it, except that it might use these virtual edges and vertices added to vortex societies. To show that this still implies a k -path in the original graph, we need to strengthen the Local Structure Theorem to ensure that society vertices are connected by appropriate paths (with which virtual edges will be replaced). Indeed, in a local near-embedding one can always replace each edge of the embedded part with a small vortex (with two society vertices, one edge inside the vortex, and an accommodating disk around where the edge was originally embedded), so the embedded part can be edgeless! This is why some connectivity conditions for vortices are necessary.

Such conditions follow from [28] (in fact the results there improve the structure of a vortex significantly, in particular with a very strong connectivity condition, explaining the title ‘Taming a Vortex’). Unfortunately not all of them are stated in the statement we cite [14], so we need to explain where they come from by referring to the original terminology in [28], but we limit these references to the next three lemmas. The reader may skip their proofs and proceed to the only new contribution of this subsection: Lemma 31.

We now proceed with formal definitions of these connectivity conditions. For a large vortex, we require that its society is connected by a ‘comb’ (Fig. 3) that avoids other large vortices:

Definition 26 A *comb* is a union of a path P with some mutually vertex-disjoint (possibly trivial) paths that have their first vertex and no other vertex on P . The last vertices of those paths are called the *teeth* of the comb, they are naturally ordered by P .

Lemma 27 *In Theorem 25, we can additionally assume that:*

(vi) *for each large vortex $G_i \in \mathcal{V}$, there is a comb C_i in $G_i \cup G_0 \cup \bigcup \mathcal{W}$ whose teeth are vertices $w_1, \dots, w_{|\Omega_i|}$ of Ω_i , in the same order as in (iv).*

Proof This condition is in fact already included in the statement we cite from [14] (in point 4. of the definition of α -near-embedding, which we call an α -near *local* embedding for clarity). \square

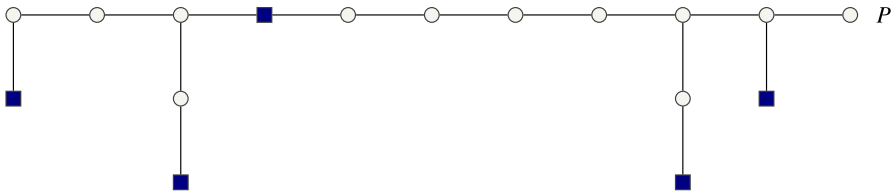


Fig. 3 A comb with five teeth

We stress that the comb C_i contains a path of length at least $|\Omega_i|$. This condition is useful for us, as a long vortex (i.e. with $\geq k$ society vertices) will imply this path to be long, so we only need to continue if all societies are relatively small.

For a small vortex, we require its society vertices to be connected by paths that are strictly inside the vortex:

Lemma 28 *In Lemma 27, we can additionally assume that:*

- (vii) *for each small vortex $G_i \in \mathcal{W}$ and every $v_1, v_2 \in \Omega_i$, there is a path in G_i between v_1 and v_2 , with no internal vertices in Ω_i .*

Proof Property (vii) is guaranteed by (9.1) in [28]. (As explained in [14], the border cells of [28] are translated into bags of large vortex decompositions, so small vortices arise only from internal cells; for an internal cell c_0 , $\alpha(c_0)^*$ is translated into a small vortex G_i , while $\gamma(\tilde{c}_0)$ is translated into Ω_i , giving exactly the statement we want.) Alternatively, we can repeat the proof of (9.1): simply partition a vortex with no v_1 - v_2 path into two vortices with at most two vertices in their societies; recall that societies of small vortices are allowed to intersect. □

Next, we define the graph obtained from the embedded part by adding virtual edges and vertices around vortices: this is the graph that can be shown to be triconnected, yet can still be embedded in the same surface. (In contrast, recall that the embedded part G_0 can be edgeless.)

Definition 29 For an α -near local embedding $(A, G_0, \mathcal{V}, \mathcal{W})$ of G , define G_0^* to be the graph obtained from G_0 by adding an edge between every two consecutive vertices in each society Ω_i , and a new vertex for each society of a large vortex, with edges to every vertex of the society. The embedding of G_0 is naturally extended to an embedding of G_0^* : the new vertices and edges are embedded in place of the accommodating disks.

Lemma 30 *In Lemma 28, we can additionally assume that:*

- (viii) *G_0^* is triconnected.*

Proof Suppose G_0^* has a separation (X, Y) of order ≤ 2 . Since societies of small vortices induce cliques in G_0^* , they are contained in X or Y . Similarly for each large vortex $G_i \in \mathcal{V}$, its society Ω_i , together with the new vertex in G_0^* with Ω_i as neighborhood, induces a wheel in G_0^* , which is triconnected, hence contained in X or Y . In the language of [28], this translates back into a partition I, J of cells, whose intersection corresponds to at most two embedded vertices v, v' (that is, $\gamma(I) \cap \gamma(J) = X \cap Y = \{v, v'\}$). But this contradicts (11.1) in [28], which states that such an intersection has size at least 3. □

From these connectivity conditions, we are able to deduce what we need from the Local Structure Theorem:

Lemma 31 *In Lemma 30, we can additionally assume that there is a constant α' depending on H only such that:*

(ix) *for any integer k , if G does not contain a path of length $\geq k$, then $|V(G_0)| < \alpha' \cdot k^{22}$.*

Proof For a given graph H , let α be the constant given by Theorem 25. By Theorem 18, there is a universal constant ε such that any triconnected graph G_0 embeddable in a surface of genus at most α contains a cycle of length at least $|G_0|^\varepsilon / 2^{(2\alpha+2)^2}$. Let $c := \frac{2}{\varepsilon}$ and $\alpha' := (\alpha \cdot 2^{(2\alpha+2)^2})^{1/\varepsilon}$. For an H -minor-free graph G , suppose the α -near local embedding given by Lemma 30 has $|V(G_0)| \geq \alpha' \cdot k^c$. Then G_0^* is a triconnected graph (by property (viii)) embedded in a surface of genus at most α (by Definition 29) with at least as many vertices, so G_0^* must contain a cycle C of length at least $(\alpha' k^c)^\varepsilon / 2^{(2\alpha+2)^2} = \alpha \cdot k^2$.

If the society Ω_i of any vortex of the α -near local embedding of G has at least k vertices, then G contains a path of at least that length (in the comb from property (vi)), in which case the lemma follows. Otherwise, there are at most α large vortices, hence at most $\alpha \cdot (k - 1)$ vertices in their societies, and at most α virtual vertices that were inserted as the centers of the wheels. Let P be the longest subpath of C avoiding these $\alpha \cdot (k - 1) + \alpha$ vertices. Then P is a path of length at least $\frac{\alpha \cdot k^2}{\alpha \cdot k} = k$ in G_0^* whose edges in $E(G_0^*) \setminus E(G_0)$ came only from small vortices, by the definition of G_0^* (Definition 29). By property (vii), these edges can be replaced with paths inside the corresponding small vortices, giving a path in G . Only one or two consecutive edges can come from the same small vortex (since their societies have at most 3 vertices), so the resulting path has length at least k .

The constant from Theorem 18 given by Chen et al. [9] is $\varepsilon = \log_{1729} 2$, hence the universal constant here is $c = \frac{2}{\varepsilon} = 2 \log_2(1729) < 22$. \square

Finally, we deduce Theorem 22 (our variant of the Global Structure Theorem) from the improved variant of Theorem 25 (the Local Structure Theorem) by a standard induction, exactly as done by Diestel et al. [12, Theorem 4]. (Note that in case the excluded graph H is planar, we can already conclude the theorem trivially from the fact that there is a tree-decomposition of width $\mathcal{O}_H(1)$ [26]). The only difference is that in the Local Structure Theorem we add the bound on the size of the embedded part from Lemma 31. In the proof from [12], every bag of the created decomposition is either constructed as a set of size bounded as $\mathcal{O}_H(1)$, or is constructed from an α -near local embedding by taking the vertices of: the embedded part G_0 , the set of apices A , and for each large vortex, the intersection of every two consecutive bags of its path decomposition, each of size at most $\alpha = \mathcal{O}_H(1)$. The number of bags in a path decomposition of a large vortex is equal to the size of its society, and large vortex societies are disjoint subsets of $V(G_0)$. Therefore every tree decomposition bag constructed in the proof has size at most $|V(G_0)| + \alpha + \alpha \cdot |V(G_0)| = \mathcal{O}_H(k^{22})$, by Lemma 31. This proves the additional condition we require in Theorem 22.

5 Excluding a Topological Minor

In this section we tackle graph classes excluding a topological minor, that is, we prove Theorem 1 by proving the following.

Theorem 32 *For every graph H , the k -PATH problem restricted to H -topological-minor-free graphs can be solved in time $\mathcal{O}_H(n^2m)$ using $\mathcal{O}_H(kn)$ calls to AUXILIARY LINKAGE on instances being induced subgraphs of the input graph of size $k^{\mathcal{O}_H(1)}$.*

This follows as before from the following decomposition theorem. Note the exponent in the polynomial bound on the width (bag size) now depends on H .

Theorem 33 *For every graph H , the class of H -topological-minor-free graphs is (w, h) -decomposable for $w(k) = k^{\mathcal{O}_H(1)}$ and $h = \mathcal{O}_H(1)$.*

We proceed with the proof of the above theorem. Grohe and Marx [18] proved that when excluding a topological minor, graphs admit a similar decomposition as for excluding a minor, but apart from nearly embeddable parts, one needs to consider parts that have bounded degree except for a bounded number of vertices:

Definition 34 A graph G has (a, d) -bounded degree (for $a, d \in \mathbb{N}$) if $\leq a$ vertices of G have degree $> d$ in G .

The main ‘Global Structure Theorem’ of [18] is that graphs excluding a topological minor H admit a tree decomposition with bounded size adhesions, where the torso of every bag is either nearly embeddable (as when excluding a minor), or has (h, d) -bounded degree, for $h, d = \mathcal{O}_H(1)$. We still need the slightly stronger notion of near embeddability (with combs) for our proofs, which is why we will work with the following definition of ‘Local Structure’, analogous to α -near local embeddings.

A *star decomposition* of a graph is a (rooted) tree decomposition $(\mathcal{T}, \mathcal{X})$ where T is a star: a root node t_{root} with only leaf nodes attached. We call the bag $\mathcal{X}(t_{\text{root}})$ the *root bag* and the remaining ones *leaf bags*. The *root torso* is $\text{TORSO}(G, \mathcal{X}(t_{\text{root}}))$. A star decomposition *respects* a tangle \mathcal{T} if the large side of any separation in \mathcal{T} is not fully contained in any single *leaf bag*. Grohe and Marx [18, Lemma 6.12] proved the following Local Structure Theorem.²

Theorem 35 [18] *For every integer k there exist integers θ, d, k' such that: for every graph G excluding K_k as a topological minor and every tangle \mathcal{T} of order $\geq \theta$ that controls a $K_{k'}$ -minor, there is a star decomposition of G that respects \mathcal{T} , has adhesion size $< \theta$, and such that the root torso has (k, d) -bounded degree.*

Corollary 36 *For every graph H there exist integers θ, α, d such that: for every graph G excluding H as a topological minor and every tangle \mathcal{T} in G of order $\geq \theta$, the graph G has one of the following:*

² Grohe and Marx [18] use a slightly stronger definition of *respecting a tangle*, which is however equivalent to the conjunction of this one and the assertion that all adhesions in the star decomposition are strictly smaller than the order of \mathcal{T} , which we prefer to state separately. Their theorem also works even under a slightly weaker condition than that of a tangle controlling a minor (which they call *not being removed* by the tangle).

- a star decomposition that respects \mathcal{T} , has adhesion size α , and such that the root torso has $(|V(H)|, d)$ -bounded degree, or
- an α -near local embedding that respects \mathcal{T} , satisfying properties (vi)–(ix).

Proof For $k = |V(H)|$ let the constants given by Theorem 35 be θ', d', k' . Let α'', θ'' be the constants given by Theorem 25 for graphs excluding $K_{k'}$ as a minor. We prove the claim for $\theta = \max(\theta', \theta'')$, $\alpha = \max(\theta', \alpha'')$ and $d = d'$. If the tangle \mathcal{T} of G controls no $K_{k'}$ -minor, then the claim follows from Theorem 25. Otherwise \mathcal{T} does control a $K_{k'}$ -minor and Theorem 35 applies. \square

The main theoretical tool we use, besides the Grohe and Marx decomposition in the form of Corollary 36, is the following lower bound on cycles in graphs of bounded degree by Shan [30] (we note the constant $\Delta - 1$ is asymptotically optimal; for small Δ , previous results by Chen et al. [7] give a slightly better bound).

Theorem 37 [30] *If G is a triconnected graph with maximum degree at most $\Delta \geq 425$, then G has a cycle of length at least $n^{1/\log_2(\Delta-1)}/4 + 2$.*

With this in hand, we will bound the bags of almost bounded degree as follows:

Lemma 38 *In Corollary 36, we can additionally assume that in the first outcome, for any integer k , if G has no k -path, then the root bag has size $k^{\mathcal{O}_H(1)}$.*

Before we prove this, let us show how this concludes our main result. The global decomposition stated in Theorem 33 (and hence the polynomial Turing kernel) follows, as before, from the Local Structure Theorem in Corollary 36 improved by the bound on bag size in Lemma 38. The construction only changes in that it can take root bags from the local structure (the star decomposition) given by the first outcome of Corollary 36 and use them as bags of the global decomposition, without any modifications. (We note this can also be seen as an instance of a general theorem (11.1) by Robertson and Seymour [27] which allows to turn ‘local decompositions’ with respect to a tangle into global ones; however, giving a precise translation would involve too much notational overhead.)

The remainder of this section is devoted to proving Lemma 38. That is, assuming a star decomposition with a root of almost bounded degree, we want to show that it contains a k -path or that we can improve the star decomposition so that the root bag has size polynomial in k . Similarly as before, this would follow trivially from Theorem 37 if the root bag, after removing the few high-degree vertices, was triconnected.

Otherwise, the informal idea is to cut the root bag into two parts, putting one part (the one pointed to by the tangle) as the new root bag and the other part as a new leaf bag, repeating this until the root bag is triconnected. However, as before, we will not be able to guarantee that the subgraph induced by the root bag is by itself triconnected. Instead, we consider a graph obtained by contracting each leaf bag into a single vertex. Furthermore, instead of repeatedly cutting this graph until we make it triconnected, we will cut it all at once by considering its Tutte decomposition (which subsumes all separations of order ≤ 2), as detailed later.

We now proceed with the formal proof. Without loss of generality assume that $\theta \geq \alpha(|V(H)| + 2)$, in the statement of Corollary 36 (otherwise increase θ). Let us first ensure that the star decomposition (T, \mathcal{X}) is connected:

Claim 39 *We can additionally assume that*

For every leaf node $t \in V(T)$, the set $\mathcal{X}(t) \setminus \mathcal{X}(t_{root})$ is connected, and (C1)

$\mathcal{X}(t) \setminus \mathcal{X}(t_{root})$ has edges to each vertex in the adhesion $\mathcal{X}(t) \cap \mathcal{X}(t_{root})$ (C2)

Proof If (C1) fails we can partition the leaf bag into many leaf bags. If (C2) fails we can remove the adhesion vertex in question from the leaf bag. □

We need to define a variant of the torso that is similarly connected as the root torso, but has no virtual edges (so that long paths in it imply long paths in G). Construct G_{root}^* from G as follows: contract $\mathcal{X}(t) \setminus \mathcal{X}(t_{root})$ to a single vertex x_t for each leaf $t \in T$ and merge any such vertices $x_t, x_{t'}$ that have equal neighborhoods (corresponding to the same adhesion) into one. Let X be the set of all those new vertices.

Claim 40 *If G has no k -path, then neither does G_{root}^* .*

Proof For each leaf node $t \in V(T)$, the set $\mathcal{X}(t) \setminus \mathcal{X}(t_{root})$ is connected by (C1) and can therefore be contracted to a single vertex x_t by repeated edge contractions. Since G_{root}^* is a subgraph of the graph obtained by these edge contractions, it follows that G_{root}^* is a minor of G . As minor operations do not increase the length of a longest path, the claim follows.

We show that G_{root}^* , just as the root torso, has almost bounded degree.

Claim 41 *G_{root}^* has $(|V(H)|, \max(\alpha, d + 2^d))$ -bounded degree.*

Proof Observe that G_{root}^* can be obtained from the root torso of (T, \mathcal{X}) by removing virtual edges and adding the vertices of X . Each vertex in X has degree equal to the size of some corresponding adhesion, which is $\leq \alpha$. Each vertex v in the torso gains as neighbors only vertices in X , which correspond to different (as vertex subsets) adhesions containing v ; since these adhesions induced cliques containing v in the torso (which had $(|V(H)|, d)$ -bounded degree), their number is bounded by 2^d (unless v was already one of the $\leq |V(H)|$ vertices of high degree).

The last two claims allow us to apply Theorem 37 to any triconnected subgraph of G_{root}^* (after removing high-degree vertices). We apply it to bags of a Tutte decomposition, obtaining:

Claim 42 *If G does not contain a path of length at least k , then G_{root}^* has a tree decomposition of adhesion size at most $|V(H)| + 2$ and width $k^{\mathcal{O}_H(1)}$.*

Proof Assume that G , and hence G_{root}^* , does not contain a path of length at least k . By the previous claim, in G_{root}^* one can delete $\leq |V(H)|$ vertices to obtain a subgraph of degree bounded by $\Delta := \max(\alpha, d + 2^d)$; name this subgraph G_{root}^\dagger . Consider the Tutte decomposition [32] of G_{root}^\dagger . In modern terms (cf. [21, Theorem 1]), it is a tree decomposition of G_{root}^\dagger of adhesion size at most two, whose torsos are triconnected

topological minors of G_{root}^\dagger . Consequently, each torso again has degree bounded by Δ and cannot contain a path of length at least k . By Theorem 37, if n denotes the size of such a bag, then $n^\varepsilon/4 + 2 < k$, where $\varepsilon = 1/\log_2(\max(425, \alpha, d + 2^d) - 1)$. Thus the size of each bag is $n < (4(k - 2))^{1/\varepsilon} = k^{\mathcal{O}(d + \log_2 \alpha)} = k^{\mathcal{O}_H(1)}$. Adding the $\leq |V(H)|$ deleted vertices back to every bag, we obtain a tree decomposition of G_{root}^* as claimed.

Denote the root torso of (T, \mathcal{X}) as G_{root} . We now transfer the decomposition of G_{root}^* to G_{root} .

Claim 43 *For every tree decomposition of G_{root}^* , one can find a tree decomposition of G_{root} with adhesion and maximum bag size at most α times larger than the adhesion and maximum bag size of the original tree decomposition.*

Proof Let (S, \mathcal{Y}) be a tree decomposition of G_{root}^* . Define $\mathcal{Y}'(s)$ as $\mathcal{Y}(s)$ with vertices $x_t \in X$ replaced by all of $N_{G_{\text{root}}^*}(x_t)$. We claim (S, \mathcal{Y}') is a tree decomposition of G_{root} . Every edge of G_{root} is either an edge of $G_{\text{root}}^* - X$, still contained in the same bag, or a virtual edge between two vertices of an adhesion, hence contained in $N_{G_{\text{root}}^*}(x_t)$ for some $x_t \in X$ and thus in a bag $\mathcal{Y}'(s)$ (for an s such that $\mathcal{Y}(s)$ contained x_t). To show that the subset $\mathcal{Y}'^{-1}(v) := \{s \in S : v \in \mathcal{Y}'(s)\}$ is connected, for any given vertex $v \in V(G_{\text{root}})$, observe that it is the union of $\mathcal{Y}^{-1}(v)$ with $\mathcal{Y}^{-1}(x_t)$ for all $x_t \in X$ adjacent to v in G_{root}^* ; since such sets $\mathcal{Y}^{-1}(v)$ and $\mathcal{Y}^{-1}(x_t)$ are connected in S and intersect (as x_t is adjacent to v), their union is connected. Each vertex in each bag has been replaced by at most $|N_{G_{\text{root}}^*}(x_t)|$ vertices, which is the size of an adhesion in (T, \mathcal{X}) and hence at most α . It follows that the maximum bag size increases by a factor at most α .

To bound the size of any adhesion, observe similarly that for $s_1, s_2 \in V(S)$, the adhesion is

$$\mathcal{Y}'(s_1) \cap \mathcal{Y}'(s_2) = ((\mathcal{Y}(s_1) \cap \mathcal{Y}(s_2)) \setminus X) \cup \bigcup_{x_{t_1} \in \mathcal{Y}(s_1), x_{t_2} \in \mathcal{Y}(s_2)} N(x_{t_1}) \cap N(x_{t_2}).$$

For any vertex $v \in N(x_{t_1}) \cap N(x_{t_2})$ (for some $x_{t_i} \in \mathcal{Y}(s_i)$), the set $\mathcal{Y}^{-1}(v)$ is connected and must intersect both $\mathcal{Y}^{-1}(x_{t_1})$ and $\mathcal{Y}^{-1}(x_{t_2})$. Hence either $\mathcal{Y}^{-1}(x_{t_1})$ contains both s_1 and s_2 , or $\mathcal{Y}^{-1}(x_{t_2})$ does, or $\mathcal{Y}^{-1}(v)$ does. This means x_{t_1}, x_{t_2} or v is contained in $\mathcal{Y}(s_1) \cap \mathcal{Y}(s_2)$. Therefore

$$\mathcal{Y}'(s_1) \cap \mathcal{Y}'(s_2) \subseteq ((\mathcal{Y}(s_1) \cap \mathcal{Y}(s_2)) \setminus X) \cup \bigcup_{x_t \in \mathcal{Y}(s_1) \cap \mathcal{Y}(s_2)} N(x_t).$$

This again implies each adhesion increases at most α times in size. □

By the last two claims, assuming G has no k -path, the root torso G_{root} has a tree decomposition (S, \mathcal{Y}) of adhesion size $\alpha' := \alpha(|V(H)| + 2)$ and width $k^{\mathcal{O}_H(1)}$.

Having a decomposition of the root torso G_{root} allows us to ‘compose it’ with the star decomposition (T, \mathcal{X}) of the whole graph G . The rough idea of this ‘composition’ is to replace the root bag of the star decomposition (T, \mathcal{X}) by a subset containing the

unique bag of (S, \mathcal{Y}) that is pointed to by the tangle. The composition results in a star decomposition with a small root bag, which will conclude the proof of Lemma 38.

To formally define the bag in (S, \mathcal{Y}) that is pointed to by the tangle \mathcal{T} , let us first see how \mathcal{T} orients separations of G_{root} . For a separation (A, B) of G_{root} , every adhesion $\mathcal{X}(t) \cap \mathcal{X}(t_{\text{root}})$ of (T, \mathcal{X}) induces a clique in G_{root} , hence it is fully contained in A or in B ; thus (A, B) can be extended to a separation of G by adding all of $\mathcal{X}(t)$ to A or B , correspondingly (the choice is arbitrary for $\mathcal{X}(t)$ such that $\mathcal{X}(t) \cap \mathcal{X}(t_{\text{root}}) \subseteq A \cap B$). We call such separations of G *extensions* of the separation (A, B) of G_{root} . For a separation (A, B) of order $\leq \theta$ of G_{root} , observe that either all extensions of (A, B) and no extensions of (B, A) belong to \mathcal{T} , or vice-versa: this follows from tangle axioms and the fact that (T, \mathcal{X}) respects \mathcal{T} (that is, each $\mathcal{X}(t)$ such that $\mathcal{X}(t) \cap \mathcal{X}(t_{\text{root}}) \subseteq A \cap B$ is small w.r.t. \mathcal{T} , so putting it one side of an extension (A', B') instead of the other cannot change whether A' is small, as otherwise the two small sides together with $\mathcal{X}(t)$ would cover all of G).

We now orient edges of the decomposition (S, \mathcal{Y}) according to \mathcal{T} , to find a (unique) bag we can focus on. For each edge $s_1s_2 \in E(S)$, if S_1, S_2 denote the two connected components of $S - s_1s_2$ containing s_1, s_2 , respectively, then $(\mathcal{Y}(S_1), \mathcal{Y}(S_2))$ is a separation with $|\mathcal{Y}(S_1) \cap \mathcal{Y}(S_2)| = |\mathcal{Y}(s_1) \cap \mathcal{Y}(s_2)| \leq \alpha' \leq \theta$, hence either all of its extensions or all extensions of $(\mathcal{Y}(S_2), \mathcal{Y}(S_1))$ belong to the tangle \mathcal{T} . We direct the edge s_1s_2 to either s_2 or s_1 accordingly (towards the side with large extensions). After repeating this for every edge of S , we find a node s_0 with no outgoing edges. This means that if we root S at s_0 and S_1, S_2, \dots, S_ℓ are the subtrees of $S - s_0$, then all extensions of $(\mathcal{Y}(S_i), \mathcal{Y}(S - S_i))$ belong to \mathcal{T} , for $i \in [\ell]$.

We can now forget about some details of the tree decomposition and define the following star decomposition (S', \mathcal{Y}') of G_{root} : its nodes are $s_{\text{root}} := s_0$ and its direct children s_1, \dots, s_ℓ in S , while its bags are $\mathcal{Y}'(s_{\text{root}}) := \mathcal{Y}(s_0)$ and $\mathcal{Y}'(s_i) := \mathcal{Y}(S_i)$ for $i \in [\ell]$. By construction (S', \mathcal{Y}') is a star decomposition of G_{root} of adhesion size α' , with a root bag of size $k^{\mathcal{O}_H(1)}$, which satisfies the following:

$$\text{all extensions of } (\mathcal{Y}(s_i), \mathcal{Y}(S - s_i)) \text{ are in } \mathcal{T}, \text{ for } i \in [\ell]. \tag{2}$$

The following claim thus concludes the proof of the lemma, by extending this star decomposition of G_{root} to a star decomposition of G respecting \mathcal{T} , with a root torso of almost bounded degree and size $k^{\mathcal{O}_H(1)}$.

Claim 44 *If G_{root} has a star decomposition (S, \mathcal{Y}) of adhesion size $\alpha' \leq \theta$ satisfying (2), then G has a star decomposition of adhesion size $\max(\alpha, \alpha')$ respecting \mathcal{T} where the root torso has $(|V(H)|, d + \alpha' \cdot d)$ -bounded degree and its vertex set is equal to the root bag of (S, \mathcal{Y}) .*

Proof Let (S, \mathcal{Y}) be a star decomposition of G_{root} satisfying (2). Without loss of generality assume that for every leaf $s \in V(S)$, every vertex of the adhesion $\mathcal{Y}(s) \cap \mathcal{Y}(s_{\text{root}})$ has a neighbor in $\mathcal{Y}(s) \setminus \mathcal{Y}(s_{\text{root}})$ (otherwise we can remove it from $\mathcal{Y}(s)$). Every adhesion of the (larger) star decomposition (T, \mathcal{X}) of G induces a clique in G_{root} by definition, hence we can choose for each $t \in V(T)$ a node $s(t) \in V(S)$ such that $\mathcal{X}(t) \cap \mathcal{X}(t_{\text{root}}) \subseteq \mathcal{Y}(s(t))$. We construct a new star decomposition (T', \mathcal{X}') of G from (T, \mathcal{X}) as follows: the root bag becomes $\mathcal{Y}(s_{\text{root}})$, leaves t with $s(t) = s_{\text{root}}$

and their bags remain unchanged, the remaining leaves are deleted, and each leaf s of (S, \mathcal{Y}) is added as a new leaf of T' , with $\mathcal{X}'(s) := \mathcal{Y}(s) \cup \bigcup_{t: s(t)=s} \mathcal{X}(t)$.

It is straightforward to check that (T', \mathcal{X}') is a star decomposition of G of adhesion size $\max(\alpha, \alpha')$, as follows. Every edge is covered by a bag in \mathcal{X}' , because it was covered by a bag in \mathcal{X} , which is either a subset of a bag in \mathcal{X}' , or in $\mathcal{X}(t_{\text{root}})$, and hence in a bag of \mathcal{Y} , which are again subsets of bags in \mathcal{X}' . Every vertex occurs in bags \mathcal{X}' of a connected subtree of T' , because if a vertex occurs in two different leaf bags of \mathcal{X}' , then either it occurs in two different leaf bags of \mathcal{Y} and hence in $\mathcal{Y}(s_{\text{root}}) = \mathcal{X}'(t_{\text{root}})$, or in two leaf bags $\mathcal{X}(t)$ and $\mathcal{X}(t')$ with $s(t) \neq s(t')$ and hence in $\mathcal{X}(t_{\text{root}})$, which means it occurs in the adhesions of $\mathcal{X}(t)$ and $\mathcal{X}(t')$; these adhesions are contained in $\mathcal{Y}(s(t))$ and $\mathcal{Y}(s(t'))$ respectively, so the vertex must be contained in $\mathcal{Y}(s_{\text{root}}) = \mathcal{X}'(t_{\text{root}})$ as well. Finally to check the sizes of adhesions, observe that each adhesion of (T, \mathcal{X}') is either (for leaf nodes t coming from T) a subset of an adhesion in \mathcal{X} , namely $\mathcal{X}'(t) \cap \mathcal{X}'(t_{\text{root}}) \subseteq \mathcal{X}(t) \cap \mathcal{X}(t_{\text{root}})$, or (for leaf nodes s coming from S) equal to an adhesion of (S, \mathcal{Y}) :

$$\mathcal{X}'(s) \cap \mathcal{X}'(t_{\text{root}}) = \left(\mathcal{Y}(s) \cup \bigcup_{t: s(t)=s} \mathcal{X}(t) \right) \cap \mathcal{Y}(s_{\text{root}}) = \mathcal{Y}(s) \cap \mathcal{Y}(s_{\text{root}}),$$

since every vertex in $\mathcal{X}(t) \cap \mathcal{Y}(s_{\text{root}}) \subseteq \mathcal{X}(t) \cap \mathcal{X}(t_{\text{root}})$ is contained in $\mathcal{Y}(s(t))$.

To show that the new decomposition respects \mathcal{T} , suppose to the contrary that some leaf bag contains the large side B of a separation $(A, B) \in \mathcal{T}$. Then either this bag is $\mathcal{X}'(t) = \mathcal{X}(t)$ for $t \in V(T)$, contradicting that (T, \mathcal{X}) respects \mathcal{T} , or $\mathcal{X}'(s) = \mathcal{Y}(s) \cup \bigcup_{t: s(t)=s} \mathcal{X}(t)$ for some $s \in V(S)$. Then since $(\mathcal{X}'(s), \bigcup_{s' \neq s \in V(S)} \mathcal{X}'(s'))$ is a separation induced by the adhesion at $\mathcal{X}'(s)$, of order $\leq \alpha'$ (as shown in the previous paragraph), $\mathcal{X}'(s)$ (which contains B) must be the large side of this separation in \mathcal{T} . But this is an extension of the separation $(\mathcal{Y}(s), \bigcup_{s' \neq s \in V(S)} \mathcal{Y}(s'))$, contradicting (2).

The part that is not entirely straightforward is bounding the degrees in the root torso of the new star decomposition (T', \mathcal{X}') . Observe that this new torso can be obtained from the root torso G_{root} of (T, \mathcal{X}) by taking the subgraph induced by $\mathcal{X}'(t_{\text{root}}) = \mathcal{Y}(s_{\text{root}})$, removing certain virtual edges (from adhesions corresponding to deleted leaves) and adding virtual edges between every two vertices of each new adhesion $\mathcal{X}'(s) \cap \mathcal{X}'(t_{\text{root}})$, $s \in V(S)$. We need to bound the number of such edges added to a vertex $v \in \mathcal{Y}(s_{\text{root}})$, assuming it was not already one of the $|V(H)|$ high-degree vertices of the root torso of (T, \mathcal{X}) . Thus v had at most d neighbors in G_{root} . Since for each new adhesion $\mathcal{X}'(s) \cap \mathcal{X}'(t_{\text{root}}) = \mathcal{Y}(s) \cap \mathcal{Y}(s_{\text{root}})$ (as shown above) which contains v we assumed that v has a neighbor in $\mathcal{Y}(s) \setminus \mathcal{Y}(s_{\text{root}})$ (and these are disjoint sets, for distinct s), it follows that v received new edges from at most d distinct new adhesions, at most α' edges from each. Therefore, the degree of all but $|V(H)|$ vertices in the root torso of (T', \mathcal{X}') is bounded by $d + \alpha' \cdot d$, that is, Corollary 36 holds with this constant in place of d . □

This concludes the proof of Lemma 38.

6 Adding a Modulator

In this section we prove Theorem 2. In fact, in line with the setting of Sect. 3, we show the following generic version. Theorem 2 follows directly from it by applying Theorem 33 (on the (w, h) -decomposability of H -topological-minor-free graphs for $w(k) = k^{\mathcal{O}_H(1)}$ and $h = \mathcal{O}_H(1)$).

Theorem 45 *One can solve in polynomial time a given k -PATH instance (G, k) , given access to a set $M \subseteq V(G)$ such that $G - M$ admits a tree decomposition of width less than w and adhesion size $h = \mathcal{O}(1)$, and an oracle that solves the AUXILIARY LINKAGE problem for instances $(G', k', S, (R_i)_{i=1}^r)$ with G' being a subgraph of G , $\max(r, k') \leq k$, $|S| \leq |M| + \mathcal{O}(1)$, and $|V(G')|$ being bounded polynomially in k, w , and $|M|$.*

Contrary to Sect. 3, in this section we are not precise with the polynomial dependency on the parameters k, w , and $|M|$, as well as the dependency on the adhesion size $h = \mathcal{O}(1)$ of the decomposition. This is due to the fact that the latter dependency on h is substantial, most notably in the exponent of the polynomial bound on the size of the oracle calls. Therefore, the result of this section is a purely theoretical result classifying the aforementioned parameterization as admitting a polynomial Turing kernel.

Decomposition We apply the algorithm of Theorem 11 to the graph $G - M$ and separation size h . Since $h = \mathcal{O}(1)$, the algorithm runs in polynomial time. By Lemma 12, the output decomposition (T_0, \mathcal{X}_0) has width $w^{\mathcal{O}(1)}$ and adhesion size $\mathcal{O}(1)$.

Our algorithm will need a bound on the number of children of each node in the decomposition tree. This is easy to guarantee by modifying the decomposition (T_0, \mathcal{X}_0) as follows. For every node $t \in V(T)$, we group the children t' of t according to the adhesions $\mathcal{X}(t) \cap \mathcal{X}(t')$. For every group $U \subseteq N_T(t)$, we add a new child t_U of t with $\mathcal{X}(t_U) = \mathcal{X}(t) \cap \mathcal{X}(t')$, $t' \in U$, and reattach the nodes of U as children of t_U . Now t has $w^{\mathcal{O}(1)}$ children (using the trivial bound on the adhesion degree), but t_U might have many children. However, the bag $\mathcal{X}(t_U)$ has constant size, we can hence duplicate t_U into $\mathcal{O}(|U|)$ copies, arranged in a binary tree, with the nodes of U being different leaves of the binary tree. Let (T, \mathcal{X}) be the final decomposition: it has unchanged width $w^{\mathcal{O}(1)}$ and adhesion size $\mathcal{O}(1)$, and additionally every node has $w^{\mathcal{O}(1)}$ children.

Marking We continue the proof of Theorem 45 with a somewhat standard marking process. The idea will be to mark enough paths that realize a traverse in $G - M$ between each possible pair $u, v \in M$, so that we can assume every traverse of the sought k -path intersects marked vertices. If there are many unmarked vertices, we will be able to find a moderately large unmarked part A in the decomposition with a constant-size boundary Z (separating A from marked vertices), which will be the guarded set and the guard, respectively. This allows us to perform the Reduction Rule on A .

For an integer $0 \leq k' \leq k - 2$ and two vertices $u, v \in M$, define a (u, v, k') -path in G as a (simple) path with $k' + 2$ vertices: two endpoints u and v and exactly k' internal vertices, all in $G - M$. Note that a $(u, v, 0)$ -path is a path consisting of an edge uv .

Similarly, for $0 \leq k' \leq k - 1$ and a vertex $u \in M$, a (u, k') -path in G is a path with $k' + 1$ vertices: one endpoint being u and all other vertices in $G - M$.

For every $0 \leq k' \leq k - 2$ and every $u, v \in M$, we shall compute an inclusion-wise maximal family $\mathcal{P}_{u,v,k'}^0$ of internally vertex-disjoint (u, v, k') -paths. Similarly, for every $0 \leq k' \leq k - 1$ and $u \in M$ we compute an inclusion-wise maximal family $\mathcal{P}_{u,k'}^0$ of internally vertex-disjoint (u, k') -paths. The computation can be done via the previously established kernel for decomposable graph classes.

Lemma 46 *The families $\mathcal{P}_{u,v,k'}^0$ and $\mathcal{P}_{u,k'}^0$ can be computed in polynomial time using the assumed AUXILIARY LINKAGE oracle.*

Proof We focus on $\mathcal{P}_{u,v,k'}^0$; the proof for the second family is analogous. To find an inclusion-wise maximal family, it suffices to find paths one-by-one and greedily add them to the family as long as possible. That is, it suffices to find a path of the desired length $k' \leq k$ between two given vertices u and v in a given induced subgraph of $G - (M \setminus \{u, v\})$ (obtained by removing the internal vertices of the paths previously found), or conclude that none exists (meaning the packing we found is maximal). We show that a slight modification of the previously established kernel in Sect. 3 for decomposable graph classes (Corollary 14) works when endpoints u and v are given.

More precisely, first observe that the Reduction Rule is still correct when we are looking for a k' -path between fixed endpoints u and v (as opposed to any k' -path) as long as the endpoints do not lie in the set A . Thus, we can (just as in Corollary 14) iteratively use the decomposition (T, \mathcal{X}) of $G - M$, reintroduce u and v to every bag, and find a separation (A', B') by Lemma 9 such that the Reduction Rule applies (with $A = A' \setminus B'$ not containing u nor v), and iterate this until we find the k' -path between u and v or conclude there is none, as in Lemma 7. \square

We define $\mathcal{P}_{u,v,k'}$ to be a subfamily of $k + 1$ arbitrarily chosen paths from $\mathcal{P}_{u,v,k'}^0$, or let $\mathcal{P}_{u,v,k'} = \mathcal{P}_{u,v,k'}^0$ if $|\mathcal{P}_{u,v,k'}^0| < k + 1$. We define $\mathcal{P}_{u,k'}$ similarly.

Let $|M| = s$. Let X_1 be the set of all vertices of $G - M$ that lie on some path in any of the sets $\mathcal{P}_{u,v,k'}$ or $\mathcal{P}_{u,k'}$ ($u, v \in M, k' \leq k$). Note that, as we picked at most $k + 1$ paths to the families $\mathcal{P}_{u,v,k'}$ and $\mathcal{P}_{u,k'}$, each consisting of at most $k - 1$ vertices from $G - M$, for each of at most k choices of the length $k' \leq k$, and $\binom{s}{2} + s \leq 2s^2$ choices for the endpoint(s), we have

$$|X_1| < (k + 1) \cdot (k - 1) \cdot k \cdot 2s^2.$$

For every $x \in X_1$, pick one node $t(x) \in V(T)$ such that $x \in \mathcal{X}(t(x))$. Let $B_1 = \{t(x) \mid x \in X_1\}$. Let $B_2 \subseteq V(T)$ consist of all of B_1 , the root of T , and the lowest common ancestor of every pair t_1, t_2 in B_1 . Let $X_2 = \mathcal{X}(B_2)$. It is folklore that $|B_2| \leq 2|B_1|$, hence:

$$\begin{aligned} |B_2| &\leq 2(k + 1) \cdot (k - 1) \cdot k \cdot 2s^2, \\ |X_2| &= w^{\mathcal{O}(1)} \cdot k^3 \cdot s^2. \end{aligned}$$

Let \mathcal{C} be the set of connected components of $T - B_2$. As B_2 is closed under taking lowest common ancestors of nodes, we have that $|N_T(C)| \leq 2$ for any $C \in \mathcal{C}$ (indeed,

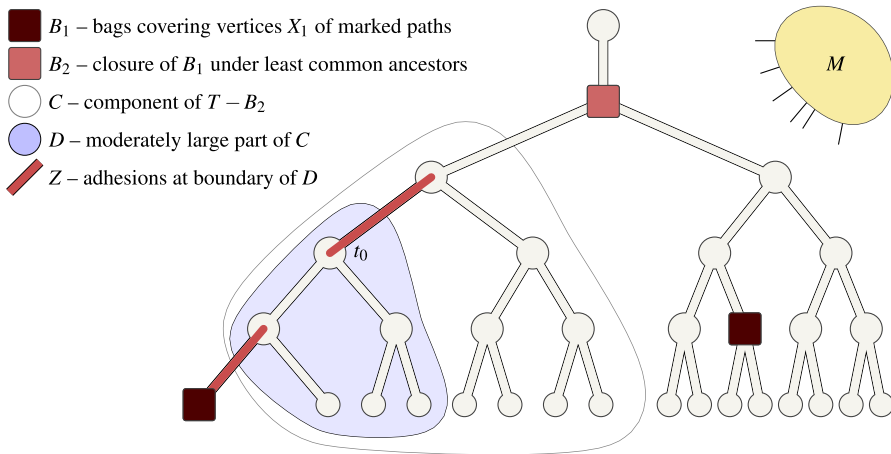


Fig. 4 Schematic illustration of the steps used to define the set A that is guarded by Z . The figure shows a tree decomposition of $G - M$, whose vertices can have arbitrary connections to the modulator M

the set $N_T(C) \subseteq B_2$ consists of the parent of the root of C , if it has a parent, and the children of leaves of C , of which there can be at most one, as otherwise their common ancestor would be in C and B_2 , a contradiction).

Reducing a large component Our main goal is to show that if the bags of any component C contain in total more than $m = \text{poly}(k, w, s)$ vertices of G , that is, $|\mathcal{X}(C)| > m$, then one can use the Reduction Rule from Sect. 3 to reduce the graph G by at least one vertex. Fix such a component C , and refer to Fig. 4 for an illustration of the following steps.

Let $t_0 \in V(C)$ be the lowest node in C such that, if D is the set of nodes of C in the subtree of T rooted in t_0 , then $|\mathcal{X}(D)| > m$. Since in (T, \mathcal{X}) nodes have $w^{\mathcal{O}(1)}$ children and bags have size $w^{\mathcal{O}(1)}$, the choice of t_0 ensures that $|\mathcal{X}(D)| \leq w^{\mathcal{O}(1)} \cdot m$.

Let $Z = \mathcal{X}(D) \cap \mathcal{X}(N_T(D))$ and $A = \mathcal{X}(D) \setminus \mathcal{X}(N_T(D))$. Note that D has the same neighborhood in T as C , except possibly for a different parent of its root. Hence $|N_T(D)| \leq 2$ and since adhesions are of constant size, $|Z| = \mathcal{O}(1)$. Furthermore, since $D \subseteq C$ is disjoint from B_2 , we have (by properties of tree decompositions):

$$\begin{aligned} \mathcal{X}(D) \cap X_2 &= \mathcal{X}(D) \cap \mathcal{X}(B_2) \subseteq \mathcal{X}(D) \cap \mathcal{X}(T - D) \\ &= \mathcal{X}(D) \cap \mathcal{X}(N_T(D)) = Z. \end{aligned}$$

Therefore $A \cap X_2 = \emptyset$. Again by properties of tree decompositions, Z separates A from the rest: $N_{G-M}(A) \subseteq Z$, hence $N_G(A) \subseteq Z \cup M$.

The marking process ensures the following crucial property.

Lemma 47 *The set A is guarded by Z .*

Proof Let P be a k -path in G that contains a maximum possible number of vertices from X_1 ; we claim that P is guarded by Z in A . Assume the contrary, let Q be a traverse of P through A for which neither endpoint belongs to Z . Since P is not

completely contained in A (as otherwise it would be guarded, by Definition 4) and $N_G(A) \subseteq Z \cup M$, we have that Q is either a (u, v, k') -path for some $u, v \in M$ and $1 \leq k' \leq k - 2$ or a (u, k') -path for some $u \in M$ and $1 \leq k' \leq k - 1$. Assume the first case; the proof for the second case is analogous.

Since Q is contained in $A \cup M$ and A is disjoint from $X_2 \supseteq X_1$ (the marked vertices), it is internally vertex-disjoint from all paths in $\mathcal{P}_{u,v,k'}$. If $\mathcal{P}_{u,v,k'}$ were equal to $\mathcal{P}_{u,v,k'}^0$, we could have added Q to the latter, contradicting its choice as inclusion-wise maximal. Consequently, $|\mathcal{P}_{u,v,k'}| = k + 1$. Thus, at least one (u, v, k') -path $Q' \in \mathcal{P}_{u,v,k'}$ has no internal vertices on the k -path P . Consequently, by replacing Q with Q' on P , we obtain a simple k -path P' with strictly more vertices of X_1 than P . This is a contradiction. \square

Thus, if $m = \text{poly}(k, w, s)$ is large enough, we can apply the Reduction Rule for the set A and the guard Z ; recall that $|Z| = \mathcal{O}(1)$ while $m - \mathcal{O}(1) < |A| \leq \text{poly}(w) \cdot m$. The Reduction Rule deletes at least one vertex after invoking a number of calls to the AUXILIARY LINKAGE oracle on the subgraph $G[N[A]]$; note that $N[A] \subseteq A \cup M \cup Z$.

Wrap up Recall that we have shown that one can apply the Reduction Rule if there exists a component $C \in \mathcal{C}$ with $|\mathcal{X}(C)| > m$ for some m bounded polynomially in k, w , and s . However, since $|B_2| \leq 2(k+1) \cdot (k-1) \cdot k \cdot 2s^2$ and a node of T has $\text{poly}(w)$ children, there are at most $\text{poly}(w)k^3s^2$ components. Consequently, if the reduction is not applicable, we have $|V(G)|$ bounded polynomially in k, w , and s . Such an instance can be solved with a single call to AUXILIARY LINKAGE with $k' = k, S = \emptyset, r = 1$, and $R_1 = \emptyset$. This finishes the proof of Theorem 45 and, consequently, of Theorem 2.

7 Conclusions

We significantly extended the graph classes on which k -PATH has a polynomial Turing kernel. In addition, we showed that even an instance that does not belong to such a class, but has a small vertex modulator whose deletion makes it so, can be solved efficiently using small queries to an oracle. A subdivision-based argument (cf. [15]) shows that we cannot generalize much beyond H -topological-minor-free graphs without settling the problem in general. In particular, the existence of a polynomial Turing kernel for graphs of bounded expansion implies its existence in general graphs.

While our narrative focused on k -PATH, after small modifications our techniques can also be applied to prove analogues of Theorems 1 and 2 for the k -CYCLE problem of detecting a simple cycle of length *at least* k . The main difficulty in adapting our arguments to k -CYCLE is the fact that, a priori, the only cycles of length at least k may be arbitrarily much larger than k . However, this issue can easily be resolved in the following way. Since a cycle is contained within a single biconnected component, a Turing kernelization can decompose its input into biconnected components and solve the problem independently in each of them. We then start by testing for the existence of a *path* with k^2 vertices using the algorithms developed in the paper. If there is a path of length k^2 in a biconnected component, then by a classic theorem of Dirac [13] there is a cycle of length at least k , and we are done. If no such path exists, then the longest cycle in G has length less than $2k$, and we can continue under the guarantee

that the cycle we are looking for has length at least k and less than $2k$. In this setting, our arguments can be easily adapted. In particular, the absence of a path of length k^2 implies the existence of suitable tree decompositions from which reducible separations can be extracted.

A significant portion of the technical work in this paper was devoted to modifying the graph minors decomposition to obtain the win/win that either answers the problem or finds a reducible separation. In this way, the algorithmic question has driven a challenging graph-theoretic project. It would be interesting to find more problems amenable to such an approach. We conclude with some concrete open problems. Does k -PATH have a polynomial Turing kernel on chordal graphs? How about INDUCED k -PATH or DIRECTED k -PATH, on planar graphs? The current approach fails for these problems, since the presence of a large triconnected component does not necessarily imply a positive answer to the problem. For example, there are triconnected planar graphs on n vertices whose longest induced path has only $\mathcal{O}(\log n)$ vertices [17].

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

A Omitted proofs

A.1 Proof of Lemma 17

Consider a node $t \in V(T)$, let T_t be the subtree of T rooted in t , and let $G_t = G[\mathcal{X}(T_t)]$. Note that (T_t, \mathcal{X}) is a tree decomposition of G_t with t being a root; with regards to this decomposition, we consider the torso $H_t := \text{TORSO}(G_t, \mathcal{X}(t))$. Note that the connectivity of (T, \mathcal{X}) implies that H_t can be obtained from $G_t[\mathcal{X}(t)]$ by turning the neighborhood $N_{G_t}(C)$ into a clique for every connected component C of $G_t - \mathcal{X}(t)$. Hence every adhesion to a child of t corresponds to a clique in H_t , and we may bound the adhesion degree by bounding the number of distinct cliques in H_t .

To show the bound on the number of cliques, we show a longer, more elaborate process that turns G_t into H_t . Let \mathcal{C} be the family of connected components of $G_t - \mathcal{X}(t)$. For every $C \in \mathcal{C}$ and every vertex $v \in N(C) \subseteq \mathcal{X}(t)$, pick one neighbor $x_{v,C} \in N(v) \cap C$. Let S_C be a minimal connected subgraph of C that contains every vertex $x_{v,C}$ for $v \in N(C)$. Note that by minimality, S_C is a tree with at most $|N(C)| \leq h$ leaves and thus less than h vertices of degree at least 3. Delete from G_t all vertices and edges of $G[C]$ that do not lie in S_C , and suppress every node of degree 2 in S_C that is not of the form $x_{v,C}$ for $v \in N(C)$. By performing this operation for every component $C \in \mathcal{C}$, we obtain a topological minor G_t^1 of G_t with the following property: the graph $G_t[\mathcal{X}(t)]$ remains untouched in G_t^1 , while every $C \in \mathcal{C}$ is turned into its topological minor C^1 with $N(C) = N(C^1)$ and $|C^1| < 2|N(C)| \leq 2h$. Since G does not contain H as a topological minor, neither does G_t^1 .

We now switch to the theory of graphs of bounded expansion; for an introduction and more notation we refer to the textbook [25]. A graph class \mathcal{G} is of *bounded expansion*

if there exists a function $\nabla : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ such that for every radius r , every graph $H \in \mathcal{G}$, and every r -shallow minor H' of H , we have $|E(H')|/|V(H')| \leq \nabla(r)$.

Let G_t^2 be a graph obtained from G_t^1 by contracting every component C^1 for $C \in \mathcal{C}$ into a single vertex x_C . Then, as $|C^1| < 2h$, G_t^2 is a $2h$ -shallow minor of G_t^1 . Let G_t^3 be a graph obtained from G_t^2 by replacing every vertex x_C for $C \in \mathcal{C}$ with a clique K_C of $|N(C)|$ copies of x_C . Finally, note that H_t can be obtained from G_t^3 by contracting every vertex of K_C onto a distinct vertex of $N(C)$, that is, H_t is a 1-shallow minor of G_t^3 .

Since taking an $\mathcal{O}(1)$ -shallow-minor or replacing every vertex with a clique of constant size in a graph from a class of bounded expansion gives a graph also from a class of bounded expansion (but possibly worse function ∇) [25], we have that there exists a graph class \mathcal{G} of bounded expansion (with the function ∇ depending on H and h) such that $H_t \in \mathcal{G}$. Since graphs of bounded expansion have bounded degeneracy and therefore have linearly many cliques [25, Lemma 3.1], we have that H_t contains at most $f'(h, H) \cdot |\mathcal{X}(t)|$ cliques for some constant $f'(h, H)$ depending on h and H . Thus, in Lemma 17 we can take $f(h, H) = 1 + f'(h, H)$ to accommodate for the additional adhesion corresponding to the parent of t .

References

1. Ambalath, A.M., Balasundaram, R., Rao, R., Koppula, V., Misra, N., Philip, G., Ramanujan, M.S.: On the kernelization complexity of colorful motifs. In: Proceedings of the 5th IPEC, pp. 14–25 (2010). https://doi.org/10.1007/978-3-642-17493-3_4
2. Archdeacon, D.: Topological graph theory: a survey. *Congr. Numerantium* **115**(5–54), 18 (1996)
3. Barbero, F., Paul, C., Pilipczuk, M.: Exploring the complexity of layout parameters in tournaments and semicomplete digraphs. *ACM Trans. Algorithms* **14**(3), 38:1–38:31 (2018). <https://doi.org/10.1145/3196276>
4. Binkele-Raible, D., Fernau, H., Fomin, F.V., Lokshantov, D., Saurabh, S., Villanger, Y.: Kernel(s) for problems with no kernel: on out-trees with many leaves. *ACM Trans. Algorithms* **8**(4), 38 (2012). <https://doi.org/10.1145/2344422.2344428>
5. Bodlaender, H.L., Demaine, E.D., Fellows, M.R., Guo, J., Hermelin, D., Lokshantov, D., Müller, M., Raman, V., Rooij, J.V., Rosamond, F.A.: Open problems in parameterized and exact computation—IWPEC 2008. Technical Report UU-CS-2008-017, Utrecht University (2008)
6. Bodlaender, H.L., Jansen, B.M.P., Kratsch, S.: Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.* **28**(1), 277–305 (2014). <https://doi.org/10.1137/120880240>
7. Chen, G., Gao, Z., Yu, X., Zang, W.: Approximating longest cycles in graphs with bounded degrees. *SIAM J. Comput.* **36**(3), 635–656 (2006). <https://doi.org/10.1137/050633263>
8. Chen, G., Yu, X.: Long cycles in 3-connected graphs. *J. Comb. Theory Ser. B* **86**(1), 80–99 (2002). <https://doi.org/10.1006/jctb.2002.2113>
9. Chen, G., Yu, X., Zang, W.: The circumference of a graph with no $K_{3,t}$ -minor, II. *J. Comb. Theory Ser. B* **102**(6), 1211–1240 (2012). <https://doi.org/10.1016/j.jctb.2012.07.003>
10. Cygan, M., Lokshantov, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Minimum bisection is fixed parameter tractable. In: Proceedings of STOC 2014, pp. 323–332. ACM (2014). <https://doi.org/10.1145/2591796.2591852>
11. Diestel, R.: *Graph Theory*, 4th edn. Springer, Heidelberg (2010)
12. Diestel, R., Kawarabayashi, K., Müller, T., Wollan, P.: On the excluded minor structure theorem for graphs of large tree-width. *J. Comb. Theory Ser. B* **102**(6), 1189–1210 (2012). <https://doi.org/10.1016/j.jctb.2012.07.001>
13. Dirac, G.A.: Some theorems on abstract graphs. *Proc. Lond. Math. Soc.* **s3–2**(1), 69–81 (1952). <https://doi.org/10.1112/plms/s3-2.1.69>

14. Fröhlich, J., Müller, T.: Linear connectivity forces large complete bipartite minors: an alternative approach. *J. Comb. Theory Ser. B* **101**(6), 502–508 (2011). <https://doi.org/10.1016/j.jctb.2011.02.002>
15. Gajarský, J., Hliněný, P., Obdržálek, J., Ordyniak, S., Reidl, F., Rossmanith, P., Villaamil, F.S., Sikdar, S.: Kernelization using structural parameters on sparse graph classes. *J. Comput. Syst. Sci.* **84**, 219–242 (2017). <https://doi.org/10.1016/j.jcss.2016.09.002>
16. Garnero, V., Weller, M.: Parameterized certificate dispersal and its variants. *Theor. Comput. Sci.* **622**, 66–78 (2016). <https://doi.org/10.1016/j.tcs.2016.02.001>
17. Giacomo, E.D., Liotta, G., Mchedlidze, T.: Lower and upper bounds for long induced paths in 3-connected planar graphs. *Theor. Comput. Sci.* **636**, 47–55 (2016). <https://doi.org/10.1016/j.tcs.2016.04.034>
18. Grohe, M., Marx, D.: Structure theorem and isomorphism test for graphs with excluded topological subgraphs. *SIAM J. Comput.* **44**(1), 114–159 (2015). <https://doi.org/10.1137/120892234>
19. Hermelin, D., Kratsch, S., Soltys, K., Wahlström, M., Wu, X.: A completeness theory for polynomial (Turing) kernelization. *Algorithmica* **71**(3), 702–730 (2015). <https://doi.org/10.1007/s00453-014-9910-8>
20. Hüffner, F., Komusiewicz, C., Sorge, M.: Finding highly connected subgraphs. In: Proceedings of 41st SOFSEM, pp. 254–265 (2015). https://doi.org/10.1007/978-3-662-46078-8_21
21. Jansen, B.M.P.: Turing kernelization for finding long paths and cycles in restricted graph classes. *J. Comput. Syst. Sci.* **85**, 18–37 (2017). <https://doi.org/10.1016/j.jcss.2016.10.008>
22. Jansen, B.M.P., Marx, D.: Characterizing the easy-to-find subgraphs from the viewpoint of polynomial-time algorithms, kernels, and Turing kernels. In: Proceedings of 26th SODA, pp. 616–629 (2015). <https://doi.org/10.1137/1.9781611973730.42>
23. Kolay, S., Panolan, F.: Parameterized algorithms for deletion to (r, ℓ) -graphs. In: Proceedings of 35th FSTTCS, pp. 420–433 (2015). <https://doi.org/10.4230/LIPIcs.FSTTCS.2015.420>
24. Lokshantov, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Manuscript (2019)
25. Nešetřil, J., Ossona de Mendez, P.: Sparsity: Graphs, Structures, and Algorithms, Algorithms and Combinatorics, vol. 28. Springer, Berlin (2012). <https://doi.org/10.1007/978-3-642-27875-4>
26. Robertson, N., Seymour, P.D.: Graph minors. V. Excluding a planar graph. *J. Comb. Theory Ser. B* **41**(1), 92–114 (1986). [https://doi.org/10.1016/0095-8956\(86\)90030-4](https://doi.org/10.1016/0095-8956(86)90030-4)
27. Robertson, N., Seymour, P.D.: Graph minors. X. Obstructions to tree-decomposition. *J. Comb. Theory Ser. B* **52**(2), 153–190 (1991)
28. Robertson, N., Seymour, P.D.: Graph minors: XVII. Taming a vortex. *J. Comb. Theory Ser. B* **77**(1), 162–210 (1999). <https://doi.org/10.1006/jctb.1999.1919>
29. Schäfer, A., Komusiewicz, C., Moser, H., Niedermeier, R.: Parameterized computational complexity of finding small-diameter subgraphs. *Optim. Lett.* **6**(5), 883–891 (2012). <https://doi.org/10.1007/s11590-011-0311-5>
30. Shan, S.: Homeomorphically irreducible spanning trees, Halin graphs, and long cycles in 3-connected graphs with bounded maximum degrees. Ph.D. thesis, Georgia State University (2015). http://scholarworks.gsu.edu/math_diss/23/. Accessed 5 Nov 2015
31. Thomassé, S., Trotignon, N., Vuskovic, K.: A polynomial turing-kernel for weighted independent set in bull-free graphs. *Algorithmica* **77**(3), 619–641 (2017). <https://doi.org/10.1007/s00453-015-0083-x>
32. Tutte, W.T.: Connectivity in Graphs. Mathematical expositions. University of Toronto Press, Toronto (1966)