


Robust Contraction Decomposition for Minor-Free Graphs and Its Applications

Sayan Bandyapadhyay 

Portland State University, OR, USA

William Lochet 


LIRMM, Université de Montpellier, CNRS, France

Daniel Lokshtanov 

University of California, Santa Barbara, CA, USA

Dániel Marx 

CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Pranabendu Misra 

Chennai Mathematical Institute, India

Daniel Neuen 


Max Planck Institute for Informatics, SIC, Saarbrücken, Germany

Saket Saurabh 

The Institute of Mathematical Sciences, Chennai, India

Prafullkumar Tale 

Indian Institute of Science Education and Research, Pune, India

Jie Xue 

New York University Shanghai, China

Abstract

We prove a robust contraction decomposition theorem for H -minor-free graphs, which states that given an H -minor-free graph G and an integer p , one can partition in polynomial time the vertices of G into p sets Z_1, \dots, Z_p such that $\text{tw}(G/(Z_i \setminus Z')) = O(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$. Here, $\text{tw}(\cdot)$ denotes the treewidth of a graph and $G/(Z_i \setminus Z')$ denotes the graph obtained from G by contracting all edges with both endpoints in $Z_i \setminus Z'$.

Our result generalizes earlier results by Klein [SICOMP 2008] and Demaine et al. [STOC 2011] based on partitioning $E(G)$, and some recent theorems for planar graphs by Marx et al. [SODA 2022], for bounded-genus graphs (more generally, almost-embeddable graphs) by Bandyapadhyay et al. [SODA 2022], and for unit-disk graphs by Bandyapadhyay et al. [SoCG 2022].

The robust contraction decomposition theorem directly results in parameterized algorithms with running time $2^{\tilde{O}(\sqrt{k})} \cdot n^{O(1)}$ or $n^{O(\sqrt{k})}$ for every vertex/edge deletion problems on H -minor-free graphs that can be formulated as PERMUTATION CSP DELETION or 2-CONN PERMUTATION CSP DELETION. Consequently, we obtain the first subexponential-time parameterized algorithms for SUBSET FEEDBACK VERTEX SET, SUBSET ODD CYCLE TRANSVERSAL, SUBSET GROUP FEEDBACK VERTEX SET, 2-CONN COMPONENT ORDER CONNECTIVITY on H -minor-free graphs. For other problems which already have subexponential-time parameterized algorithms on H -minor-free graphs (e.g., ODD CYCLE TRANSVERSAL, VERTEX MULTIWAY CUT, VERTEX MULTICUT, etc.), our theorem gives much simpler algorithms of the same running time.

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

Keywords and phrases subexponential time algorithms, graph decomposition, planar graphs, minor-free graphs, graph contraction

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.17

Category Track A: Algorithms, Complexity and Games

Related Version Full Version: <https://arxiv.org/abs/2412.04145>

Funding Sayan Bandyapadhyay: The work has been supported by the NSF grant 2311397.

Pranabendu Misra: Supported by SERB StartUp Grant.



© Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Dániel Marx, Pranabendu Misra, Daniel Neuen, Saket Saurabh, Prafullkumar Tale, and Jie Xue; licensed under Creative Commons License CC-BY 4.0

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).

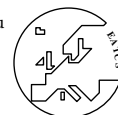
Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis

Article No. 17; pp. 17:1–17:19



Leibniz International Proceedings in Informatics

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



1 Introduction

Baker's layering technique is a fundamental tool for a certain type of algorithms on planar graphs. It was first used implicitly by Baker [1] and can be formalized as the following statement: given a planar graph G and an integer p , we can find in polynomial-time a partitioning Z_1, \dots, Z_p of the vertex set of G such that $G - Z_i$ has treewidth $O(p)$ for every $i \in [p]$. This decomposition can be used for problems where we can argue that there exist an $i \in [p]$ such that Z_i is irrelevant to the solution (or has negligible contribution to it) and hence Z_i can be deleted from G . Then we are left with an instance $G - Z_i$ having treewidth $O(p)$ and techniques for bounded-treewidth graphs can be used. This approach has been used in the design of polynomial-time approximation schemes (PTAS) [1, 6, 19] and parameterized algorithms [4, 10, 12, 15, 16, 29] for planar graphs. Moreover, this decomposition algorithm can be further generalized to H -minor free graphs [6].

Contraction Decomposition Theorems

There are many natural problems where a set Z_i cannot be removed even if it is irrelevant to the solution: most notably, problems involving connectivity typically have this property. To handle such problems, Klein [21] introduced a dual version of Baker's layering technique, which is based on contractions of edges. This *(Edge) Contraction Decomposition Theorem* was later generalized to H -minor free graphs by Demaine, Hajiaghayi, and Kawarabayashi [7].

► **Theorem 1** (Demaine, Hajiaghayi, and Kawarabayashi [7]). *Let H be a fixed graph. Given an H -minor-free graph G and an integer p , one can compute in polynomial time a partition of $E(G)$ into p sets Z_1, \dots, Z_p such that $\text{tw}(G/Z_i) = O(p)$ for all $i \in [p]$, where the constant hidden in $O(\cdot)$ depends on H .*

Contraction decomposition theorems of this form can be used as a building block in designing approximation schemes for, e.g., the TRAVELING SALESMAN PROBLEM (TSP) and SUBSET TSP [4, 7, 8, 20, 21], and parameterized algorithms for BISECTION, k -WAY CUT, ODD CYCLE TRANSVERSAL, SUBSET FEEDBACK VERTEX SET and many other problems [3, 18, 24].

Subexponential Parameterized Algorithms (Edge Problems)

To illustrate this, let us briefly discuss how Theorem 1 can be used to obtain a subexponential-time parameterized algorithm for EDGE MULTIWAY CUT in H -minor-free graphs. In EDGE MULTIWAY CUT, we are given a graph G , a set $T \subseteq V(G)$ of terminals, and an integer k , and the task is to find a set $S \subseteq E(G)$ of at most k edges such that every component of $G \setminus S$ contains at most one terminal.

- (1) A result of Wahlström [31] gives a randomized quasipolynomial kernel for the problem, which allows us to assume that $|V(G)| = k^{\text{polylog}(k)}$.
- (2) Given the decomposition Z_1, \dots, Z_p of Theorem 1 where $p := \lceil \sqrt{k} \rceil$, there is an $i \in [p]$ such that $|Z_i \cap S| \leq \sqrt{k}$ for the hypothetical solution S of size k .
- (3) Guess this $i \in [p]$ (there are \sqrt{k} possibilities) and the set $Z_i \cap S$ (there are $|V(G)|^{O(\sqrt{k})} = k^{\sqrt{k} \cdot \text{polylog}(k)}$ possibilities).
- (4) Contracting $Z_i \setminus S$ does not change the problem at hand, since the solution S is not affected.

(5) Since the graph G/Z_i has treewidth $O(\sqrt{k})$, we get that the graph $G/(Z_i \setminus S)$ has treewidth $O(\sqrt{k} + |Z_i \cap S|) = O(\sqrt{k})$ (contracting an edge can decrease treewidth by at most one).

Finally, we solve EDGE MULTIWAY CUT on $G/(Z_i \setminus S)$ in time $2^{O(\sqrt{k} \log k)} \cdot |V(G)|^{O(1)}$ using standard bounded-treewidth techniques.

Overall, we obtain a $2^{\sqrt{k} \cdot \text{polylog}(k)} \cdot n^{O(1)}$ time randomized algorithm for EDGE MULTIWAY CUT. The same approach also works for many other problems [2, 24].

Subexponential Parameterized Algorithms (Vertex Problems)

Let us try to apply a similar technique for the problem VERTEX MULTIWAY CUT, where instead of deleting a set of edges, we need to delete now a set of vertices. There are two main difficulties if we try to apply Theorem 1. First, it would be convenient to have a partition Z_1, \dots, Z_p of *vertices* such that for every $i \in [p]$ contracting Z_i leaves a graph of treewidth $O(p)$. Here contracting a vertex set Z_i amounts to contracting all edges that have both endpoints in the set Z_i . Second, in Step 5 above, we exploited that the decomposition of Theorem 1 is *inherently robust*: if Z'_i is obtained from Z_i by omitting a set of s edges, then the treewidth of G/Z'_i is at most s higher than the treewidth of G/Z_i .

The following example shows that an analogous statements for contracting vertex sets is not true. Let G be obtained from a $k \times k$ grid by adding two universal vertices x and y . Let A and B be the two bipartition classes of the $k \times k$ grid, and define $Z_1 := A \cup \{x\}$ and $Z_2 := B \cup \{y\}$. Then G/Z_i has treewidth 2 for both $i \in [2]$. On the other hand, $G/(Z_1 \setminus \{x\}) = G/A = G$ and $G/(Z_2 \setminus \{y\}) = G/B = G$ which means the jump in treewidth can be unbounded even after omitting just a single vertex. Note that G is K_7 -minor free, so we cannot hope that a vertex partition version of Theorem 1 is inherently robust even on H -minor-free graphs.

Robust Contraction Decomposition Theorem

The above naturally leads to the question of a *robust* vertex contraction decomposition theorem, where omitting vertices from some Z_i increases the treewidth only in a bounded way. Such decomposition theorems were recently introduced independently for planar graphs [24] and for bounded-genus graphs (and more generally, almost-embeddable graphs) [2] by subsets of the authors, who used them to obtain the first subexponential-time (parameterized) algorithms for a number of problems, such as EDGE BIPARTIZATION, ODD CYCLE TRANSVERSAL, EDGE/VERTEX MULTIWAY CUT, EDGE/VERTEX MULTICUT, and GROUP FEEDBACK EDGE/VERTEX SET on the above graph classes. More precisely, [2, 24] design parameterized algorithms with running time $f(k) \cdot n^{O(\sqrt{k})}$ or even $2^{\tilde{O}(\sqrt{k})} \cdot n^{O(1)}$, where k is the size of the solution (and $\tilde{O}(\cdot)$ hides $\log k$ factors).

The main result of this paper is a *vertex* contraction decomposition theorem on H -minor-free graphs that is *robust* in the sense described above.

► **Theorem 2.** *Let H be a fixed graph. Given an H -minor-free graph G and an integer p , one can compute in polynomial time a partition of $V(G)$ into p sets Z_1, \dots, Z_p such that $\text{tw}(G/(Z_i \setminus Z')) = O(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$, where the constant hidden in $O(\cdot)$ depends on H .*

Theorem 2 generalizes the robust vertex contraction theorem for planar graphs [24] and almost-embeddable graphs [2] to arbitrary H -minor-free graphs. It also generalizes the edge contraction decomposition (Theorem 1) for H -minor free graphs by Demaine et al. [7].

Combined with the results from [24], Theorem 2 directly yields parameterized algorithms of running time $2^{\tilde{O}(\sqrt{k})} \cdot n^{O(1)}$ or $n^{O(\sqrt{k})}$ for all vertex/edge deletion problems on H -minor-free graphs that can be formulated as PERMUTATION CSP DELETION or 2-CONN PERMUTATION CSP DELETION, two broad classes of problems defined in [24]. Roughly speaking, a permutation CSP instance is a binary CSP instance satisfying certain nice properties, and thus corresponds to a graph in which the vertices are variables and the edges are constraints. The PERMUTATION CSP EDGE/VERTEX DELETION problem basically asks whether one can delete at most k vertices (resp., edges) from (the graph of) a permutation CSP instance such that every connected component in the resulting graph has a satisfying assignment. The 2-CONN PERMUTATION CSP EDGE/VERTEX DELETION problem is defined similarly, but we only care about the satisfiability on every 2-connected component (instead of connected component). We refer to Section 3 for the precise definition of these problems. Combining the results from [24] and Theorem 2, we get the following result.

► **Theorem 3.** *Let H be a fixed graph. Then PERMUTATION CSP EDGE/VERTEX DELETION and 2-CONN PERMUTATION CSP EDGE/VERTEX DELETION on H -minor-free instances (Γ, w, δ, k) can be solved in $(|\Gamma| + \delta)^{O(\sqrt{k})}$ time.*

The above theorem directly gives us parameterized algorithms of running time exponential in $O(\sqrt{k})$ for many problems on H -minor-free graphs. Specifically, we obtain the first subexponential-time parameterized algorithms for the following fundamental problems on H -minor-free graphs.

- $n^{O(\sqrt{k})}$ time algorithms for SUBSET FEEDBACK VERTEX SET, SUBSET ODD CYCLE TRANSVERSAL, SUBSET GROUP FEEDBACK VERTEX SET, 2-CONN COMPONENT ORDER CONNECTIVITY, and the edge-deletion version of these problems.
- $2^{\tilde{O}(\sqrt{k})} \cdot n^{O(1)}$ time algorithms for SUBSET FEEDBACK VERTEX SET and SUBSET FEEDBACK EDGE SET.

Additionally, the main algorithmic results from [2] (such as subexponential-time parameterized algorithms for ODD CYCLE TRANSVERSAL, VERTEX MULTIWAY CUT, VERTEX MULTICUT, etc.), that required a complicated two-layered dynamic programming algorithm over the structural decomposition theorem of Robertson and Seymour [28] for H -minor free graphs, now admit much cleaner and more direct algorithms by exploiting Theorem 2 (these problems belong to the PERMUTATION CSP DELETION category in Theorem 3).

Other Relevant Literature

The study of subexponential-time parameterized algorithms on planar and H -minor free graphs has been one of the most active sub-areas of parameterized algorithms, which led to exciting results and powerful methods. Examples include Bidimensionality [5], applications of Baker’s layering technique [4, 10, 12, 15, 29], bounds on the treewidth of the solution [14, 22, 23, 25], and pattern coverage [15, 26]. The central theme of all these results is that planar graphs exhibit the “*square root phenomenon*”: parameterized problems whose fastest parameterized algorithm run in time $f(k) \cdot n^{O(k)}$ or $2^{O(k)} \cdot n^{O(1)}$ on general graphs admit $f(k) \cdot n^{O(\sqrt{k})}$ or even $2^{O(\sqrt{k})} \cdot n^{O(1)}$ time algorithms when input is restricted to planar or H -minor free graphs. However, these techniques do not apply for designing subexponential time parameterized algorithms for cut and cycle hitting problems such as MULTIWAY CUT, MULTICUT, or ODD CYCLE TRANSVERSAL. Robust vertex contraction decomposition theorems, first obtained in [2, 24], provide the necessary tools to design subexponential-time parameterized algorithms for some of the cut and cycle hitting problems on H -minor free graphs.

On the decomposition front, apart from edge contraction decomposition theorems, there are several other kind of decomposition theorems that have been used to design polynomial-time approximation schemes (PTASs) and FPT algorithms on planar graphs or more generally on H -minor free graphs. Most of these decomposition theorems prove strengthening of the classic Baker's layering technique [1]. This generally yields, in what is known as (Vertex) Edge Decomposition Theorems [1, 6, 9, 11, 13] (see [27] for a detailed introduction).

Finally, let us also point to some recent developments on the structural theory of H -minor-free graphs [17, 30]. Our current proof of Theorem 2 mostly relies on the original Robertson-Seymour decomposition for H -minor-free graphs [28], and is independent of [17, 30]. Reformulating some of our arguments in the language of [17, 30] may allow us to obtain a more streamlined proof in the future.

2 Overview of Proof of Theorem 2

In this section, we give an informal overview of our proof of Theorem 2. For convenience, we say Z_1, \dots, Z_p is a *robust contraction decomposition (RCD)* of a graph G if Z_1, \dots, Z_p is a partition of $V(G)$ satisfying $\text{tw}(G/(Z_i \setminus Z')) = O(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$. A p -RCD simply refers to an RCD of size p . Our goal is to compute a p -RCD of an H -minor-free graph G for a given integer $p \geq 1$.

Our starting point is the well-known Robertson-Seymour decomposition for H -minor-free graphs. The Robertson-Seymour decomposition of an H -minor-free graph G is a tree decomposition (T, β) of G in which the *torso* of each node $t \in V(T)$ is h -almost-embeddable and the *adhesion* of each node $t \in V(T)$ is of size at most h , for some constant h depending on H . Here, the adhesion of t , denoted by $\sigma(t)$, is the intersection of the bag $\beta(t)$ and the bag of the parent of t . The torso of t , denoted by $\text{tor}(t)$, is a supergraph of $G[\beta(t)]$ (the subgraph of G induced by $\beta(t)$), obtained by adding edges to $G[\beta(t)]$ to make the adhesion of each child of t a clique. Almost-embeddable graphs generalize bounded-genus graphs. Roughly speaking, an h -almost-embeddable graph has its main part embedded in a surface of genus at most h , and the remaining part consists of at most h well-structured subgraphs (called *vortices*) and a set of at most h “bad” vertices (called *apex vertices* or *apices*). In this overview, the reader can feel free to ignore the vortices/apices of an almost-embeddable graph and think about it as a graph embedded in a surface of bounded genus. The formal definitions of tree decompositions and almost-embeddable graphs can be found in the full version of the paper.

Intuitively, the Robertson-Seymour decomposition partitions an H -minor-free graph into almost-embeddable “pieces”. Since it is known that almost-embeddable graphs admit RCDs [2], a natural idea comes: Can we exploit Robertson-Seymour decomposition to somehow “lift” the RCDs of the almost-embeddable pieces to the H -minor-free graph? To explore this idea, let us consider an H -minor-free graph G and the Robertson-Seymour decomposition (T, β) of G . Since the torsos of (T, β) are h -almost-embeddable, we can compute a p -RCD $Z_1^{(t)}, \dots, Z_p^{(t)} \subseteq \beta(t)$ for each torso $\text{tor}(t)$ using the results of [2]. Ideally, if these RCDs are *consistent* in the sense that for every vertex $v \in V(G)$ there exists an index $i \in [p]$ such that $v \in Z_i^{(t)}$ for all nodes $t \in V(T)$ whose bag contains v , then we can combine them to obtain a partition Z_1, \dots, Z_p of $V(G)$ where $Z_i = \bigcup_{t \in V(T)} Z_i^{(t)}$ and hope it to be an RCD of G . (Of course, as we will see later, such a naïve construction of Z_1, \dots, Z_p does not work. But it can provide us some useful intuition towards a proof of the theorem.) Obtaining consistent RCDs for the torsos is actually not difficult, by properly using the small adhesion size of (T, β) . The basic idea is the following. For each node $t \in V(T)$, instead of computing a p -RCD for

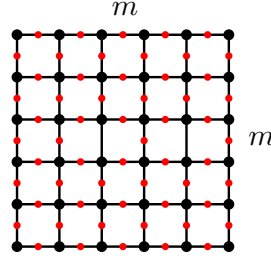
$\text{tor}(t)$, we only compute a p -RCD for $\text{tor}(t) - \sigma(t)$, and how the vertices in $\sigma(t)$ belong to the p classes is determined by the RCDs on the ancestors of t . The decompositions constructed in this way are consistent. Furthermore, since $|\sigma(t)| \leq h$, given a p -RCD for $\text{tor}(t) - \sigma(t)$, no matter how we assign the vertices in $\sigma(t)$ to the p classes, the resulting decomposition is always a p -RCD for $\text{tor}(t)$. Thus, we obtain consistent RCDs for the torsos, which give us the aforementioned partition Z_1, \dots, Z_p of $V(G)$. The remaining question is then whether Z_1, \dots, Z_p is an RCD of G , i.e., whether $\text{tw}(G/(Z_i \setminus Z')) = O(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$.

The only reason for why Z_1, \dots, Z_p could be an RCD of G is clearly the fact that every Z_i satisfies the RCD condition “locally”, i.e., when restricted to a torso. Specifically, let $i \in [p]$ and $Z' \subseteq Z_i$. Set $Z = Z_i \setminus Z'$ for convenience. What we want is $\text{tw}(G/Z) = O(p + |Z'|)$. For each $t \in V(T)$, we have $Z \cap \beta(t) = Z_i^{(t)} \setminus (Z' \cap Z_i^{(t)})$, and thus $\text{tw}(\text{tor}(t)/(Z \cap \beta(t))) = O(p + |Z'|)$. How can we go from the locally bounded treewidth of each $\text{tor}(t)/(Z \cap \beta(t))$ to the global treewidth of G/Z ? The following observation seems useful.

If a graph admits a tree decomposition \mathcal{T} in which each torso is of treewidth at most w , then the graph itself is of treewidth $O(w)$. Indeed, we can “glue” the width- w tree decompositions of the torsos along the given tree decomposition \mathcal{T} to obtain a width- $O(w)$ tree decomposition of the graph (mainly because in a torso the adhesions of the children are cliques).

This observation allows us to go from the treewidth of the torsos to the treewidth of the entire graph. At the first glance, it does not directly apply to our situation here, because we are working on the *contracted* graphs instead of the original ones: our goal is to lift the treewidth bound from the *contracted* torsos $\text{tor}(t)/(Z \cap \beta(t))$ to the *contracted* graph G/Z . However, it is a well-known fact that a tree decomposition of G naturally induces a tree decomposition of the contracted graph G/Z by “contracting” each bag. Specifically, consider the *quotient* map $\pi: V(G) \rightarrow V(G/Z)$ which maps each vertex of G to the corresponding vertex of G/Z . Set $\beta^*(t) = \pi(\beta(t))$ for all $t \in V(T)$. Then (T, β^*) is a tree decomposition of G/Z . Intuitively, the tree decomposition (T, β^*) should allow us to apply the above observation to lift the treewidth bound from $\text{tor}(t)/(Z \cap \beta(t))$ to G/Z , and eventually show that Z_1, \dots, Z_p is an RCD of G .

Although the above argument seems promising, a closer inspection reveals that it actually has a fatal issue, which is also the main barrier to proving Theorem 2. The issue comes from a somewhat counter-intuitive fact: *the contracted torsos are not identical to the torsos of the contracted graph!* Formally, let $\text{tor}^*(t)$ denote the torso of $t \in V(T)$ in the tree decomposition (T, β^*) of G/Z . Then we have $\text{tor}(t)/(Z \cap \beta(t)) \neq \text{tor}^*(t)$ in general, and even worse, the treewidth of $\text{tor}^*(t)$ can be unbounded even if the treewidth of $\text{tor}(t)/(Z \cap \beta(t))$ is bounded. The main reason for why this happens is the following. The edges of $\text{tor}(t)$ do not necessarily appear in G : some of them are manually added to make the adhesions of the children of t cliques (for convenience, we call them *fake edges*). When we contract G to G/Z , only the edges appearing in G can get contracted. However, when we contract $\text{tor}(t)$ to $\text{tor}(t)/(Z \cap \beta(t))$, all fake edges in $\text{tor}(t)[Z \cap \beta(t)]$ also get contracted. This over-contracting can make $\text{tor}(t)/(Z \cap \beta(t))$ much “smaller” than $\text{tor}^*(t)$. To see an example, suppose G is the graph obtained from an $m \times m$ grid by subdividing each edge into two edges (with an intermediate vertex); see Figure 1. So we have $n = O(m^2)$. Consider a tree decomposition (T, β) of G defined as follows. The tree T consists of a root with some children. The bag of the root contains the m^2 grid vertices. Each child of the root corresponds to an edge e of the grid, whose bag contains the two endpoints of e (two grid vertices) and the intermediate vertex for subdividing e . Now the adhesion of each child of the root consists of the two



■ **Figure 1** An $m \times m$ grid with subdivided edges. The black vertices are the grid vertices.

endpoints of the corresponding grid edge. Let $t \in V(T)$ be the root and $Z \subseteq V(G)$ consist of the grid vertices. Then $\text{tor}(t)$ is an $m \times m$ grid and $\text{tor}(t)/(Z \cap \beta(t)) = \text{tor}(t)/Z$ is a single vertex. However, Z is actually an independent set in G . Thus, $G/Z = G$ and we have $\text{tor}^*(t) = \text{tor}(t)$, which is of treewidth m . In fact, this simple example not only demonstrates that $\text{tor}^*(t)$ can have much higher treewidth than $\text{tor}(t)/(Z \cap \beta(t))$, but also directly shows that a partition Z_1, \dots, Z_p satisfying the RCD condition locally (i.e., at each torso) may be not a global RCD in general (and thus our previous construction fails). Suppose now we have an RCD $Z_1^{(t)}, \dots, Z_p^{(t)}$ of $\text{tor}(t)$ for the root $t \in V(T)$. For each child s of t , we arbitrarily assign the only vertex in $\beta(s) \setminus \sigma(s)$ to a class Z_i different from the classes the two vertices in $\sigma(s)$ belong to. Note that, since $|\beta(s)| = 3$, every partition of $\beta(s)$ is actually an RCD of $\text{tor}(s)$. In this way, we obtain a partition Z_1, \dots, Z_p of $V(G)$ that is an RCD when restricted to every torso. However, each Z_i is an independent set of G , and thus $G/Z_i = G$. So unless $p = \Omega(n)$, Z_1, \dots, Z_p is not an RCD of G .

The main technical contribution in our proof is to break the above barrier by constructing Z_1, \dots, Z_p in a much more sophisticated way. On a high-level, we still follow the idea of constructing RCDs locally for each torso and combine them to obtain Z_1, \dots, Z_p . So in what follows, we always use $Z_1^{(t)}, \dots, Z_p^{(t)}$ to denote the restriction of Z_1, \dots, Z_p to $\text{tor}(t)$, i.e., $Z_i^{(t)} := Z_i \cap \beta(t)$. To avoid ambiguity, for a subset $Z \subseteq V(G)$, we use (T, β_Z^*) to denote the tree decomposition of G/Z induced by (T, β) , and use $\text{tor}_Z^*(t)$ to denote the torso of a node $t \in V(T)$ in (T, β_Z^*) . We have seen that the main reason for why $\text{tor}_Z^*(t)$ can have a much higher treewidth than $\text{tor}(t)/(Z \cap \beta(t))$ is the fake edges in $\text{tor}(t)$. So ideally, if $\text{tor}(t)/(Z \cap \beta(t))$ does not contract any fake edge in $\text{tor}(t)$, we are happy. But this is unlikely, because in the worst case (e.g., the grid example above) every edge in $\text{tor}(t)$ is fake. Fortunately, the fake edges in $\text{tor}(t)$ are not always “uncontractable”. Indeed, if a fake edge in $\text{tor}(t)$ has two endpoints lying in the same connected component of $G[Z]$, then we can safely contract it because its two endpoints are also contracted to one vertex in G/Z . Therefore, it is enough to guarantee that $\text{tor}(t)/(Z \cap \beta(t))$ only contracts these safe fake edges, which is equivalent to saying that the two endpoints of every edge of $\text{tor}(t)[Z \cap \beta(t)]$ lie in the same connected component of $G[Z]$. (If this holds, we should be able to somehow relate $\text{tw}(\text{tor}_Z^*(t))$ to $\text{tw}(\text{tor}(t)/(Z \cap \beta(t)))$ and make the previous proof work.) However, this is still too difficult, because we are not dealing with a single set Z . Instead, we have to take care of $Z = Z_i \setminus Z'$ for all $i \in [p]$ and $Z' \subseteq Z_i$, i.e., all subsets of Z_1, \dots, Z_p . Even if the construction of Z_i makes every fake edge uv of $\text{tor}(t)[Z_i \cap \beta(t)]$ have its endpoints u and v in the same connected component of $G[Z_i]$, when a fraction $Z' \subseteq Z_i$ is taken away from Z_i , it can happen that u and v lie in different connected components of $G[Z_i \setminus Z']$ (for example, when Z' is a cut of u and v in $G[Z_i]$) and thus the fake edge uv is no longer safe for contraction when considering $Z = Z_i \setminus Z'$.

The key observation to get rid of this issue is that we do not necessarily need to make *every* fake edge of $\text{tor}(t)[(Z_i \setminus Z') \cap \beta(t)] = \text{tor}(t)[Z_i^{(t)} \setminus Z']$ safe. Indeed, the desired bound for $\text{tw}(\text{tor}_{Z_i \setminus Z'}^*(t))$ and $\text{tw}(G/(Z_i \setminus Z'))$ is $O(p + |Z'|)$, which also depends on $|Z'|$. It turns out that as long as $\text{tor}(t)[Z_i^{(t)} \setminus Z']$ only contains $O(|Z'|)$ “unsafe” fake edges (i.e., those with two endpoints in different connected components of $G[Z_i \setminus Z']$), we can obtain the bound $\text{tw}(\text{tor}_{Z_i \setminus Z'}^*(t)) = O(p + |Z'|)$. To see this, let $Z'' \subseteq (Z_i \setminus Z') \cap \beta(t) = Z_i^{(t)} \setminus Z'$ consist of the endpoints of the $O(|Z'|)$ unsafe fake edges in $\text{tor}(t)[Z_i^{(t)} \setminus Z']$, so we have $|Z''| = O(|Z'|)$. Now every edge in $\text{tor}(t)[Z_i^{(t)} \setminus (Z' \cup Z'')]$ has its two endpoints in the same connected component of $G[Z_i \setminus Z']$, and is thus safe. Since $\text{tor}(t)/(Z_i^{(t)} \setminus (Z' \cup Z''))$ only contracts safe edges, it can be shown that $\text{tor}_{Z_i \setminus Z'}^*(t)$ is (almost¹) a minor of $\text{tor}(t)/(Z_i^{(t)} \setminus (Z' \cup Z''))$. Thus, we can bound $\text{tw}(\text{tor}_{Z_i \setminus Z'}^*(t))$ using $\text{tw}(\text{tor}(t)/(Z_i^{(t)} \setminus (Z' \cup Z'')))$, where the latter is $O(p + |Z' \cup Z''|) = O(p + |Z'|)$, under the assumption that $Z_1^{(t)}, \dots, Z_p^{(t)}$ is an RCD of $\text{tor}(t)$. To summarize, Z_1, \dots, Z_p is an RCD of G if the following conditions hold.

- (A) Z_1, \dots, Z_p is an RCD when restricted to every torso, i.e., $Z_1^{(t)}, \dots, Z_p^{(t)}$ is an RCD of $\text{tor}(t)$.
- (B) All but at most $O(|Z'|)$ edges in $\text{tor}(t)[Z_i^{(t)} \setminus Z']$ have both endpoints in the same connected component of $G[Z_i \setminus Z']$, for all $t \in V(T)$, $i \in [p]$, and $Z' \subseteq Z_i$.

Condition A naturally holds as long as we insist on constructing Z_1, \dots, Z_p by combining local RCDs. So the crucial question now is how to satisfy condition B.

From Global to Local

Condition B is not tractable, because it is a global constraint on Z_1, \dots, Z_p . So we need to somehow reduce it to a local constraint on the RCD of each torso. Let us fix a node $t \in V(T)$ and an index $i \in [p]$. To have condition B, the first thing we need is that (almost) every edge of $\text{tor}(t)[Z_i^{(t)}]$ has its endpoints in the same connected component of $G[Z_i]$, which is the special case $Z' = \emptyset$. But only having this is not enough. We need in addition that losing k vertices in Z_i only makes $O(k)$ fake edges become “unsafe”. To achieve this, our main idea is to require the two endpoints of every edge in $\text{tor}(t)[Z_i^{(t)}]$ to be connected in $G[Z_i]$ by *only* the vertices “strictly below t ” in the tree decomposition (T, β) , that is, the vertices that only appear in the bags of descendants of t . Formally, for every child s of t , denote by $\gamma(s)$ the union of all bags in T_s , the subtree of T rooted at s . Then our requirement is that for every edge uv of $\text{tor}(t)[Z_i^{(t)}]$, u and v lie in the same connected component of $G[(\gamma(s) \setminus \sigma(s)) \cap Z_i] \cup \{u, v\}$ for every child s of t such that $u, v \in \sigma(s)$. Note that every non-fake edge trivially satisfies the requirement.

We show that this requirement is sufficient to guarantee condition B above for all $Z' \subseteq Z_i$. The definition of tree decompositions guarantees that the sets $\gamma(s) \setminus \sigma(s)$ are disjoint for different children s of t . We say a child s of t is *bad* if Z' contains at least one vertex in $\gamma(s) \setminus \sigma(s)$. By the disjointness of the sets $\gamma(s) \setminus \sigma(s)$, the number of bad children of t is at most $|Z'|$. For convenience, we say a child s of t *witnesses* an edge uv of $\text{tor}(t)[Z_i^{(t)}]$ if $u, v \in \sigma(s)$. Note that each child s of t can witness at most $O(h^2)$ edges, because $|\sigma(s)| \leq h$. Due to our requirement, if an edge uv of $\text{tor}(t)[Z_i^{(t)} \setminus Z']$ violates condition B above, i.e., u and v lie in different connected components of $G[Z_i \setminus Z']$, then Z' must contain at least one vertex in $\gamma(s) \setminus \sigma(s)$ for every child s of t satisfying $u, v \in \sigma(s)$, and in particular (u, v) is

¹ More precisely, $\text{tor}_{Z_i \setminus Z'}^*(t)$ is a minor of a graph obtained from $\text{tor}(t)/(Z_i^{(t)} \setminus (Z' \cup Z''))$ by adding few edges.

witnessed by a bad node. Since t has at most $|Z'|$ bad children and each of them can witness $O(h^2)$ edges, the total number of edges of $\text{tor}(t)[Z_i^{(t)} \setminus Z']$ violating condition B is bounded by $O(|Z'|)$.

Based on the above argument, it now suffices to construct p -RCDs for each torso of (T, β) such that the partition Z_1, \dots, Z_p obtained by combining these local RCDs satisfies the aforementioned requirement for all $i \in [p]$ and $t \in V(T)$. However, the current form of our requirement is global, because whether it is satisfied at a node t depends on the construction at not only t but also *all descendants* of t in T . So next, let us transform it to a “local” form which can be checked by looking at the construction at each torso *independently*. We first observe that the following condition implies our previous requirement (which can be shown by a simple induction argument).

For every edge uv of $\text{tor}(t)[Z_i^{(t)}]$, the endpoints u and v lie in the same connected component of $\text{tor}(s)[(Z_i^{(s)} \setminus \sigma(s)) \cup \{u, v\}]$ for every child s of t such that $u, v \in \sigma(s)$.

The above is actually equivalent to saying that for every child s of t , all $u, v \in \sigma(s) \cap Z_i^{(s)}$ lie in the same connected component of $\text{tor}(s)[(Z_i^{(s)} \setminus \sigma(s)) \cup \{u, v\}]$. Importantly, this condition *only* depends on the construction of $Z_1^{(s)}, \dots, Z_p^{(s)}$, the RCD of $\text{tor}(s)$. If this condition holds on every node of T , then our previous requirement is also satisfied for every node of T . Therefore, we have our new goal. For every node $t \in V(T)$, we want the following.

- (A) $Z_1^{(t)}, \dots, Z_p^{(t)}$ is an RCD of $\text{tor}(t)$.
- (B) For all $i \in [p]$, any two vertices $u, v \in \sigma(t) \cap Z_i^{(t)}$ lie in the same connected component of $\text{tor}(t)[(Z_i^{(t)} \setminus \sigma(t)) \cup \{u, v\}]$.

Now the new conditions above are both local, which allows us to consider each torso individually. We shall construct the RCDs of the torsos in a top-down manner from the root of T to the leaves. Suppose we are at the node $t \in V(T)$ and going to construct the RCD $Z_1^{(t)}, \dots, Z_p^{(t)}$ of $\text{tor}(t)$. At this point, the RCD at the parent of t has been constructed, so each vertex in $\sigma(t)$ has already been assigned to one of the p classes $Z_1^{(t)}, \dots, Z_p^{(t)}$. We then compute a p -RCD of $\text{tor}(t) - \sigma(t)$ with an additional property that the i -th class of the RCD “connects” the vertices in $\sigma(t) \cap Z_i^{(t)}$ for all $i \in [p]$. Combining this with the partition of $\sigma(t)$ gives us the desired $Z_1^{(t)}, \dots, Z_p^{(t)}$ satisfying the above two conditions. As long as such a single step can be achieved, we can eventually construct Z_1, \dots, Z_p . So from now, we can restrict ourselves to one torso.

RCD with a Connectivity Constraint

Before discussing how to construct the desired RCD of $\text{tor}(t) - \sigma(t)$, we need another twist to simplify the task in hand. We notice that constructing an RCD of $\text{tor}(t) - \sigma(t)$ satisfying the additional connectivity property is actually *impossible* if we do not add any assumption on how the vertices in $\sigma(t)$ belong to the p classes. To see this, consider the following simple example where $\text{tor}(t)$ is a star of 5 vertices, one central vertex connecting with 4 other vertices. All vertices are contained in $\sigma(t)$ except the central vertex. In $\sigma(t)$, two vertices belong to the class $Z_1^{(t)}$ and the other two vertices belong to $Z_2^{(t)}$. Now in order to connect the two vertices in $\sigma(t) \cap Z_1^{(t)}$, the RCD of $\text{tor}(t) - \sigma(t)$ must assign the central vertex to $Z_1^{(t)}$. But if this is the case, we fail to connect the two vertices in $\sigma(t) \cap Z_2^{(t)}$. Therefore, in this example, it is impossible to construct the desired RCD of $\text{tor}(t) - \sigma(t)$. In order to fix this issue, we have to somehow guarantee the “connectivity task” that $\sigma(t)$ leaves to us is not too complicated.

Fortunately, using a stronger version of Robertson-Seymour decomposition, we are able to reach a nice situation where only *one* class of vertices in $\sigma(t)$ need to be connected by the RCD of $\text{tor}(t) - \sigma(t)$. This version of Robertson-Seymour decomposition, as stated in [6, 9],

guarantees that for every node $t \in V(T)$ and every child s of t , the adhesion $\sigma(s)$ contains at most *three* vertices in the embeddable part of $\text{tor}(t)$ (recall that $\text{tor}(t)$ is an h -almost-embeddable graph consisting of an embeddable part, vortices, and apices). To see why this result helps us, let us consider an ideal case where every torso in the Robertson-Seymour decomposition has no vortices and apices. In this case, every adhesion $\sigma(t)$ is of size at most three, because all vertices in the torso $\text{tor}(t')$ of the parent t' of t are in the embeddable part of $\text{tor}(t')$ and thus $\sigma(t)$ can contain at most three of them. Since $|\sigma(t)| \leq 3$, there can be at most one class $Z_i^{(t)}$ that contains at least two vertices in $\sigma(t)$. Therefore, when constructing the RCD of $\text{tor}(t) - \sigma(t)$, we only need the i -th class to connect the vertices in $\sigma(t) \cap Z_i^{(t)}$. In the general case where the torsos have vortices and apices, the situation becomes complicated. But we can still manage to achieve this, which requires us to construct the RCD of each torso more carefully and then use the “three-vertex” property as above.

Now our goal becomes to construct an RCD of $\text{tor}(t) - \sigma(t)$ in which one class is required to connect the corresponding vertices in $\sigma(t)$. Essentially, we achieve this by proving the following.²

► **Lemma 4 (simplified version).** *Given a connected h -almost-embeddable graph G , a set $\Phi \subseteq V(G)$ of size c , and a number p , one can compute in polynomial time p disjoint sets $Z_1, \dots, Z_p \subseteq V(G)$ such that the following two conditions hold.*

- (1) $\text{tw}(G/(Z_i \setminus Z')) = O_{h,c}(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$.
- (2) Φ is contained in one connected component of $G - \bigcup_{i=1}^p Z_i$.

To see why the above lemma achieves our goal, note that $\text{tor}(t) - \sigma(t)$ is h -almost-embeddable. Furthermore, by constructing the Robertson-Seymour decomposition (T, β) carefully, we can always make $\text{tor}(t) - \sigma(t)$ connected and make every vertex in $\sigma(t)$ have a neighbor in $\beta(t) \setminus \sigma(t)$ in the torso $\text{tor}(t)$. Without loss of generality, suppose we want to connect the vertices in $\sigma(t) \cap Z_k^{(t)}$ using the k -th class of the RCD of $\text{tor}(t) - \sigma(t)$. For each $v \in \sigma(t) \cap Z_k^{(t)}$, we mark a vertex $v' \in \beta(t) \setminus \sigma(t)$ that is neighboring to v in $\text{tor}(t)$. Let $\Phi \subseteq \beta(t) \setminus \sigma(t)$ be the set of marked vertices. Now apply Lemma 4 with $G = \text{tor}(t) - \sigma(t)$ and the set Φ . We then get the disjoint sets $Z_1, \dots, Z_p \subseteq \beta(t) \setminus \sigma(t)$. The vertices in Z_i are assigned to the class $Z_i^{(t)}$ for all $i \in [p]$. Finally, the vertices in $(\beta(t) \setminus \sigma(t)) \setminus (\bigcup_{i=1}^p Z_i)$ are assigned to the class $Z_k^{(t)}$. Then condition 1 of Lemma 4 implies that $Z_1^{(t)}, \dots, Z_p^{(t)}$ is an RCD of $\text{tor}(t)$ and condition 2 guarantees that all $u, v \in \sigma(t) \cap Z_k^{(t)}$ lie in the same connected component of $\text{tor}(s)[(Z_k^{(t)} \setminus \sigma(t)) \cup \{u, v\}]$. Next, we summarize our ideas for proving Lemma 4.

Proof Sketch of Lemma 4

As mentioned at the beginning, the recent work [2] already gives RCDs for almost-embeddable graphs, that is, it proves the special case of Lemma 4 without the set Φ and condition 2. Therefore, it is natural to ask whether one can directly generalize (possibly with slight modifications) the proof in [2] to obtain a proof of Lemma 4. Unfortunately, this is not the case. A close look at the proof in [2] reveals that the construction of Z_1, \dots, Z_p in [2] “inherently” prevents us from having condition 2 of Lemma 4. To see this, we need to briefly review how [2] constructs the sets Z_1, \dots, Z_p .

² The actual result is more complicated, in which the sets Z_1, \dots, Z_p have some additional properties and also there is an additional assumption on the almost-embeddable graph G . As we are not going to cover these technical details in this overview, considering this simplified version should be enough.

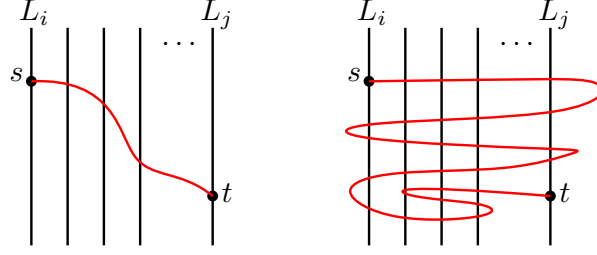
For convenience, we discuss the proof of [2] on a surface-embedded graph instead of an almost-embeddable graph. In fact, the most difficult part of the work [2] also lies in decomposing a surface-embedded graph (specifically, the embeddable part of an almost-embeddable graph), and generalizing to almost-embeddable graphs is somehow easy. In [2], the construction of Z_1, \dots, Z_p itself is simple, while the analysis of the RCD condition is involved. In the first step, it generalizes the *outerplanar layering* for planar graphs to surface-embedded graphs. Recall that the outerplanar layering partitions the vertices of a planar graph G into layers $L_1, \dots, L_m \subseteq V(G)$, where L_i consists of the vertices on the outer boundary of $G - \bigcup_{j=1}^{i-1} L_j$. Thus, L_1 is outer boundary of G , L_2 is the outer boundary of $G - L_1$, and so forth. If G is a graph embedded in a surface Σ , the same layering procedure still applies. Indeed, we can fix a reference point $x_0 \in \Sigma$ and define the outer boundary of a Σ -embedded graph as the boundary of the face containing x_0 (assuming the image of the embedding is disjoint from x_0). Given this, we can layer G in the same way as above, by defining L_i to be the vertices on the outer boundary of $G - \bigcup_{j=1}^{i-1} L_j$. We call this generalization *radial layering* for surface-embedded graphs. Now let L_1, \dots, L_m be the radial layering of G . The analysis in [2] implies the following property of the layers (though not stated explicitly in [2]).

► **Lemma 5.** *If $Z = L_{i_1} \cup \dots \cup L_{i_k}$ where $i_1 < \dots < i_k$ and $|i_j - i_{j-1}| = O(p)$ for all $j \in [k+1]$ (set $i_0 = 0$ and $i_{k+1} = m$ for convenience), then $\mathbf{tw}(G/(Z \setminus Z')) = O(p + |Z'|)$ for all $Z' \subseteq Z$.*

Then [2] simply defines each set Z_i as the union of equidistant layers with distance $O(p)$ apart, that is, $Z_i = L_{q_i} \cup L_{\Delta+q_i} \cup L_{2\Delta+q_i} \cup \dots$ for some number $q_i \in [\Delta]$ where $\Delta = O(p)$. If the choices of q_1, \dots, q_p are different, Z_1, \dots, Z_p are disjoint. By Lemma 5, Z_1, \dots, Z_p satisfy the RCD condition.

Now let us see what is the difficulty to generalize this construction so that it further satisfies condition 2 of Lemma 4. Consider the given set Φ in Lemma 4, which is of size c . We want Φ to be contained in the complement $G - \bigcup_{i=1}^p Z_i$, and more importantly, Φ has to be in one connected component of $G - \bigcup_{i=1}^p Z_i$. To make Φ contained in $G - \bigcup_{i=1}^p Z_i$ is easy. We can mark the layers L_i intersecting Φ as “bad” layers. There can be at most c bad layers. When constructing Z_1, \dots, Z_p , we do not include these bad layers. According to Lemma 5, we can still guarantee that Z_1, \dots, Z_p satisfy condition 1 of Lemma 4 even if we miss all bad layers, because the constant hidden in the bound of condition 1 can depend on c . However, to further require the vertices in Φ lying in the same connected component of $G - \bigcup_{i=1}^p Z_i$ is impossible. Indeed, one can easily see that each layer L_i is a separator in G that separates the layers L_1, \dots, L_{i-1} from the layers L_{i+1}, \dots, L_m . The layers in Z_1, \dots, Z_p separate the remaining part of G into many “small” pieces where each piece consists of at most $O(p)$ consecutive layers. This highly-disconnected structure of $G - \bigcup_{i=1}^p Z_i$ naturally prevents the vertices in Φ from lying in the same connected component, unless the layers containing Φ are all close to each other.

Based on the above discussion, in order to satisfy both conditions in Lemma 4, we need to significantly modify the construction of [2] with various new ideas. As we have seen, in the previous construction, the layers in Z_1, \dots, Z_p serve as “barriers” that prevent the vertices in Φ from being connected in the complement graph. Therefore, a natural idea is to “break” these layers a little bit (i.e., remove some vertices from Z_1, \dots, Z_p) so that the vertices in Φ can go across the barriers to connect with each other. However, by doing this, the layers in each Z_i become broken, and thus the new Z_i might violate the RCD condition, i.e., condition 1 of Lemma 4 (as we have fewer vertices to contract). As such, we have to break the layers in some *structured* way such that the vertices in Φ can be connected in the



■ **Figure 2** Monotone path vs. back-and-forth path.

complement graph and *simultaneously* the RCD condition of Z_1, \dots, Z_p preserves, which is the main challenge in our construction. At this point, it is totally not clear how to achieve this goal. So next, we begin with some simple intuitions.

Let us consider the simplest case where Φ only contains two vertices s and t . In this case, it suffices to properly find an (s, t) -path π in G and then remove the vertices in Z_1, \dots, Z_p on this path. Since we do not want to lose the RCD condition of Z_1, \dots, Z_p , π should break each layer in Z_1, \dots, Z_p as “little” as possible. So intuitively, a path that visits the layers L_1, \dots, L_m *monotonely* (left part of Figure 2) is preferable to a path that goes back and forth across the layers many times (right part of Figure 2). If we do not have any requirement on the embedding of G in Σ , it is not always possible to find a monotone (or mostly monotone) path connecting s and t . Thus, at the beginning (before the radial layering), we need to first modify the embedding of G to a nice one, and do everything with respect to the nice embedding. For surface graphs, a 2-cell embedding, in which each face is homeomorphic to a disk, is the nice embedding we need. One can always compute a 2-cell embedding for G in polynomial time as long as the genus of G is a constant. (For almost-embeddable graphs, however, the situation is more involved, as we are not able to arbitrarily change the embedding of the embeddable part because of the vortices. We have to define another type of embedding for which we can safely modify a given embedding of the embeddable part to.) Now if L_1, \dots, L_m are the radial layers for a 2-cell embedding of G , then one can go from every vertex in a layer L_j to the previous layer L_{j-1} by walking around the boundary of a face in between L_{j-1} and L_j . It turns out that for every vertex $v \in L_j$ and index $i \leq j$, there exists a path from v to (some vertex in) L_i that visits L_j, L_{j-1}, \dots, L_i monotonely. Suppose $s \in L_i$ and $t \in L_j$ where $i \leq j$. Note that although we can go from t to L_i monotonely, there does not necessarily exist a monotone path from t to s . So the key idea here is to, instead of using one path connecting s and t , construct two monotone paths π_s and π_t , where π_s (resp., π_t) goes from s (resp., t) to L_1 . Then we do not include the layer L_1 in any of Z_1, \dots, Z_p (which is fine according to Lemma 5). Because L_1 is the outer boundary of G and the embedding of G is 2-cell, $G[L_1]$ is connected. Therefore, L_1 together with the two paths π_s and π_t connects s and t . The same idea directly generalizes to the case where Φ has more than two vertices. For each vertex $v \in \Phi$, we try to construct a path π_v which goes monotonely from v to L_1 . Now L_1 together with all the paths connect everything in Φ . It then remains to show how to construct these c monotone paths carefully so that they do not break the layers in Z_1, \dots, Z_p too much.

Since c is treated as a constant in Lemma 4, if we are able to construct one path, it is not surprising that the same idea generalizes to constructing c paths. Thus, in what follows, we focus on the case $c = 1$. We have $\Phi = \{v\}$. Since the path π from v to L_1 we are going to construct is monotone, it crosses each layer L_i at most once, that is, after it leaves L_i

for L_{i-1} , it never comes back to L_i . Therefore, we construct each part $L_i \cap \pi$ of π from larger i to smaller i iteratively. When constructing $L_i \cap \pi$, we basically need to consider how to walk in the layer L_i from an arbitrary starting vertex $s \in L_i$ to an “exit” vertex that is adjacent to L_{i-1} so that the walk does not break L_i too much. To this end, we have to figure out what “too much” means. In other words, how much can we break each layer L_i so that Z_1, \dots, Z_p still satisfy the RCD condition? By checking the proof of Lemma 5 in [2], we see that the bound in Lemma 5 mainly relies on the following two properties of each layer L_i (where $L_{>i}$ is the union of L_{i+1}, \dots, L_m and $L_{\leq i}$ is the union of L_1, \dots, L_i).

- (I) The neighbors of each connected component of $G[L_{>i}]$ lie in one face of $G[L_{\leq i}]$.
- (II) For every $L' \subseteq L_i$, each connected component of $G[L_{>i}]$ is adjacent to $O(|L'|)$ vertices in the graph $G/(L_i \setminus L')$.

Using the above two properties and the fact that the layers in each of Z_1, \dots, Z_p are only $O(p)$ distance apart, one can finally show that Z_1, \dots, Z_p satisfies the RCD condition. Now we have the path π that breaks L_i . So we can only include in Z_1, \dots, Z_p the part $L_i \setminus \pi$. In this case, in order to make the proof work, we need the modified version of condition II above: for every $L' \subseteq L_i \setminus \pi$, each connected component of $G[L_{>i}]$ is adjacent to $O(|L'|)$ vertices in the graph $G/((L_i \setminus \pi) \setminus L')$. However, it is easy to see that this is impossible. Consider the simplest case where $L' = \emptyset$. We want each connected component of $G[L_{>i}]$ to have $O(1)$ neighbors in $G/(L_i \setminus \pi)$. But in the worst case, after π reaches L_i , it needs to go inside L_i for a long distance in order to arrive at an exit vertex to leave L_i for L_{i-1} . Therefore, it can happen that π contains a large fraction of L_i in which many vertices can be adjacent to the same connected component C of $G[L_{>i}]$. As these vertices are not contracted in $G/(L_i \setminus \pi)$, the number of neighbors of C in $G/(L_i \setminus \pi)$ can be unbounded.

Going deeper into the proof of [2] shows that the reason for why the above two properties help is basically the following. These two properties allow us to partition G into two parts $G[L_{\leq i}]$ and $G[L_{>i}]$ such that the connection between these two parts becomes “weak” after contracting a large subset of L_i , namely, each connected component of $G[L_{>i}]$ is only adjacent to few vertices in $G[L_{\leq i}]$ which lie in one face of $G[L_{\leq i}]$ (before contraction). Now because the part in $L_i \cap \pi$ cannot be contracted, we are no longer able to have a weak connection between $G[L_{\leq i}]$ and $G[L_{>i}]$. To get rid of this issue, our key idea here is to partition G into two parts in a different way. Specifically, when determining the part of π contained in L_i , we also compute another subset $L_i^+ \subseteq L_i$. Then we partition G into two parts $G[L_{\leq i} \setminus L_i^+]$ and $G[L_{>i} \cup L_i^+]$, that is, we move L_i^+ from the part $G[L_{\leq i}]$ to the part $G[L_{>i}]$. The choice of L_i^+ will guarantee the aforementioned weak connection between the two new parts. Formally, L_i^+ (and π) satisfies the following properties.

- (I) The neighbors of each connected component of $G[L_{>i} \cup L_i^+]$ lie in one face of $G[L_{\leq i}]$ (and therefore one face of $G[L_{\leq i} \setminus L_i^+]$).
- (II) For every $L' \subseteq L_i \setminus \pi$, each connected component of $G[L_{>i} \cup L_i^+]$ is adjacent to $O(|L'|)$ vertices in the graph $G/((L_i \setminus \{\pi \cup L_i^+\}) \setminus L')$.

Note that such a set L_i^+ does not always exist for an arbitrary path π . Therefore, we have to construct $L_i \cap \pi$ and L_i^+ simultaneously, and both of them need to be constructed carefully. As this step is technical and requires insights for surface-embedded graphs, we are not going to discuss it in detail. Essentially, once we are able to construct each part $L_i \cap \pi$ of π and the corresponding set L_i^+ satisfying the above two conditions, we can combine them to obtain the desired path π from v to L_1 and show that $\text{tw}(G/((Z_i \setminus \pi) \setminus Z')) = O(p + |Z'|)$ for all $Z' \subseteq Z_i \setminus \pi$. The sets L_i^+ are only used in the analysis of the treewidth bounds, and the analysis is built on the argument in [2]. The same idea generalizes to the case where we need to construct c paths (with a bit more work). With this in hand, we are finally able to prove Lemma 4 for surface-embedded graphs. Of course, for almost-embeddable graphs, there are more technical details to be dealt with, but the basic idea remains the same.

Minimal Embeddings of Almost-Embeddable Graphs

In the last part of this overview, we discuss an interesting intermediate result achieved in our proof. As aforementioned, we can modify the embedding of a (connected) surface graph to a 2-cell embedding. However, for an almost-embeddable graph, we cannot require its embeddable part is 2-cell embedded. There are two reasons. First, when we change the embedding of the embeddable part, the structure of the vortices might be lost. Second, even if the graph itself is connected, the subgraph excluding the apices is not necessarily connected, and thus does not admit a 2-cell embedding. In order to make our proof work, we define a variant of 2-cell embedding, which we call *minimal embedding*. Basically, an almost-embeddable graph G whose embeddable part is embedded with a minimal embedding satisfies the following condition. Let f be a face of the embeddable part of G , and $\tilde{\partial}f$ be the subgraph of G consisting of the boundary of f and all vortices contained in f . Then different connected components of $\tilde{\partial}f$ belong to different connected components of $G - A$ where A is the set of apices of G . If G does not have vortices and apices (i.e., G is a surface graph), then the above condition is equivalent to saying that the boundary of every face of G is connected. We show that given an almost-embeddable graph G , we can compute (in polynomial time) a new almost-embeddable structure for G in which the embeddable part is embedded with a minimal embedding. This result is important for our proof, and might be of independent interest.

3 Applications

In this section, we briefly discuss some applications of Theorem 2 in parameterized complexity building on [2, 24]. Indeed, Theorem 2 directly results in parameterized algorithms of running time $n^{O(\sqrt{k})}$ or $2^{\tilde{O}(\sqrt{k})} \cdot n^{O(1)}$ for a broad class of vertex/edge deletion problems on H -minor-free graphs. For example, this includes all problems that can be formulated as PERMUTATION CSP DELETION or 2-CONN PERMUTATION CSP DELETION [24].

An instance of the (binary) CSP problem is specified by a triple $\Gamma = (X, D, \mathcal{C})$ where X is a finite set of variables, D is a finite domain, and \mathcal{C} is a set of constraints each of which is of the form $c = (x, y, R)$ where $x, y \in X$ and $R \subseteq D^2$. We say Γ is a *permutation CSP instance* if for every constraint $c = (x, y, R) \in \mathcal{C}$ it holds that $|\{b \in D \mid (a, b) \in R\}| \leq 1$ and $|\{b \in D \mid (b, a) \in R\}| \leq 1$ for all $a \in D$. We write $|\Gamma| = |X| + |D|$ as the *size* of Γ . An *assignment* of Γ is a function $\alpha : Y \rightarrow D$ on a subset $Y \subseteq X$, and we say α is *satisfying* if for all $c = (x, y, R) \in \mathcal{C}$ such that $x, y \in Y$, we have $(\alpha(x), \alpha(y)) \in R$. A permutation CSP instance $\Gamma = (X, D, \mathcal{C})$ naturally induces a graph G_Γ with vertex set X in which two variables $x, y \in X$ are connected by an edge if there exists $c \in \mathcal{C}$ such that $c = (x, y, R)$ for some $R \subseteq D^2$. We say the permutation CSP instance Γ is *H-minor-free* if its underlying graph G_Γ is H -minor-free. For a subset $\mathcal{C}' \subseteq \mathcal{C}$ of constraints, we write $\Gamma - \mathcal{C}' = (X, D, \mathcal{C} \setminus \mathcal{C}')$. For a subset $X' \subseteq X$ of variables, we write $\Gamma - X' = (X \setminus X', D, \mathcal{C} \setminus \mathcal{C}_{X'})$, where $\mathcal{C}_{X'} \subseteq \mathcal{C}$ consists of all constraints that involve at least one variable from X' .

Let $\Gamma = (X, D, \mathcal{C})$ be a permutation CSP instance. A *size constraint* for Γ is a pair (w, δ) where $w : X \times D \rightarrow \mathbb{N}$ is a weight function and $\delta \in \mathbb{N}$ is a threshold. We say an assignment $\alpha : Y \rightarrow D$ of Γ *respects* the size constraint (w, δ) if $\sum_{y \in Y} w(y, \alpha(y)) \leq \delta$. We say Γ is *satisfiable* subject to (w, δ) on a subset $Y \subseteq X$ if there is a satisfying assignment $\alpha : Y \rightarrow D$ of Γ that respects (w, δ) .

3.1 Permutation CSP Deletion Problems

In [24] a subset of the authors define the problems PERMUTATION CSP VERTEX DELETION and PERMUTATION CSP EDGE DELETION. These problems essentially ask whether one can remove from a given permutation CSP instance a small number of variables (resp., constraints), or equivalently vertices (resp., edges) in the underlying graph, such that the resulting instance is satisfiable on every connected component of the underlying graph. The formal definitions are the following³.

The PERMUTATION CSP VERTEX DELETION problem is a parameterized problem that takes as input a tuple (Γ, w, δ, k) , where $\Gamma = (X, D, \mathcal{C})$ is a permutation CSP instance, (w, δ) is a size constraint for Γ , and k is the solution size parameter. The goal is to find a subset $X' \subseteq X$ with $|X'| \leq k$ such that $\Gamma - X'$ is satisfiable subject to (w, δ) ⁴ on every subset $Y \subseteq X \setminus X'$ satisfying that $G_{\Gamma - X'}[Y]$ is a connected component of $G_{\Gamma - X'}$.

Similarly, the PERMUTATION CSP EDGE DELETION problem takes the same input, but the goal is to find a subset $\mathcal{C}' \subseteq \mathcal{C}$ with $|\mathcal{C}'| \leq k$ such that $\Gamma - \mathcal{C}'$ is satisfiable subject to (w, δ) on every subset $Y \subseteq X$ satisfying that $G_{\Gamma - \mathcal{C}'}[Y]$ is a connected component of $G_{\Gamma - \mathcal{C}'}$.

► **Theorem 6.** *Let H be a fixed graph. Then the PERMUTATION CSP VERTEX DELETION (or the PERMUTATION CSP EDGE DELETION) problem on H -minor-free instances (Γ, w, δ, k) can be solved in $(|\Gamma| + \delta)^{O(\sqrt{k})}$ time.*

Proof. It is shown in [24] that if Theorem 2 (which is stated as a conjecture in [24]) is true, then PERMUTATION CSP VERTEX/EDGE DELETION on H -minor-free instances can be solved in $(|\Gamma| + \delta)^{O(\sqrt{k})}$ time, which completes the proof. In the following, we briefly sketch the details (the same ideas are also used in [2]).

We focus on the vertex-deletion version. Let (Γ, w, δ, k) be the input instance and let $G = G_\Gamma$ be the underlying graph of Γ which is H -minor-free. Using Theorem 2, we can compute the partition $Z_1, \dots, Z_p \subseteq V(G)$ for $p = \lceil \sqrt{k} \rceil$. Suppose $S \subseteq V(G)$ is an unknown solution for (Γ, w, δ, k) . Since $|S| \leq k$ there exists some $i \in [p]$ such that $|Z_i \cap S| \leq \sqrt{k}$. We guess the index i and the vertices in $Z_i \cap S$. Note that the number of guesses is bounded by $k \cdot |\Gamma|^{\sqrt{k}} = |\Gamma|^{O(\sqrt{k})}$. Now suppose we already know i and $Z_i \cap S$. Thus, we know that there exists a feasible solution that is disjoint from $Z_i \setminus (Z_i \cap S)$. We can find such a solution by applying the standard dynamic programming approach on a tree decomposition of $G/(Z_i \setminus (Z_i \cap S))$ with running time $(|D| + \delta)^{O(w)} \cdot |\Gamma|^{O(1)}$ where w is the width of the tree decomposition and D is the domain of Γ . By Theorem 2, $\text{tw}(G/(Z_i \setminus (Z_i \cap S))) = O(\sqrt{k})$ and the desired $(|\Gamma| + \delta)^{O(\sqrt{k})}$ -time follows.

The key insight for the DP on the tree decomposition is the following. Since $Z_i \setminus (Z_i \cap S)$ is disjoint from the solution, these vertices are “undeletable” variables of Γ . Since we restrict our attention to *permutation* CSP, each connected component of $G[Z_i \setminus (Z_i \cap S)]$ can have only $|D|$ satisfying assignments (since the value of one variable already determines the values of the others). This essentially allows us to treat each connected component of $G[Z_i \setminus (Z_i \cap S)]$ as a single vertex (or variable). Equivalently, we can contract the part $Z_i \setminus (Z_i \cap S)$ in G and do DP on a tree decomposition of $G/(Z_i \setminus (Z_i \cap S))$. ◀

³ We remark that our definition of (2-connected) permutation CSP deletion is a simplified version of the original definition in [24], which is already sufficient for our applications. The original definition is more complicated and allows multiple size constraints of a more general form.

⁴ Strictly speaking, (w, δ) is a size constraint for Γ instead of $\Gamma - X'$. But it naturally induces a size constraint for $\Gamma - X'$ by restricting w to $(X \setminus X') \times D$. For convenience, we still denote it by (w, δ) here.

Many natural problems can be formulated as PERMUTATION CSP VERTEX/EDGE DELETION problems including the following examples (see [24] for more details), which leads us to the following corollary.

► **Corollary 7.** *There exist algorithms with running time $n^{O(\sqrt{k})}$ for ODD CYCLE TRANSVERSAL, VERTEX MULTIWAY CUT, VERTEX MULTICUT, GROUP FEEDBACK VERTEX SET, COMPONENT ORDER CONNECTIVITY, and the edge-deletion version of these problems on H -minor-free graphs, where n is the number of vertices of the input graph and k is the solution-size parameter.*

Proof. All problems except VERTEX MULTICUT (and EDGE MULTICUT) can be formulated as PERMUTATION CSP VERTEX/EDGE DELETION problems, so the $n^{O(\sqrt{k})}$ -time algorithms directly follow from Theorem 6. The problem VERTEX MULTICUT (and EDGE MULTICUT) can be solved in exactly the same way as described in the proof of Theorem 6 (see also [2, Section 5.3]). ◀

Using the minor-preserving (quasi-)polynomial kernels from [24] or the (quasi-)polynomial-size candidate sets given in [2], we can also obtain (randomized) subexponential-time FPT algorithms for these problems. Here the notation $\tilde{O}(\cdot)$ hides $\log k$ factors.

► **Corollary 8.** *There exist randomized algorithms with running time $2^{\tilde{O}(\sqrt{k})} \cdot n^{O(1)}$ for ODD CYCLE TRANSVERSAL, VERTEX MULTIWAY CUT, GROUP FEEDBACK VERTEX SET (for a fixed group), and the edge-deletion version of these problems on H -minor-free graphs, where n is the number of vertices of the input graph and k is the solution-size parameter.*

3.2 Two-Connected Permutation CSP Deletion Problems

To extend the scope of problems covered by this approach, [24] also considers the 2-CONN PERMUTATION CSP DELETION problem. The problem is similar to PERMUTATION CSP DELETION defined above, but the satisfiability we care about is on the 2-connected components of the underlying graph (after deletion), instead of connected components.

The 2-CONN PERMUTATION CSP VERTEX DELETION problem is a parameterized problem that takes as input a tuple (Γ, w, δ, k) , where $\Gamma = (X, D, \mathcal{C})$ is a permutation CSP instance, (w, δ) is a size constraint for Γ , and k is the solution size parameter. The goal is to find a subset $X' \subseteq X$ with $|X'| \leq k$ such that $\Gamma - X'$ is satisfiable subject to $(w, \delta)^5$ on every subset $Y \subseteq X \setminus X'$ satisfying that $G_{\Gamma-X'}[Y]$ is a 2-connected component of $G_{\Gamma-X'}$.

Similarly, the 2-CONN PERMUTATION CSP EDGE DELETION problem takes the same input, but the goal is to find a subset $\mathcal{C}' \subseteq \mathcal{C}$ with $|\mathcal{C}'| \leq k$ such that $\Gamma - \mathcal{C}'$ is satisfiable subject to (w, δ) on every subset $Y \subseteq X$ satisfying that $G_{\Gamma-\mathcal{C}'}[Y]$ is a 2-connected component of $G_{\Gamma-\mathcal{C}'}$.

► **Theorem 9.** *Let H be a fixed graph. Then the 2-CONN PERMUTATION CSP VERTEX DELETION (or the 2-CONN PERMUTATION CSP EDGE DELETION) problem on H -minor-free instances (Γ, w, δ, k) can be solved in $(|\Gamma| + \delta)^{O(\sqrt{k})}$ time.*

Proof. Again, it is shown in [24] that if Theorem 2 (which is stated as a conjecture in [24]) is true, then 2-CONN PERMUTATION CSP VERTEX/EDGE DELETION on H -minor-free instances can be solved in $(|\Gamma| + \delta)^{O(\sqrt{k})}$ time, which completes the proof. Unlike

⁵ Strictly speaking, (w, δ) is a size constraint for Γ instead of $\Gamma - X'$. But it naturally induces a size constraint for $\Gamma - X'$ by restricting w to $(X \setminus X') \times D$. For convenience, we still denote it by (w, δ) here.

Theorem 6, the algorithm for 2-CONN PERMUTATION CSP VERTEX/EDGE DELETION is rather complicated. Roughly speaking, it is designed by first applying Theorem 2 to obtain a so-called “body guessing” lemma and then using the body guessing lemma to (essentially) guess the 2-connected components in the solution (together with a DP on the tree decomposition). Discussing the technical details of this algorithm is out of the scope of this paper, and we refer the interested reader to [24]. ◀

The following problems can be formulated as 2-connected permutation CSP deletion problems (see [24] for more details), which leads us to the following corollary.

► **Corollary 10.** *There exist algorithms with running time $n^{O(\sqrt{k})}$ for SUBSET FEEDBACK VERTEX SET, SUBSET ODD CYCLE TRANSVERSAL, SUBSET GROUP FEEDBACK VERTEX SET, 2-CONN COMPONENT ORDER CONNECTIVITY, and the edge-deletion version of these problems on H -minor-free graphs, where n is the number of vertices of the input graph and k is the solution-size parameter.*

Using the minor-preserving (quasi-)polynomial kernels for SUBSET FEEDBACK VERTEX SET and a reduction from SUBSET FEEDBACK EDGE SET to SUBSET FEEDBACK VERTEX SET given in [24], we can also obtain (randomized) subexponential-time FPT algorithms for these two problems.

► **Corollary 11.** *There exist randomized algorithms with running time $2^{\tilde{O}(\sqrt{k})} \cdot n^{O(1)}$ for SUBSET FEEDBACK VERTEX SET and SUBSET FEEDBACK EDGE SET on H -minor-free graphs, where n is the number of vertices of the input graph and k is the solution-size parameter.*

References

- 1 Brenda S. Baker. Approximation algorithms for np-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994. doi:10.1145/174644.174650.
- 2 Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, and Jie Xue. Subexponential parameterized algorithms for cut and cycle hitting problems on H -minor-free graphs. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 2063–2084. SIAM, 2022. doi:10.1137/1.9781611977073.82.
- 3 Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, and Jie Xue. True contraction decomposition and almost ϵ -tight bipartization for unit-disk graphs. *ACM Trans. Algorithms*, 20(3):20, 2024. doi:10.1145/3656042.
- 4 Thang Nguyen Bui and Andrew Peck. Partitioning planar graphs. *SIAM J. Comput.*, 21(2):203–215, 1992. doi:10.1137/0221016.
- 5 Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H -minor-free graphs. *J. ACM*, 52(6):866–893, 2005. doi:10.1145/1101821.1101823.
- 6 Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 637–646. IEEE Computer Society, 2005. doi:10.1109/SFCS.2005.14.
- 7 Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Contraction decomposition in h -minor-free graphs and algorithmic applications. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 441–450. ACM, 2011. doi:10.1145/1993636.1993696.

- 8 Erik D. Demaine, MohammadTaghi Hajiaghayi, and Bojan Mohar. Approximation algorithms via contraction decomposition. *Comb.*, 30(5):533–552, 2010. doi:10.1007/s00493-010-2341-5.
- 9 Matt DeVos, Guoli Ding, Bogdan Oporowski, Daniel P. Sanders, Bruce A. Reed, Paul D. Seymour, and Dirk Vertigan. Excluding any graph as a minor allows a low tree-width 2-coloring. *J. Comb. Theory B*, 91(1):25–41, 2004. doi:10.1016/j.jctb.2003.09.001.
- 10 Frederic Dorn, Fedor V. Fomin, Daniel Lokshantov, Venkatesh Raman, and Saket Saurabh. Beyond bidimensionality: Parameterized subexponential algorithms on directed graphs. *Inf. Comput.*, 233:60–70, 2013. doi:10.1016/j.ic.2013.11.006.
- 11 Zdenek Dvorák. Thin graph classes and polynomial-time approximation schemes. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1685–1701. SIAM, 2018. doi:10.1137/1.9781611975031.110.
- 12 David Eppstein. Subgraph isomorphism in planar graphs and related problems. *J. Graph Algorithms Appl.*, 3(3):1–27, 1999. doi:10.7155/jgaa.00014.
- 13 David Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000. doi:10.1007/s004530010020.
- 14 Fedor V. Fomin, Daniel Lokshantov, Sudeshna Kolay, Fahad Panolan, and Saket Saurabh. Subexponential algorithms for rectilinear Steiner tree and arborescence problems. *ACM Trans. Algorithms*, 16(2):21:1–21:37, 2020. doi:10.1145/3381420.
- 15 Fedor V. Fomin, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. Subexponential parameterized algorithms for planar and apex-minor-free graphs via low treewidth pattern covering. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 515–524. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.62.
- 16 Martin Grohe. Local tree-width, excluded minors, and approximation algorithms. *Comb.*, 23(4):613–632, 2003. doi:10.1007/s00493-003-0037-9.
- 17 Ken-ichi Kawarabayashi, Robin Thomas, and Paul Wollan. Quickly excluding a non-planar graph. *CoRR*, abs/2010.12397, 2020. arXiv:2010.12397.
- 18 Ken-ichi Kawarabayashi and Mikkell Thorup. The minimum k-way cut of bounded size is fixed-parameter tractable. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 160–169. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.53.
- 19 Sanjeev Khanna and Rajeev Motwani. Towards a syntactic characterization of PTAS. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 329–337. ACM, 1996. doi:10.1145/237814.237979.
- 20 Philip N. Klein. A subset spanner for planar graphs, with application to subset TSP. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 749–756. ACM, 2006. doi:10.1145/1132516.1132620.
- 21 Philip N. Klein. A linear-time approximation scheme for TSP in undirected planar graphs with edge-weights. *SIAM J. Comput.*, 37(6):1926–1952, 2008. doi:10.1137/060649562.
- 22 Philip N. Klein and Dániel Marx. Solving planar k-terminal cut in $O(n^{c\sqrt{k}})$ time. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 569–580. Springer, 2012. doi:10.1007/978-3-642-31594-7_48.
- 23 Philip N. Klein and Dániel Marx. A subexponential parameterized algorithm for subset TSP on planar graphs. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual*

- ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1812–1830. SIAM, 2014. doi:10.1137/1.9781611973402.131.
- 24 Dániel Marx, Pranabendu Misra, Daniel Neuen, and Prafullkumar Tale. A framework for parameterized subexponential algorithms for generalized cycle hitting problems on planar graphs. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 2085–2127. SIAM, 2022. doi:10.1137/1.9781611977073.83.
 - 25 Dániel Marx, Marcin Pilipczuk, and Michał Pilipczuk. On subexponential parameterized algorithms for Steiner tree and directed subset TSP on planar graphs. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 474–484. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00052.
 - 26 Jesper Nederlof. Detecting and counting small patterns in planar graphs in subexponential parameterized time. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1293–1306. ACM, 2020. doi:10.1145/3357713.3384261.
 - 27 Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Contraction decomposition in unit disk graphs and algorithmic applications in parameterized complexity. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1035–1054. SIAM, 2019. doi:10.1137/1.9781611975482.64.
 - 28 Neil Robertson and Paul D. Seymour. Graph minors. XVI. excluding a non-planar graph. *J. Comb. Theory B*, 89(1):43–76, 2003. doi:10.1016/S0095-8956(03)00042-X.
 - 29 Siamak Tazari. Faster approximation schemes and parameterized algorithms on (odd-)h-minor-free graphs. *Theor. Comput. Sci.*, 417:95–107, 2012. doi:10.1016/j.tcs.2011.09.014.
 - 30 Dimitrios M. Thilikos and Sebastian Wiederrecht. Excluding surfaces as minors in graphs. *CoRR*, abs/2306.01724, 2023. arXiv:2306.01724.
 - 31 Magnus Wahlström. On quasipolynomial multicut-mimicking networks and kernelization of multiway cut problems. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 101:1–101:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.101.