SIAM J. COMPUT.
Vol. 51, No. 6, pp. 1866-1930

SUBEXPONENTIAL PARAMETERIZED ALGORITHMS FOR PLANAR AND APEX-MINOR-FREE GRAPHS VIA LOW TREEWIDTH PATTERN COVERING*

FEDOR V. FOMIN†, DANIEL LOKSHTANOV‡, DÁNIEL MARX§, MARCIN PILIPCZUK¶, MICHAŁ PILIPCZUK¶, AND SAKET SAURABH‖

Abstract. We prove the following theorem. Given a planar graph G and an integer k, it is possible in polynomial time to randomly sample a subset A of vertices of G with the following properties: A induces a subgraph of G of treewidth $\mathcal{O}(\sqrt{k}\log k)$, and for every connected subgraph H of G on at most k vertices, the probability that A covers the whole vertex set of H is at least $(2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot n^{\mathcal{O}(1)})^{-1}$, where n is the number of vertices of G. Together with standard dynamic programming techniques for graphs of bounded treewidth, this result gives a versatile technique for obtaining (randomized) subexponential-time parameterized algorithms for problems on planar graphs, usually with running time bound $2^{\mathcal{O}(\sqrt{k}\log^2 k)}n^{\mathcal{O}(1)}$. The technique can be applied to problems expressible as searching for a small, connected pattern with a prescribed property in a large host graph; examples of such problems include DIRECTED k-PATH, WEIGHTED k-PATH, VERTEX COVER LOCAL SEARCH, and SUBGRAPH ISOMORPHISM, among others. Up to this point, it was open whether these problems could be solved in subexponential parameterized time on planar graphs, because they are not amenable to the classic technique of bidimensionality. Furthermore, all our results hold in fact on any class of graphs that exclude a fixed apex graph as a minor, in particular on graphs embeddable in any fixed surface.

Key words. parameterized complexity, subexponential algorithms, treewidth, planar graphs, subgraph isomorphism

MSC codes. 05C75, 05C85, 68R10

DOI. 10.1137/19M1262504

1. Introduction.

Most of the natural NP-hard problems on graphs remain NP-hard even when the input graph is restricted to be planar. However, it was realized

*Received by the editors May 17, 2019; accepted for publication (in revised form) August 9, 2022; published electronically December 20, 2022. An extended abstract of this work was presented at the 57th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2016.
https://doi.org/10.1137/19M1262504
Funding: This work was supported by the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreements 267959 (first author), 280152 (third author), and 306992 (sixth author). The work of the third author was also supported by Hungarian Scientific Research Fund (OTKA) grant NK105645. The work of the fourth author was supported by Polish National Science Centre grant UMO-2013/09/B/ST6/03136. The work of the fifth author was supported by Polish National Science Centre grant UMO-2013/11/D/ST6/03073 and the Foundation for Polish Science (FNP) via the START stipend programme.

†Department of Informatics, University of Bergen, N-5020 Bergen, Norway (fomin@ii.uib.no).
‡Department of Computer Science, University of California, Santa Barbara, CA 93106-5110 USA (daniello@ucsb.edu).
§CISPA Helmholtz Center for Information Security, Saarland Informatics Campus, 66123 Saarbrücken, Germany (dmarx@cs.bme.hu).
¶Institute of Informatics, University of Warsaw, 02-097 Warsaw, Poland (malcin@mimuw.edu.pl, michal.pilipczuk@mimuw.edu.pl).
‖Institute of Mathematical Sciences, Chennai 600 113, India (saket@imsc.res.in), and Department of Informatics, University of Bergen, N-5020 Bergen, Norway (Saket.Saurabh@ii.uib.no).

already in the dawn of algorithm design that the planarity of the input can be exploited algorithmically. Using the classic planar separator theorem of Lipton and Tarjan [30], one can design algorithms working in subexponential time, usually of the form $2^{\mathcal{O}(\sqrt{n})}$ or $2^{\mathcal{O}(\sqrt{n}\log n)}$, for a wide variety of problems that behave well with respect to separators; such running time cannot be achieved on general graph unless the exponential time hypothesis (ETH) fails [27]. From the modern perspective, the planar separator theorem implies that a planar graph on n vertices has treewidth $\mathcal{O}(\sqrt{n})$, and the obtained tree decomposition can be used to run a divide-and-conquer algorithm or a dynamic programming subroutine. The idea of exploiting small separators plays a crucial role in modern algorithm design on planar graphs and related graph classes, including polynomial-time, approximation, and parameterized algorithmic paradigms.

Let us take a closer look at the area of parameterized complexity. For most parameterized NP-hard problems in general graphs the exponential dependence on the parameter is the best we can hope for, assuming ETH. However, there are plenty of problems that when restricted to planar graphs admit *subexponential parameterized algorithms*, that is, algorithms with running time of the form $2^{o(k)} \cdot n^{\mathcal{O}(1)}$. This was first observed in 2000 by Alber et al. [1], who obtained an algorithm for deciding whether a given n-vertex planar graph contains a dominating set of size k in time $2^{\mathcal{O}(\sqrt{k})}n^{\mathcal{O}(1)}$. It turned out that the phenomenon is much more general. A robust framework explaining why various problems like Feedback Vertex Set, Vertex Cover, Dominating Set, or Longest Path admit subexponential parameterized algorithms on planar graphs, usually with running times of the form $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ or $2^{\mathcal{O}(\sqrt{k}\log k)} \cdot n^{\mathcal{O}(1)}$, is provided by the bidimensionality theory of Demaine et al. [12]. Roughly speaking, the idea is to exploit the relation between the treewidth of a planar graph and the size of a grid minor that can be found in it. More precisely, the following win/win approach is implemented. Either the treewidth of the graph is $\mathcal{O}(\sqrt{k})$ and the problem can be solved using dynamic programming on a tree decomposition, or a $c\sqrt{k} \times c\sqrt{k}$ grid minor can be found, for some large constant c, which immediately implies that we are working with a yes- or with a no-instance of the problem. Furthermore, it turns out that for a large majority of problems the running time yielded by bidimensionality is essentially optimal under ETH: no $2^{o(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$-time algorithm can be expected. We refer the reader to the survey [14], as well as the textbook [10, Chapter 7] for an overview of bidimensionality and its applications.

While the requirement that the problem can be solved efficiently on bounded treewidth graphs is usually not restrictive, the assumption that uncovering *any* large grid minor provides a meaningful insight into the instance considerably limits the applicability of the bidimensionality methodology. Therefore, while bidimensionality can be extended to more general classes, like excluding some fixed graph as a minor [12, 15], map graphs [11], or unit disk graphs [25], there are many problems that are "almost" bidimensional, and yet their parameterized complexity remained open for years.

One example where such a situation occurs is the Directed Longest Path problem. While the existence of a $\sqrt{k} \times \sqrt{k}$ grid minor in an undirected graph immediately implies the existence of an undirected path on k vertices, the same principle cannot be applied in the directed setting: even if we uncover a large grid minor in the underlying undirected graph, there is no guarantee that a long directed path can be found, because we do not control the orientation of the arcs. Thus, Directed Longest Path can be solved in time $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ on general graphs using color coding [3], but no subexponential parameterized algorithm on planar graphs was known.

On the other hand, a $2^{o(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$-time algorithm on planar graphs cannot be expected under ETH, which leaves a large gap between the known upper and lower bounds. Closing this perplexing gap was mentioned as an open problem in [7, 18, 40, 41]. A similar situation happens for Weighted Longest Path, where we are looking for a k-path of minimum weight in an edge-weighted planar graph; the question about the complexity of this problem was raised in [7, 39]. Another example is k-Cycle: deciding whether a given planar graph contains a cycle of length exactly k. While the property of admitting a cycle of length *at least* k is bidimensional, and therefore admits a $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$-time algorithm on planar graphs, the technique fails for the variant when we ask for length *exactly* k . This question was asked in [7]. We will mention more problems with this behavior later on.

The theme of "subexponential algorithms beyond bidimensionality" has recently been intensively investigated, with various success. For a number of specific problems such algorithms were found; these include planar variants of Steiner Tree parameterized by the size of the tree [35, 36], Subset TSP parameterized by the number of terminals, in both the undirected and the directed variants [28, 32], and Max Leaf Outbranching [18]. On the other hand, recently a subset of the current authors proved in [31] that Steiner Tree on planar graphs, parameterized by the number of terminals, does not admit a subexponential parameterized algorithm unless ETH fails; this contrasts the existence of such an algorithm for the parameterization by the size of the tree. In all the abovementioned positive cases, the algorithms are technically very involved and depend heavily on the combinatorics of the problem at hand.

A more systematic approach is offered by the work of Dorn et al. [18] and Tazari [40, 41], who obtained "almost" subexponential algorithm for Directed Longest Path on planar and, more generally, apex-minor-free graphs. More precisely, they proved that for any $\varepsilon > 0$ there is δ such that the Directed Longest Path problem is solvable in time $\mathcal{O}((1+\varepsilon)^k \cdot n^\delta)$ on planar directed graphs and, more generally, on directed graphs whose underlying undirected graph excludes a fixed apex graph as a minor. This technique can be extended to other problems that can be characterized as searching for a small connected pattern in a large host graph, which suggests that some more robust methodology is hiding just beyond the frontier of our understanding.

Main result. In this paper, we introduce a versatile technique for solving such problems in subexponential parameterized time, by proving the following theorem.

Theorem 1. *Let $\mathcal{C}$ be a class of graphs that exclude a fixed apex graph as a minor. Then there exists a randomized polynomial-time algorithm that, given an n-vertex graph G from $\mathcal{C}$ and an integer k, samples a vertex subset $A \subseteq V(G)$ with the following properties:*

(P1) *The induced subgraph $G[A]$ has treewidth $\mathcal{O}(\sqrt{k}\log k)$.*

(P2) *For every vertex subset $X \subseteq V(G)$ with $|X| \leqslant k$ that induces a connected subgraph of G, the probability that X is covered by A, that is $X \subseteq A$, is at least $(2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot n^{\mathcal{O}(1)})^{-1}$.*

Here, by an apex graph we mean a graph that can be made planar by removing one vertex. Note that Theorem 1 in particular applies to planar graphs, and to graphs embeddable in a fixed surface.

Applications. Similarly as in the case of bidimensionality, Theorem 1 provides a simple recipe for obtaining subexponential parameterized algorithms: check how fast the considered problem can be solved on graphs of bounded treewidth, and then combine the treewidth-based algorithm with Theorem 1. We now show how

Theorem 1 can be used to obtain randomized subexponential parameterized algorithms for a variety of problems on apex-minor-free classes; for these problems, the existence of such algorithms so far was open even for planar graphs. We only list the most interesting examples to showcase possible applications.

Directed and weighted paths and cycles. As mentioned earlier, the question about the existence of subexponential parameterized algorithms for Directed Longest Path and Weighted Longest Path on planar graphs was asked in [7, 18, 40, 41]. Let us observe that on a graph of treewidth[1] t, both Directed Longest Path and Weighted Longest Path, as well as their different combinations, like finding a maximum or minimum weight directed path or cycle on k vertices, are solvable in time $2^{\mathcal{O}(t \log t)} n^{\mathcal{O}(1)}$ by the standard dynamic programming; see, e.g., [10, Chapter 7]. This running time can be improved to single-exponential time $2^{\mathcal{O}(t)} n^{\mathcal{O}(1)}$ [5, 19, 24].

In order to obtain a subexponential parameterized algorithm for, say, Directed Longest Path on planar directed graphs, we do the following. Let G be the given planar directed graph, and let $U(G)$ be its underlying undirected graph. Apply the algorithm of Theorem 1 to $U(G)$, which in polynomial time samples a subset $A \subseteq V(U(G))$ such that $G[A]$ has treewidth at most $\mathcal{O}(\sqrt{k} \log k)$, and the probability that A covers some directed k-path in G, provided it exists, is at least $(2^{\mathcal{O}(\sqrt{k} \log^2 k)} \cdot n^{\mathcal{O}(1)})^{-1}$. Then, we verify whether $G[A]$ admits a directed k-path using standard dynamic programming in time $2^{\mathcal{O}(\sqrt{k} \log k)} \cdot n^{\mathcal{O}(1)}$. Provided some directed k-path exists in the graph, this algorithm will find one with probability at least $(2^{\mathcal{O}(\sqrt{k} \log^2 k)} \cdot n^{\mathcal{O}(1)})^{-1}$. Thus, by making $2^{\mathcal{O}(\sqrt{k} \log^2 k)} \cdot n^{\mathcal{O}(1)}$ independent runs of the algorithm, we can reduce the error probability to at most $1/2$. All in all, the obtained algorithm runs in time $2^{\mathcal{O}(\sqrt{k} \log^2 k)} \cdot n^{\mathcal{O}(1)}$ and can only report false negatives with probability at most $1/2$.

Note that in order to apply the dynamic programming algorithm, we need to construct a suitable tree decomposition of $G[A]$. However, a variety of standard algorithms, e.g., the classic 4-approximation of Robertson and Seymour [38], can be used to construct such an approximate tree decomposition within the same asymptotic running time. Actually, a closer look into the proof of Theorem 1 reveals that the algorithm can construct, within the same running time a tree decomposition of $G[A]$ certifying the upper bound on its treewidth.

Observe that the same approach works also for any apex-minor-free class $\mathcal{C}$ and can be applied also to Weighted Longest Path and k-Cycle. We obtain the following corollary.

Corollary 2. *Let $\mathcal{C}$ be a class of graphs that exclude some fixed apex graph as a minor. Then all the following problems are solvable in randomized time $2^{\mathcal{O}(\sqrt{k} \log^2 k)} \cdot n^{\mathcal{O}(1)}$ on graphs from $\mathcal{C}$:* Weighted Longest Path, k-Cycle, *and* Directed Longest Path. *In case of* Directed Longest Path, *we mean that the underlying undirected graph of the input graph belongs to $\mathcal{C}$.*

Note here that the approach presented above works in the same way for various combinations and extensions of problems in Corollary 2, like weighted, colored, or directed variants, possibly with some constraints on in- and out-degrees, etc. In essence, the only properties that we need is that the sought pattern persists in the subgraph induced by the covering set A and that it can be efficiently found using

[1] For Directed Longest Path we speak about the treewidth of the underlying undirected graph.

dynamic programming on a tree decomposition. To give one more concrete example, Sau and Thilikos in [39] studied the problem of finding a connected k-edge subgraph with all vertices of degree at most some integer Δ; for $\Delta = 2$ this corresponds to finding a k-path or a k-cycle. For fixed Δ they gave a subexponential algorithm on (unweighted) graphs excluding some fixed graph as a minor and asked if the weighted version of this problem can be solved in subexponential parameterized time. Theorem 1 immediately implies that for fixed Δ the weighted variant of the problem is solvable in randomized time $2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot n^{\mathcal{O}(1)}$ on apex-minor-free graphs.

Subgraph Isomorphism. SUBGRAPH ISOMORPHISM is a fundamental problem, where we are given two graphs: an n-vertex *host* graph G and a k-vertex *pattern* graph P. The task is to decide whether P is isomorphic to a subgraph of G. Eppstein [20] gave an algorithm solving SUBGRAPH ISOMORPHISM on planar graphs in time $k^{\mathcal{O}(k)}n$, which was subsequently improved by Dorn [17] to $2^{\mathcal{O}(k)}n$. The first implication of our main result for SUBGRAPH ISOMORPHISM concerns the case when the maximum degree of P is bounded by a constant. Matoušek and Thomas [33] proved that if a tree decomposition of the host graph G of width t is given, and the pattern graph P is connected and of maximum degree at most some constant Δ, then deciding whether P is isomorphic to a subgraph of G can be done in time $\mathcal{O}(k^{t+1}n)$. By combining this with Theorem 1 as before, we obtain the following.

COROLLARY 3. *Let $\mathcal{C}$ be a class of graphs that exclude some fixed apex graph as a minor, and let Δ be a fixed constant. Then, given a connected graph P with at most k vertices and maximum degree not exceeding Δ, and a graph $G \in \mathcal{C}$ on n vertices, it is possible to decide whether P is isomorphic to a subgraph of G in randomized time $2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot n^{\mathcal{O}(1)}$.*

In a very recent work, Bodlaender, Nederlof, and van der Zanden [6] proved that SUBGRAPH ISOMORPHISM on planar graphs cannot be solved in time $2^{o(n/\log n)}$ unless ETH fails. The lower bound of Bodlaender et al. holds for two very special cases. The first case is when the pattern graph P is a tree and has only one vertex of super-constant degree. The second case is when P is not connected, but its maximum degree is a constant. Thus, the results of Bodlaender et al. show that both the connectivity and the bounded degree constraints on pattern P in Corollary 3 are necessary to keep the square root dependence on k in the exponent. However, a possibility of solving SUBGRAPH ISOMORPHISM in time $2^{\mathcal{O}(k/\log k)} \cdot n^{\mathcal{O}(1)}$, which is still parameterized subexponential, is not ruled out by the work of Bodlaender et al. Interestingly enough, Bodlaender, Nederlof, and van der Zanden [6] also give a matching dynamic programming algorithm that can be combined with our theorem.

THEOREM 4 (Theorem 7 of [6]). *Let H be a fixed graph, and let us fix any $\varepsilon > 0$. Given a pattern graph P on at most k vertices and an H-minor-free host graph G of treewidth at most $\mathcal{O}(k^{1-\varepsilon})$, it is possible to decide whether P is isomorphic to a subgraph of G in time $2^{\mathcal{O}(k/\log k)} \cdot n^{\mathcal{O}(1)}$.*

By combining Theorem 4 with Theorem 1 in the same way as before, we obtain the following.

COROLLARY 5. *Let $\mathcal{C}$ be a class of graphs that exclude some fixed apex graph as a minor. Then, given a connected graph P with at most k vertices, and a graph $G \in \mathcal{C}$ on n vertices, it is possible to decide whether P is isomorphic to a subgraph of G in randomized time $2^{\mathcal{O}(k/\log k)} \cdot n^{\mathcal{O}(1)}$.*

Let us stress here that the lower bounds of Bodlaender, Nederlof, and van der Zanden [6] show that the running time given by Corollary 5 is tight: no $2^{o(k/\log k)} \cdot n^{\mathcal{O}(1)}$-time algorithm can be expected under ETH.

Local search. Fellows et al. [23] studied the following parameterized local search problem on apex-minor-free graphs. In the LS Vertex Cover problem we are given an n-vertex graph G, a vertex cover S in G, and an integer k. The task is to decide whether G contains a vertex cover S', such that $|S'| < |S|$ and the Hamming distance $|S \triangle S'|$ between sets S and S' is at most k. In other words, for a given vertex cover, we ask if there is a smaller vertex cover which is k-close to the given one in terms of Hamming (edit) distance. Fellows et al. [23] gave an algorithm solving LS Vertex Cover in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ on planar graphs. The question whether this can be improved to subexponential parameterized time was raised in [13, 23].

The crux of the approach of Fellows et al. [23] is the following observation. If there is a solution to LS Vertex Cover, then there is a solution S', such that $S \triangle S'$ induces a connected subgraph in G. Since $S \triangle S'$ contains at most k vertices and is connected, our Theorem 1 can be used to sample a vertex subset A that induces a subgraph of treewidth $\mathcal{O}(\sqrt{k} \log k)$ and covers $S \triangle S'$ with high probability. Thus, by applying the same principle of independent repetition of the algorithm, we basically have to search for suitable sets $S \setminus S'$ and $S' \setminus S$ in the subgraph of G induced by A. We should, however, be careful here: there can be edges between A and its complement, and these edges also need to be covered by S', so we cannot just restrict our attention to $G[A]$. To handle this, we apply the following preprocessing. For every vertex $v \in A$, if v is adjacent to some vertex outside of A that is not included in S, then v must be in S and needs also to remain in S'. Hence, we delete all such vertices from $G[A]$, and it is easy to see that now the problem boils down to looking for feasible $S \setminus S'$ and $S' \setminus S$ within the obtained induced subgraph. This can be easily done in time $2^{\mathcal{O}(t)} \cdot n^{\mathcal{O}(1)}$, where $t \leqslant \mathcal{O}(\sqrt{k} \log k)$ is the treewidth of this subgraph; hence we obtain the following:

Corollary 6. *Let $\mathcal{C}$ be a class of graphs that exclude some fixed apex graph as a minor. Then* LS Vertex Cover *on graphs from $\mathcal{C}$ can be solved in randomized time* $2^{\mathcal{O}(\sqrt{k} \log^2 k)} \cdot n^{\mathcal{O}(1)}$.

Steiner tree. Steiner Tree is a fundamental network design problem: for a graph G with a prescribed set of terminal vertices S and an integer k, we ask whether there is a tree on at most k edges that spans all terminal vertices. Pilipczuk et al. [35] gave an algorithm for this problem with running time $2^{\mathcal{O}((k \log k)^{2/3})} \cdot n$ on planar graphs and on graphs of bounded genus. With much more additional work, the running time was improved to $2^{\mathcal{O}(\sqrt{k \log k})} \cdot n$ in [36].

Again, by combining the standard dynamic programming solving Steiner Tree on graphs of treewidth t in time $2^{\mathcal{O}(t \log t)} n^{\mathcal{O}(1)}$ (see e.g. [9]) with Theorem 1, we immediately obtain the following.

Corollary 7. *Let $\mathcal{C}$ be a class of graphs that exclude some fixed apex graph as a minor. Then* Steiner Tree *on graphs from $\mathcal{C}$ can be solved in randomized time* $2^{\mathcal{O}(\sqrt{k} \log^2 k)} \cdot n^{\mathcal{O}(1)}$.

Contrary to the much more involved algorithm of Pilipczuk et al. [36], the algorithm above can equally easily handle various variants of the problem. For instance, we can look for a Steiner tree on k edges that minimizes the total weight of the edges, or we can ask for a Steiner out-branching in a directed graph, or we can put additional constraints on vertex degrees in the tree, and so on.

Outline. In section 2 we give an informal overview of the proof of Theorem 1 for the case of planar graphs. We try to focus on main intuitions and concepts, rather than to describe technical details necessary for a formal reasoning. Then, in sections 3 and 4 we recall the standard concepts and introduce auxiliary technical results. The full proof of Theorem 1 is contained in section 5. In section 6 we explain how to generalize the proof of Theorem 1 to the cases when the pattern has multiple connected components and when we consider an arbitrary proper minor-closed class of graphs. We conclude in section 7 by listing open problems raised by our work.

2. Overview of the proof of Theorem 1. We now give an informal overview of the proof of Theorem 1 in the case of planar graphs. In fact, the only two properties of planar graphs that are essential to the proof are (a) planar graphs are minor-closed, and (b) they have locally bounded treewidth by a linear function; that is, there exists a constant $\bar{\alpha}(\mathcal{C})$ such that every planar graph of radius k has treewidth at most $\bar{\alpha}(\mathcal{C}) \cdot k$. In fact, for planar graphs one can take $\bar{\alpha}(\mathcal{C}) = 3$ [37], and as shown in [16], the graph classes satisfying both (a) and (b) are exactly graph classes excluding a fixed apex graph as a minor. However, in a planar graph we can rely on some topological intuition, making the presentation more intuitive. In the description we assume familiarity with tree decompositions; see section 3 for a formal definition.

Locally bounded treewidth of planar graphs. As a warm-up, let us revisit a proof that planar graphs have locally bounded treewidth. The considered proof yields a worse constant than $\bar{\alpha}(\mathcal{C}) = 3$, but one of the main ideas—to find a separator in a planar graph by finding many disjoint paths, present already in [2]—it is insightful for our argumentation. Let G_0 be a graph of radius k; that is, there exists a root vertex r_0 such that every vertex of G_0 is within distance at most k from r_0.

As with most proofs showing that a graph in question has bounded treewidth, we will recursively construct a tree decomposition of bounded width. To this end, we need to carefully define the state of the recursion. We do it as follows: the recursive step aims at decomposing a subgraph G of the input graph G_0, with some chosen set of terminals $T \subseteq V(G)$ on the outer face of G. The terminals T represent connections between G and the rest of G_0. In order to be able to glue back the obtained decompositions from the recursive step, our goal is to provide a tree decomposition of G with T contained in the root bag of the decomposition, so that later we can connect this bag to decompositions of other pieces of the graph that also contain the vertices of T. During the process, we keep the invariant that $|T| \leqslant 8(k+2)$, allowing us to bound the width of the decomposition. Furthermore, the assumption that G_0 is of bounded radius projects onto the recursive subinstances by the following invariant: every vertex of G is within distance at most k from some terminal.

In the recursive step, if $T = V(G)$, $|T| < 8(k+2)$, or G is not connected, then it is easy to proceed: in the first case we may produce a single bag consisting of the whole vertex set, in the second case we may include an arbitrary nonterminal to T, and in the third case we may treat each connected component separately. The interesting case is when none of these corner cases happen, in particular $|T| = 8(k+2)$.

We partition T along the outer face into four parts of size $2(k+2)$ each, called north, east, south, and west terminals. We compute minimum separators (i.e., vertex cuts) between the north and the south terminals, and between the east and the west terminals. If, in any of these directions, a cut W of size strictly smaller than $2(k+2)$ is found, then we can make a divide-and-conquer step: for every connected component D of $G-(T \cup W)$ we recurse on the graph $G[N[D]]$ with terminals $N(D)$, obtaining a

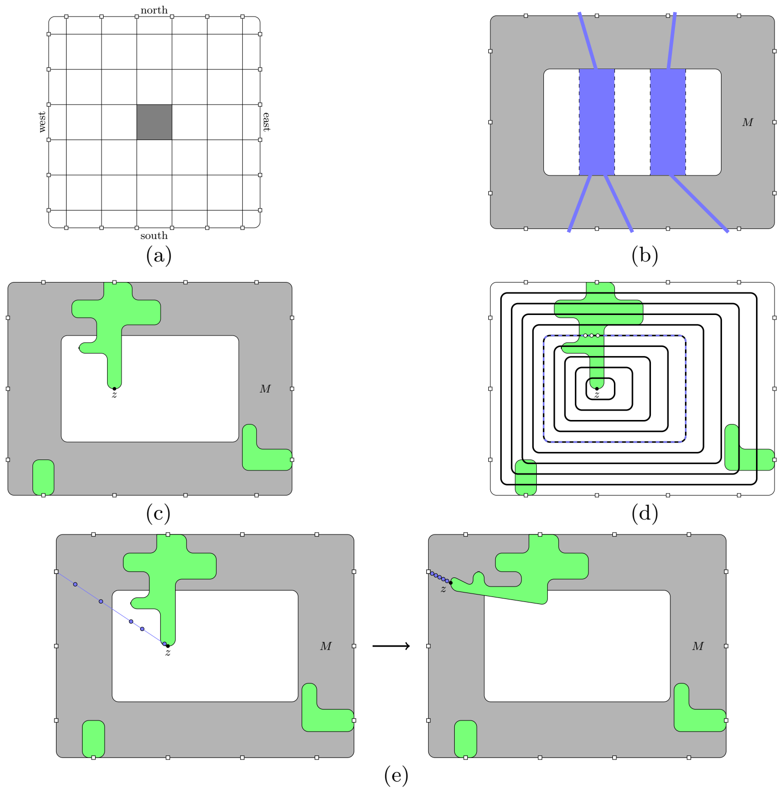

FIG. 1. *Illustrations for section* 2. (a) *The proof that planarity implies locally bounded treewidth. The vertices in the gray area are too far from the terminals.* (b) *Partitioning using a separator consisting of a few vertices of the margin and a few islands. The margin is gray and the islands are separated by dashed lines. The blue separator consists of $\widetilde{\mathcal{O}}(\sqrt{k})$ islands and vertices of the margin. If the blue islands are disjoint from the solution, we delete them and obtain a balanced separator of size $\widetilde{\mathcal{O}}(\sqrt{k})$.* (c) *The situation if the partitioning strategy* (b) *cannot be applied, because the islands in the separator intersect the pattern. We have a component of the pattern (green) stretched between a light terminal and a vertex z outside of the margin.* (d) *Chain of z-T^{li} separators: a sparse separator that partitions the pattern in a balanced fashion is highlighted.* (e) *Contraction of a path P_i (blue) onto its public vertices (blue circles). A significant number of the vertices of the pattern become much closer to the light terminals.*

tree decomposition $\mathcal{T}_D$. Finally, we attach all the obtained tree decompositions below a fresh root node with bag $T \cup W$, which is of size less than $10(k+2)$.

The crux is that such a separator W is always present in the graph. Indeed, otherwise there would exist $2(k+2)$ disjoint paths between the north and the south terminals and $2(k+2)$ disjoint paths between the east and the west terminals. Consider the region bounded by the two middle north-south and the two middle east-west paths: the vertices contained in this region are within distance larger than k from the outer face, on which all terminals lie (see Figure 1(a)). This contradicts our invariant.

Our recursion. In our case, we use a similar, but much more involved recursion scheme. In the recursive step, we are given a *minor* G of the input graph G_0, with some *light* terminals $T^{\mathsf{li}} \subseteq V(G)$ on the outer face and some *heavy* terminals $T^{\mathsf{he}} \subseteq V(G) \setminus T^{\mathsf{li}}$ lying anywhere in the graph. As before, the terminals represent connections to the other parts of the graph. The names *light* and *heavy* correspond to the amount of potential these terminals bear in the final amortized analysis that is used to bound the error probability; this will become clear later on. We require that the terminals $T := T^{\mathsf{li}} \cup T^{\mathsf{he}}$ have to be contained in the root bag of the tree decomposition that is going to be constructed in this recursion step. Moreover, we maintain an invariant that $|T| = \widetilde{\mathcal{O}}(\sqrt{k})$, in order to bound the width of the decomposition. The graph G is a minor of the input graph G_0, since we often prefer to contract some edges instead of deleting them; thus we maintain some distance properties of G.

In our recursion, the light terminals originate from cutting the graph in a similar fashion as in the proof that planar graphs have locally bounded treewidth, presented above. Hence, we keep the invariant that light terminals lie on the outer face. We sometimes need to cut deeply inside G. The produced terminals are heavy, but every such step corresponds to a significant progress in detecting the pattern, and hence such steps will be rare. In every such step, we artificially provide connectivity of the subinstances through the heavy terminals; this is technical and omitted in this overview.

Recall that our goal is to preserve a connected k-vertex pattern from the input graph. Here, the pattern can become disconnected by recursing on subsequent separations, but such cuttings will always be along light terminals. Therefore, we define a *pattern* in a subinstance $(G, T^{\mathsf{li}}, T^{\mathsf{he}})$ solved in the recursion as a set $X \subseteq V(G)$ of size at most k such that every connected component of $G[X]$ contains a light terminal. Hence, compared to the presented proof of planarity implying locally bounded treewidth, we aim at more restricted width of the decomposition, namely $\widetilde{\mathcal{O}}(\sqrt{k})$, but we can contract or delete parts of G, as long as the probability of spoiling a fixed, but unknown k-vertex pattern X remains inverse subexponential in k.

Clustering. Upon deleting a vertex or an edge, some distance properties that we rely on can be broken. We need the following sampling procedure that partitions the graph into connected components of bounded radii, such that the probability of spoiling a particular pattern is small. The proof of the following theorem is similar to the metric decomposition tool of [29] and to the recursive decomposition used in the construction of Bartal's HSTs [4].

THEOREM 8. *There exists a randomized polynomial-time algorithm that, given a graph G on $n > 1$ vertices and a positive integer k, returns a vertex subset $B \subseteq V(G)$ with the following properties:*

(a) *The radius of each connected component of $G[B]$ is less than $9k^2 \lg n$.*
(b) *For each vertex subset $X \subseteq V(G)$ of size at most k, the probability that $X \subseteq B$ is at least $1 - \frac{1}{k}$.*

Proof sketch. Start with $H := G$ and iteratively, as long as $V(H) \neq \emptyset$, perform the following procedure. Pick arbitrary $v \in V(H)$, and choose a radius r as follows. Start with $r = 1$ and iteratively, given current radius r, with probability $p := (2k^2)^{-1}$ accept r, and with probability $1 - p$ increase it by one and continue (i.e., choose r according to the geometric distribution with success probability p). Given an accepted radius r, put all vertices within distance *less* than r from v into B, and delete from H all vertices within distance *at most* r from v.

Since the procedure performs at most n steps, by the union bound the probability of some radius exceeding $9k^2 \lg n$ is at most $\frac{1}{2k}$. Fix a vertex $x \in V(G)$. We have $x \notin B$ only if at some point $x \in V(H)$ and the distance between v and x in H is exactly r when the radius r gets accepted. However, in this case, if the radius r is increased, x is put into B regardless of subsequent random choices. Consequently, for a fixed vertex $x \in V(G)$, the probability that $x \notin B$ is at most p. By the union bound, the probability that $X \not\subseteq B$ is at most $|X|p$, which is at most $\frac{1}{2k}$ for $|X| \leqslant k$. ∎

Splitting along a separator. We would like to apply a similar divide-and-conquer step as in the presented proof that planar graphs have locally bounded treewidth. The problem is that we can only afford a separator W of size $\widetilde{\mathcal{O}}(\sqrt{k})$; however, the radius of the graph can be much larger.

Let us define the *margin* M to be the set of vertices within distance at most $2000\sqrt{k}\lg k = \widetilde{\mathcal{O}}(\sqrt{k})$ from any light terminal. Intuitively, our case should be easy if every vertex of the pattern X is in the margin: we could then just throw away all vertices of $G - M$ and use the fact that planar graphs have locally bounded treewidth, as the light terminals lie on the outer face (precisely, we may add a special vertex embedded in the outer face, connect it to the light terminals, and apply the argument about locally bounded treewidth to this vertex; this is how we do it in the formal reasoning). However, we cannot just branch (guess) whether this is the case: the information that $G - M$ contains a vertex of the pattern is not directly useful.

Instead, we make a localized analogue of this guess: we identify a relatively compact set of vertices of $G - M$ that prohibit us from making a single step of the recursion sketched above. First, we apply the clustering procedure (Theorem 8) to the graph $G - M$, so that we can assume that every connected component of $G - M$, henceforth called *an island*, is of radius bounded polynomially in k and $\lg n$. Second, we construct an auxiliary graph H by contracting every island C into a single vertex u_C. Note that now in H every vertex is within distance at most $2000\sqrt{k}\lg k + 1 = \widetilde{\mathcal{O}}(\sqrt{k})$ from a light terminal. Thus H has treewidth $\widetilde{\mathcal{O}}(\sqrt{k})$. By standard arguments, we can find a balanced separator W_H in H, that is, a separator of size $\widetilde{\mathcal{O}}(\sqrt{k})$ such that every connected component of $H - W_H$, after lifting it back to G by reversing contractions, contains (a) at most $|T|/2$ terminals from G and (b) at most $|V(G) \setminus T|/2$ nonterminal vertices of G.

The separator W_H can be similarly lifted to a separator W in G that corresponds to $\widetilde{\mathcal{O}}(\sqrt{k})$ vertices of M and $\widetilde{\mathcal{O}}(\sqrt{k})$ islands. Now it is useful to make a guess if some vertex of an island in W (i.e., a vertex of $W \setminus M$) belongs to the solution. If this is not the case, we can delete the whole $W \setminus M$ from the graph and apply the procedure recursively to connected components of $G - W$. If this is the case, we know that there is a vertex of the solution outside of M, which moreover lies within one of $\widetilde{\mathcal{O}}(\sqrt{k})$ components of radius polynomial in k and $\lg n$; we will later localize such a vertex more closely, as its knowledge will be pivotal for our further analysis. Therefore, with some probability q we decide to assume that $W \setminus M$ contains a vertex of the pattern, and with the remaining probability $1 - q$ we decide that this is not the case. In the latter case, we remove $W \setminus M$ from the graph and recurse using $W \cap M$, which has size $\widetilde{\mathcal{O}}(\sqrt{k})$, as a separator; see Figure 1(b). The fact that every connected component of $G - W$ contains at most $|T|/2$ terminals allows us to keep the invariant that $|T| = \widetilde{\mathcal{O}}(\sqrt{k})$.

Let us now analyze what probability q we can afford. Observe that in every subinstance solved recursively the number of nonterminal vertices is halved. Thus, every vertex x of the pattern X is contained in G only in $\mathcal{O}(\lg n)$ subinstances in the whole recursion tree; here we exclude the subinstances where x is a light terminal, because then its treatment is determined by the output specification of the recursive procedure. Consequently, we care about correct choices only in $\mathcal{O}(k \lg n)$ steps of the recursion. In these steps, we do not want to make a mistake during the clustering procedure ($1/k$ failure probability) and we want to correctly guess that $W \setminus M$ is disjoint with the pattern, provided this is actually the case (q failure probability). Thus, if we put $q = 1/k$, then the probability that we succeed in all $\mathcal{O}(k \lg n)$ steps we care about is inverse-polynomial in n; this is sufficient for our needs.

Island with a vertex of the pattern. We are left with the second case, where some island $C \subseteq W$ intersects the pattern. We have $q = 1/k$ probability of guessing correctly that this is the case, and independently we have $(1 - 1/k)$ probability of not making a mistake in the clustering step.

The bound on the radii of the islands, as well as the fact that only $\widetilde{\mathcal{O}}(\sqrt{k})$ islands are contained in W, allow us to localize this vertex of the pattern even closer. Recall that the radius of each island is bounded by $9k^2 \lg n$. For the rest of this overview we assume that $\lg n$ is bounded polynomially in k, and hence the radius of each island is polynomial in k. Intuitively, this is because if, say, we had $\lg n > 100 \cdot k^{100}$, then $n > 2^{100k^{100}}$ and having allowed factor $2^{100k^{100}}$ in the running time bound we may apply a variety of other algorithmic techniques. More formally, we observe that $(\lg n)^{\widetilde{\mathcal{O}}(\sqrt{k})}$ is bounded by $2^{\widetilde{\mathcal{O}}(\sqrt{k})} \cdot n^{o(1)}$, which is sufficient to make sure that all the experiments whose success probability depend on $\lg n$ succeed simultaneously with probability at least $(2^{\widetilde{\mathcal{O}}(\sqrt{k})} \cdot n^{o(1)})^{-1}$.

We first guess (by sampling at random) an island $C \subseteq W$ that contains a vertex of the pattern. Then, we pick an arbitrary vertex $z \in C$ and guess (by sampling at random) the distance d in C between z and the closest vertex of the pattern in C. By contracting all vertices within distance less than d from z, with success probability inverse-polynomial in k, we arrive at the following situation: (see Figure 1(c))

$$\text{we have a vertex } z \notin M \text{ such that}$$
$$\text{either } z \text{ or a neighbor of } z \text{ belongs to the pattern } X.$$

Chain of separators. Hence, one of the components of $G[X]$ is stretched across the margin M, between a light terminal on the outer face and the vertex z inside the margin. Our idea now is to use this information to cut X in a balanced fashion. Note that we have already introduced an inverse-polynomial in k multiplicative factor in the success probability. Hence, to maintain the overall inverse-subexponential dependency on k in the success probability, we should aim at a progress that will allow us to bound the number of such steps by $\widetilde{\mathcal{O}}(\sqrt{k})$.

Unfortunately, it is not obvious how to find such a separation. It is naive to hope for a z-T^{li} separator of size $\widetilde{\mathcal{O}}(\sqrt{k})$, and a larger separator seems useless, if there is only one. However, we can aim at a Baker-style argument: if we find a chain of p pairwise disjoint z-T^{li} separators $C_1, C_2, \ldots, C_p$ (see Figure 1(d)), each of size polynomial in k, then we may guess a "sparse one" and separate along it. Since the separators are pairwise disjoint, there exists a "sparse" separator C_i containing at most k/p vertices of the pattern X. On the other hand, since the pattern contains a component stretched from z to a light terminal, every C_i intersects X. If we ignore the first and the last $p/4$ separators, there is a sparse separator in between containing at most $2k/p$ vertices of X. We can then guess the at most $2k/p$ vertices of X in this separator and break the instance into two along it; in each of the resulting instances, at least $p/4$ vertices of X remain. Assuming separators $C_1, C_2, \ldots, C_p$ are of size bounded polynomially in k, the optimal choice is $p \sim k^{2/3}$, which leads to success probability inverse in $2^{\widetilde{\mathcal{O}}(k^{2/3})}$.

However, we can apply a bit smarter counting argument. Take $p = 120\sqrt{k} \lg k$. Look at $C_{p/2}$ and assume that at most half of the vertices of X lie on the side of $C_{p/2}$

with separators C_i for $i < p/2$; the other case is symmetric. The crucial observation is the following: there exists an index $i \leqslant p/2$ such that if $|C_i \cap X| = \alpha$, then C_i partitions X into two parts of size at least $\alpha\sqrt{k}/10$ each. Indeed, otherwise we have that for every $i \leqslant p/2$ it holds that

$$|X \cap C_i| \geqslant \frac{10}{\sqrt{k}} \cdot \sum_{j<i} |X \cap C_j|.$$

This implies $|X \cap \bigcup_{j \leqslant i} C_j| \geqslant (1 + 10/\sqrt{k})^i$, and $|X \cap \bigcup_{j \leqslant p/2} C_j| > k$ for $p = 120\sqrt{k} \lg k$.

Hence, we guess (by sampling at random) such an index i, the value of $\alpha = |X \cap C_i|$, and the set $X \cap C_i$. If the size of C_i is bounded polynomially in k, with success probability $k^{-\mathcal{O}(\alpha)}$ we partition the pattern into two parts of size at least $\alpha\sqrt{k}/10$ each. A simple amortization argument shows that all these guessings incur only the promised $2^{-\widetilde{\mathcal{O}}(\sqrt{k})}$ multiplicative factor in the overall success probability. Furthermore, as such a step creates α heavy terminals, it can be easily seen that the total number of heavy terminals will never grow beyond $\widetilde{\mathcal{O}}(\sqrt{k})$.

However, the above argumentation assumes we are given such a chain of separators C_i: they are not only pairwise disjoint, but also of size polynomial in k. Let us now inspect how to find them.

Duality. In the warm-up proof of planar graphs having locally bounded treewidth, the separator W is obtained from the classic Menger maximum flow/ minimum cut duality. Here, we aim at a chain of disjoint separators, but we require that their sizes are polynomial in k. It turns out that we can find such a chain by formulating a maximum flow of minimum cost problem and extracting the separator chain in question from the optimum solution to its (LP) dual. Thus we obtain the following result, which can be considered a variant of the classic max-flow-min-cut duality, where we allow paths to be only "almost" disjoint, but in the dual we extract many disjoint small separators.

THEOREM 9. *There is a polynomial-time algorithm that given a connected graph G, a pair $s, t \in V(G)$ of different vertices, and positive integers p, q, outputs one of the following structures in G:*

(a) *A chain $(C_1, \ldots, C_p)$ of (s,t)-separators with $|C_j| \leqslant 2q$ for each $j \in [p]$.*
(b) *A sequence $(P_1, \ldots, P_q)$ of (s,t)-paths with $|(V(P_i) \cap \bigcup_{i' \neq i} V(P_{i'})) \setminus \{s,t\}| \leqslant 4p$ for each $i \in [q]$.*

Proof sketch. We formulate the second outcome as a maximum flow of minimum cost problem in an auxiliary graph, where every vertex $v \in V(G) \setminus \{s,t\}$ is duplicated into two copies: one of cost 0 and capacity 1, and one of cost 1 and infinite capacity. We ask for a minimum-cost flow of size $2q$ from s to t. If the cost of such flow is at most $2pq$, the projection onto G of the cheapest q flow paths gives the second output. Otherwise, we read the desired chain of separators $(C_1, \ldots, C_p)$ as distance layers from s in the graph with distances imposed by the solution to the dual linear program. ∎

Since all light terminals lie on the outer face, we can attach an auxiliary root vertex r_0 adjacent to all light terminals and apply Theorem 9 to $(s,t) := (r_0, z)$, $p := 120\sqrt{k} \lg k$, and $q := \text{poly}(k)$. If the algorithm of Theorem 9 returns a chain of separators, we proceed as described before. Thus, we are left with the second output: $q = \text{poly}(k)$ nearly-disjoint paths from T^{li} to z.

Nearly disjoint paths. The vertex set of every path P_i can be partitioned into *public* vertices $\mathsf{Pub}(P_i)$, the ones used also by other paths, and the remaining *private* vertices $\mathsf{Prv}(P_i)$. We have $|\mathsf{Pub}(P_i)| \leqslant 4p = 480\sqrt{k}\lg k$, and the sets $\mathsf{Prv}(P_i)$ are pairwise disjoint. We can assume $q > k$, so there exists a path P_i such that $\mathsf{Prv}(P_i)$ is disjoint with the pattern X. By incurring an additional $1/k$ multiplicative factor in the success probability, we can guess, by sampling at random, such index i.

How can we use such a path P_i? Clearly, we can delete the private vertices of P_i, because they can be assumed not to be used by the pattern X. However, we choose a different way: we contract them onto neighboring public vertices along P_i, reducing P_i to a path with vertex set $\mathsf{Pub}(P_i)$. Observe that by this operation the vertex z changes its location in G: from a vertex deeply inside G; namely, not within the margin M, it is moved to a place within distance $|\mathsf{Pub}(P_i)| \leqslant 480\sqrt{k}\lg k$ from the light terminals, which is less than a quarter of the width of the margin; see Figure 1(e). This provides crucial progress for the algorithm, as explained next.

By the connectivity assumptions on the pattern X, the vertex z drags along a number of vertices of X that are close to it. More precisely, if Q is a path in $G[X]$ connecting z or a neighbor of z with a light terminal, then the first $500\sqrt{k}\lg k$ vertices on Q are moved from being within distance at least $1500\sqrt{k}\lg k$ from all light terminals to being within distance at most $1000\sqrt{k}\lg k$ from some light terminal. Hence, if we define that a vertex $x \in X$ is *far* if it is within distance larger than $1000\sqrt{k}\lg k$ (i.e., half of the width of the margin) from all light terminals, and *close* otherwise, then by contracting the private vertices of P_i as described above, at least $500\sqrt{k}\lg k$ vertices of k change their status from far to close.

By a careful implementation of all separation steps, we can ensure that no close vertex of X becomes far again. Consequently, we ensure that the above step can happen only $\widetilde{\mathcal{O}}(\sqrt{k})$ times. Since the probability of succeeding in all guessings within this step is inverse-polynomial in k, this incurs only a $2^{-\widetilde{\mathcal{O}}(\sqrt{k})}$ multiplicative factor in the overall success probability.

This finishes the overview of the proof of Theorem 1. We invite the reader to the next sections for a fully formal proof, which is moreover conducted for an arbitrary apex-minor-free class.

3. Preliminaries.

In this section we introduce notation and recall well-known concepts underlying our work.

Notation. We use standard graph notation; see, e.g., [10] for a reference. All graphs considered in this paper are undirected and simple (without loops or multiple edges), unless explicitly stated. For a vertex u of a graph G, by $N_G(u) := \{v : uv \in E(G)\}$ and $N_G[u] := \{u\} \cup N_G(u)$ we denote the open and closed neighborhoods of u, respectively. Similarly, for a vertex subset $X \subseteq V(G)$, by $N_G[X] := \bigcup_{u \in X} N_G[u]$ and $N_G(X) := N_G[X] \setminus X$ we denote the closed and open neighborhoods of X, respectively. The subscript is dropped whenever it is clear from the context. For a graph G, $\mathsf{cc}(G)$ denotes the set of connected components of G.

For an undirected graph G and an edge $uv \in E(G)$, by *contracting* uv we mean the following operation: remove u and v from the graph, and replace them with a new vertex that is adjacent to exactly those vertices that were neighbors of u or v in G. Note that this definition preserves the simplicity of the graph. By *contracting* v *onto* u we mean the operation of contracting the edge uv and renaming the obtained vertex as u. More generally, if X is a subset of vertices with $G[X]$ being connected, and $u \notin X$ is such that u has a neighbor in X, then by *contracting* X *onto* u we mean the operation of exhaustively contracting a neighbor of u in X onto u up to the point when X becomes empty. Note that due to the connectivity of $G[X]$ such an outcome will always be achieved.

We say that H is a *minor* of G if H can be obtained from G by means of vertex deletions, edge deletions, and edge contractions. An *apex graph* is a graph that can be made planar by removing one of its vertices.

For a positive integer k, we denote $[k] := \{1, \ldots, k\}$. We denote $\lg x := \log_2 x$. Notation log is used only under the $\mathcal{O}(\cdot)$-notation, where multiplicative constants are hidden anyway. We also denote $\exp[t] := e^t$, where e is the base of the natural logarithm.

Tree decompositions. Let G be an undirected graph. A *tree decomposition* $\mathcal{T}$ of G is a rooted tree T with a *bag* $\beta(x) \subseteq V(G)$ associated with every node x, which satisfies the following conditions:

(T1) For each $u \in V(G)$ there is some $x \in V(T)$ with $u \in \beta(x)$.
(T2) For each $uv \in E(G)$ there is some $x \in V(T)$ with $\{u, v\} \subseteq \beta(x)$.
(T3) For each $u \in V(G)$ the node subset $\{x \in V(T) \colon u \in \beta(x)\}$ induces a connected subtree of T.

The *width* of a tree decomposition $\mathcal{T}$ is $\max_{x \in V(T)} |\beta(x)| - 1$, and the *treewidth* of G is equal to the minimum possible width of a tree decomposition of G. We assume the reader's familiarity with basic combinatorics of tree decompositions, and hence we often omit a formal verification that some constructed object is indeed a tree decomposition of some graph.

Balanced separator. Let G be a graph, and let $\mathbf{w} \colon V(G) \to \mathbb{R}_{\geqslant 0}$ be a nonnegative weight function on vertices of G. A $\frac{1}{2}$-*balanced separator* of $\mathbf{w}$ in G is any subset X of vertices of G such that for every connected component C of $G - X$ it holds that

$$\mathbf{w}(V(C)) \leqslant \mathbf{w}(V(G))/2,$$

where we denote $\mathbf{w}(A) := \sum_{u \in A} \mathbf{w}(u)$ for a vertex subset A. The following fact about the existence of balanced separators in graphs of bounded treewidth is well-known; see, e.g., [10, Lemma 7.19].

LEMMA 10. *For any graph G, tree decomposition $\mathcal{T}$ of G, and a weight function $\mathbf{w} \colon V(G) \to \mathbb{R}_{\geqslant 0}$, one of the bags of $\mathcal{T}$ is a $\frac{1}{2}$-balanced separator for $\mathbf{w}$.*

Note that in Lemma 10, if the given decomposition $\mathcal{T}$ has width t, then the obtained $\frac{1}{2}$-balanced separator has size at most $t + 1$. Also, given $\mathcal{T}$, any its bag satisfying the asserted properties can be found in polynomial time.

Locally bounded treewidth and apex-minor-freeness. For the whole proof we fix a class $\mathcal{C}$ of graphs that satisfies the following properties:

(C1) $\mathcal{C}$ is closed under taking minors.
(C2) $\mathcal{C}$ has locally bounded treewidth. That is, there exists a function $f \colon \mathbb{N} \to \mathbb{N}$ such that any connected graph G from $\mathcal{C}$ of radius at most r has treewidth bounded by $f(r)$.

Eppstein [21] proved that among minor-closed classes, that is, classes satisfying property (C1), property (C2) is equivalent to excluding some apex graph as a minor. Later, Demaine and Hajiaghayi [16] showed that actually for every apex-minor-free class the function f witnessing locally bounded treewidth can be chosen to be linear in the radius. As observed in [16], these facts combined yield the following.

THEOREM 11 ([16, 21]). *If a class of graphs $\mathcal{C}$ satisfies properties* (C1) *and* (C2), *then there is a constant $\alpha(\mathcal{C})$ such that for every connected graph G from $\mathcal{C}$ of radius r the treewidth of G is bounded by $\alpha(\mathcal{C}) \cdot r$.*

The results of [16, 21] in particular show that if $\mathcal{C}$ is a class of graphs that exclude a fixed apex graph as a minor, then the closure of $\mathcal{C}$ under taking minors satisfies properties (C1) and (C2), and hence also the property implied by Theorem 11. Hence, it suffices to prove Theorem 1 for any graph class $\mathcal{C}$ that satisfies properties (C1) and (C2); we will assume this from now on.

We will need an algorithmic variant of Theorem 11, which requires approximating treewidth. For this, we use the following result of Feige, Hajiaghayi, and Lee [22].

THEOREM 12 (Theorem 6.4 of [22]). *For every fixed graph H there is a polynomial time algorithm that, given an H-minor-free graph G of treewidth at most t, computes a tree decomposition of G of width at most $\gamma \cdot |V(H)|^2 \cdot t$ for some universal constant γ.*

Theorems 11 and 12, together with Lemma 10, yield the following.

COROLLARY 13. *For every class of graphs $\mathcal{C}$ satisfying properties* (C1) *and* (C2) *there exists a constant $\bar{\alpha}(\mathcal{C})$ and polynomial-time algorithm that, given a graph $G \in \mathcal{C}$ of radius at most r together with a weight function $\mathbf{w}\colon V(G) \to \mathbb{R}_{\geqslant 0}$, finds a $\frac{1}{2}$-balanced separator for $\mathbf{w}\colon V(G) \to \mathbb{R}_{\geqslant 0}$ of size at most $\bar{\alpha}(\mathcal{C}) \cdot r$.*

Proof. Observe that, by Theorem 11, $\mathcal{C}$ excludes the clique on $\alpha(\mathcal{C})+2$ vertices as a minor, for this clique has radius 1 and treewidth $\alpha(\mathcal{C})+1$. Let G be the input graph. By Theorem 11, the treewidth of G is at most $\alpha(\mathcal{C}) \cdot r$, so running the algorithm of Theorem 12 on G yields a tree decomposition of G of width at most $\gamma(\alpha(\mathcal{C})+2)^2\alpha(\mathcal{C})\cdot r$. By Lemma 10, one of the bags of this tree decomposition is a $\frac{1}{2}$-balanced separator for $\mathbf{w}$ of size at most $\bar{\alpha}(\mathcal{C}) := 1 + \gamma(\alpha(\mathcal{C}) + 2)^2\alpha(\mathcal{C}) \cdot r$. ∎

We remark that the degree of the polynomial in the running time of the algorithm of Theorem 12 is a universal constant, independent of the excluded minor H. More precisely, the algorithm runs in time $f(H)\cdot|V(G)|^c$, where f is some function and c is a constant independent of H. Consequently, the same also holds for the algorithm given by Corollary 13: the degree of the polynomial is a universal constant independent of the class $\mathcal{C}$.

4. Auxiliary tools.

In this section we introduce auxiliary technical tools that will be needed in the proof: a clustering procedure that reduces the radius of the graph, and a duality result concerning almost disjoint paths and chains of separators between a pair of vertices.

4.1. Clustering procedure.

Let us recall Theorem 8 from section 2.

THEOREM 8. *There exists a randomized polynomial-time algorithm that, given a graph G on $n > 1$ vertices and a positive integer k, returns a vertex subset $B \subseteq V(G)$ with the following properties:*

(a) *The radius of each connected component of $G[B]$ is less than $9k^2 \lg n$.*
(b) *For each vertex subset $X \subseteq V(G)$ of size at most k, the probability that $X \subseteq B$ is at least $1 - \frac{1}{k}$.*

Proof. Consider the following iterative procedure which constructs a set B_0. We start with $V_0 = V(G)$. In step i, given a set $V_{i-1} \subseteq V(G)$, we terminate the procedure if $V_{i-1} = \emptyset$. Otherwise we pick an arbitrary vertex $v_i \in V_{i-1}$ and we randomly select a radius r_i according to the geometric distribution with success probability

$p := \frac{1}{2k^2}$. Given v_i and r_i, we insert $\mathrm{Ball}_{G[V_{i-1}]}(v_i, r_i - 1)$ into B_0 and define $V_i := V_{i-1} \setminus \mathrm{Ball}_{G[V_{i-1}]}(v_i, r_i)$. That is, we delete from the graph vertices within distance at most r_i from v_i in $G[V_{i-1}]$ and put the vertices within distance *less* than r_i into B_0. Finally, at the end of the procedure, if any of the selected radii r_i is larger than $9k^2 \lg n$, we return $B = \emptyset$, and otherwise we return $B = B_0$.

Clearly, the procedure runs in polynomial time, as at every step at least the vertex v_i is removed from V_{i-1}, and hence at most n iterations are executed. For the radii of the connected components of $G[B_0]$, note that the fact that we insert into B_0 vertices within distance *less* than r_i from v_i, but delete from V_i vertices within distance *at most* r_i from v_i, ensures that at every step i of the iteration we have $N_G[B_0] \cap V_i = \emptyset$. Consequently, $\mathrm{Ball}_{G[V_{i-1}]}(v_i, r_i - 1)$ induces a connected component of $G[B_0]$ and is of radius less than r_i. Since we return $B = \emptyset$ instead of $B = B_0$ if any of the selected radii r_i exceeds $9k^2 \lg n$, the upper bound on the radii of the connected components of $G[B]$ follows.

It remains to argue that any fixed k-vertex subset $X \subseteq V(G)$ survives in B with high probability. First, note that for fixed i we have that

$$\mathbb{P}(r_i > 9k^2 \lg n) \leqslant (1-p)^{9k^2 \lg n} \leqslant e^{-4.5 \lg n} < n^{-3} < \frac{1}{2kn};$$

here, we use the fact that $p = \frac{1}{2k^2}$ and the inequality $1 - x \leqslant e^{-x}$. Thus, as there are at most n iterations with probability less than $\frac{1}{2k}$ the algorithm returns $B = \emptyset$ because of some r_i exceeding the limit of $9k^2 \lg n$.

Second, we analyze the probability that $X \subseteq B_0$. Let us fix some $x \in X$. The only moment where the vertex x could be deleted from the graph, but not put into B_0, is when x is within distance exactly r_i from the vertex v_i in an iteration i. It is now useful to think of the choice of r_i in the iteration i as follows: we start with $r_i := 1$ and then, iteratively, with probability p accept the current radius, and with probability $1-p$ increase the radius r_i by one and repeat. However, in the aforementioned interpretation of the geometric distribution, when $r_i = \mathrm{dist}_{G[V_{i-1}]}(v_i, x)$, with probability p the radius r_i is accepted (and x is deleted but not put in B_0), but with probability $(1-p)$ the radius r_i is increased, and the vertex x is included in the ball $\mathrm{Ball}_{G[V_{i-1}]}(v_i, r_i - 1) \subseteq B_0$. Consequently, the probability that a fixed vertex $x \in X$ is not put into B_0 is at most p. By the union bound, we infer that the probability that $X \not\subseteq B_0$ is at most $kp = \frac{1}{2k}$. Together with the $\frac{1}{2k}$ upper bound on the probability that the maximum radius among r_i exceeds $9k^2 \lg n$, which results in putting $B = \emptyset$ instead of $B = B_0$, we have that the probability that $X \not\subseteq B$ is at most $\frac{1}{k}$. This concludes the proof. ∎

4.2. Duality. We start with a few standard definitions.

DEFINITION 14. *For a graph H and its vertex u, by $\mathsf{reach}(u, H)$ we denote the set of vertices of H reachable from u in H. Suppose G is a connected graph and s, t are its different vertices. An (s,t)-*separator *is a subset C of vertices of G such that $s, t \notin C$ and $t \notin \mathsf{reach}(s, G - C)$. An (s,t)-separator is* minimal *if none of its proper subsets is also an (s,t)-separator.*

DEFINITION 15. *A sequence $(C_1, C_2, \ldots, C_k)$ of minimal (s,t)-separators is called an (s,t)-*separator chain *if all of them are pairwise disjoint and for each $1 \leqslant j < j' \leqslant k$ the following holds:*

$$C_j \subseteq \mathsf{reach}(s, G - C_{j'}) \qquad \textit{and} \qquad C_{j'} \subseteq \mathsf{reach}(t, G - C_j).$$

We now state and prove the main duality result, that is, Theorem 9 from section 2.

THEOREM 9. *There is a polynomial-time algorithm that given a connected graph G, a pair $s, t \in V(G)$ of different vertices, and positive integers p, q, outputs one of the following structures in G:*

(a) *A chain $(C_1, \dots, C_p)$ of (s,t)-separators with $|C_j| \leqslant 2q$ for each $j \in [p]$.*

(b) A sequence $(P_1, \dots, P_q)$ of (s,t)-*paths with* $|(V(P_i) \cap \bigcup_{i' \neq i} V(P_{i'})) \setminus \{s,t\}| \leqslant 4p$ *for each $i \in [q]$.*

Proof. Our approach is as follows: we formulate searching for the second output as a min-cost max-flow problem in an auxiliary graph H. If the cost of the computed flow is not too large, a simple averaging argument yields the desired paths P_i from the flow paths. If the cost is large, we look at the dual of the min-cost max-flow problem, expressed as a linear program, which is in fact a distance LP. Then we read the separators C_i as layers of distance from the vertex s. Let us now proceed with formal argumentation.

We define a graph H as follows. Starting with $H := G$, we replace every vertex $v \in V(G) \setminus \{s,t\}$ with two copies v_0 and v_1: the copy v_0 has capacity 1 and cost 0, while the copy v_1 has infinite capacity and cost 1. The vertex s is a source of capacity $2q$ and cost 0, and the vertex t is a sink of capacity $2q$ and cost 0. The edges of H are defined naturally: every edge uv of G gives rise to up to four edges in H, between the copies of u and the copies of v.

In the graph H, we ask for a minimum-cost vertex-capacitated flow from s to t of size $2q$. Clearly, such a flow exists for connected graphs G, as every vertex v_1 is of infinite capacity. Since all the costs and capacities are integral or infinite, in polynomial time we can find a minimum-cost solution that decomposes into $2q$ flow paths $P'_1, P'_2, \dots, P'_{2q}$, each carrying a unit flow. Let C be the total cost of this flow. Every path P'_i induces a walk P_i in G: whenever P'_i traverses a vertex v_0 or v_1, the path P_i traverses the corresponding vertex v. By shortcutting if necessary, we may assume that each P_i is a path.

In this proof, we consider every path P from s to t (either in G or in H) as oriented from s towards t; thus, the notions of a predecessor/successor on P or the relation of lying before/after on P are well-defined.

Let us define the cost of a path P_i, denoted by $c(P_i)$, as the number of internal vertices that P_i shares with other paths. That is,

$$c(P_i) := \left| (V(P_i) \cap \bigcup_{j \neq i} V(P_j)) \setminus \{s,t\} \right|.$$

Observe that, due to the capacity constraints, if a vertex $v \notin \{s,t\}$ lies on $h > 1$ paths $P_{i_1}, P_{i_2}, \dots, P_{i_h}$, then all but one of the paths P'_{i_j} have to use the vertex v_1, inducing total cost $h-1$ for the minimum-cost flow. Since $h - 1 \geqslant h/2$ for $h > 1$, it follows that

$$\sum_{i=1}^{2q} c(P_i) \leqslant 2C.$$

We infer that if $C \leqslant 2pq$, then $\sum_{i=1}^{2q} c(P_i) \leqslant 4pq$. Therefore, for at least q paths P_i we have $c(P_i) \leqslant 4p$. This yields the second desired output.

Thus we are left with the case when $C > 2pq$. Our goal is to find a separator chain suitable as the first desired output. To this end, we formulate the discussed

$$\begin{aligned}
\min \quad & \sum_{v \in V(G)\setminus\{s,t\}} \sum_{a \in N_H(v_1)} f(v_1, a) & \\
\text{s.t.} \quad & \sum_{b \in N_H(a)} f(a,b) - f(b,a) = 0 & \forall\ a \in V(H) \setminus \{s,t\} \\
& \sum_{a \in N_H(s)} f(s,a) - f(a,s) = 2q & \\
& \sum_{a \in N_H(t)} f(t,a) - f(a,t) = -2q & \\
& \sum_{a \in N_H(v_0)} f(v_0, a) \leqslant 1 & \forall\ v \in V(G) \setminus \{s,t\} \\
& f(a,b) \geqslant 0 & \forall\ ab \in E(H)
\end{aligned}$$

FIG. 2. *The minimum-cost flow problem used in the proof of Theorem* 9. *In the flow problem, the variables $f(a,b)$ correspond to the amount of flow pushed from a to b along an edge ab.*

$$\begin{aligned}
\max \quad & 2q(y_t - y_s) - \sum_{v \in V(G)\setminus\{s,t\}} z_v & \\
\text{s.t.} \quad & y_{v_0} \leqslant y_a + z_v & \forall\ v \in V(G) \setminus \{s,t\}, a \in N_H(v_0) \\
& y_{v_1} \leqslant y_a + 1 & \forall\ v \in V(G) \setminus \{s,t\}, a \in N_H(v_1) \\
& y_s \leqslant y_a & \forall\ a \in N_H(s) \\
& y_t \leqslant y_a & \forall\ a \in N_H(t) \\
& z_v \geqslant 0 & \forall\ v \in V(G) \setminus \{s,t\}
\end{aligned}$$

FIG. 3. *The dual of the minimum-cost flow problem from Figure* 2, *used in the proof of Theorem* 9.

minimum-cost flow problem as a linear program, and we analyze its dual. The precise formulations can be found in Figures 2 and 3.

In the dual formulation, the value $y_a - y_s$ can be interpreted as a distance of a from s, where traveling through a vertex v_1 costs 1 and traveling through a vertex v_0 costs z_v. The goal is to maximize the distance from s to t with weight $2q$, while paying as little as possible in the sum $\sum_{v \in V(G)\setminus\{s,t\}} z_v$.

Let $\{z_v \colon v \in V(G);\ y_a \colon a \in V(H)\}$ be an optimum solution to the dual LP. Since the primal program is a minimum-cost flow problem with integral coefficients, in polynomial time we can find such values z_v and y_a that are additionally integral. Observe that the dual is invariant under adding a constant to every variable y_a; hence, we can assume $y_s = 0$. Since traveling through a vertex v_1 incurs distance 1 and v_0 is a twin of v_1, the optimum solution never uses values z_v greater than 1; hence, $z_v \in \{0,1\}$ for every $v \in V(G) \setminus \{s,t\}$.

On one hand, C, as the optimum value of both the primal and the dual LPs, is assumed to be larger than $2pq$. On the other hand, we have that $z_v \geqslant 0$ for every $v \in V(G) \setminus \{s,t\}$. We infer that

$$2q(y_t - y_s) \geqslant C > 2pq,$$

and hence $y_t > p$. We define for every $1 \leqslant j \leqslant p$ the set

$$C_j := \{v \in V(G) \colon z_v = 1 \wedge y_{v_0} = j\}.$$

We claim that $C_1, C_2, \ldots, C_p$ is the desired separator chain. Clearly, the sets are pairwise disjoint and do not contain either s or t. We now show that they form a separator chain.

CLAIM 16. *For each $1 \leqslant j \leqslant p$, the set C_j is an (s,t)-separator.*

Proof. Consider a path P from s to t. Let P' be the corresponding path in H that traverses a vertex v_0 whenever $v \in V(G) \setminus \{s,t\}$ lies on P. Since $z_v \in \{0,1\}$ for every $v \in V(G) \setminus \{s,t\}$, we have $y_b \leqslant y_a + 1$ for every $ab \in E(H)$. As $y_t > p$, there exists a vertex b on P' with $y_b = j$; let b be the first such vertex and let a be its predecessor on P'. Note that $b \neq s$ as $y_s = 0$ and $b \neq t$ as $y_t > p$, and hence $b = v_0$ for some $v \in V(G) \setminus \{s,t\}$. Since b is the first vertex on P' with $y_b = j$, we have $y_a = j-1$, and, consequently, $z_v = 1$. Thus $v \in C_j$. Since the choice of P is arbitrary, C_j is an (s,t)-separator, as desired. ∎

Consider a path P_i'. By complementary slackness conditions, whenever the path P_i' traverses an edge $ab \in E(H)$ from a to b, the corresponding distance inequality of the dual LP is tight: $y_b = y_a + 1$ if $b = v_1$ for some v, $y_b = y_a + z_v$ if $b = v_0$, and $y_b = y_a$ if $b = s$ or $b = t$. Thus, P_i' is a shortest path from s to t in the graph H with vertex weights 0 for s, t, z_v for every v_0, and 1 for every v_1. In particular, always $y_a \leqslant y_b \leqslant y_a + 1$ for b being a successor of a on P_i'.

CLAIM 17. *For all $1 \leqslant j \leqslant p$ and $1 \leqslant i \leqslant 2q$, we have that $|V(P_i) \cap V(C_j)| = 1$.*

Proof. We first prove that the cardinality of this intersection is at most 1. Assume $b \in P_i'$ such that $b = v_0$ or $b = v_1$ for some $v \in C_j$, and let a be the predecessor of b on P_i'. Since $z_v = 1$, we have $y_b = y_a + 1 = j$, that is, b must be the first vertex on P_i' with $y_b = j$. Consequently, there exists at most one vertex b on P_i' that projects to a vertex of C_j in G, which proves that $|V(P_i) \cap V(C_j)| \leqslant 1$.

To prove the converse inequality, we show that the projection onto G of the first vertex b on P_i' such that $y_b = j$ belongs to C_j. First, observe that such a vertex b exists since $y_s = 0$, $y_t > p$, and $y_b \leqslant y_a + 1$ for every $ab \in E(H)$. Let $b = v_0$ or $b = v_1$ for some $v \in V(G) \setminus \{s,t\}$; we claim that $z_v = 1$, $y_{v_0} = j$, and hence $v \in C_j$. If $b = v_0$, the claim is immediate, as $y_b = y_a + 1$ for the predecessor a of b on P_i'. By contradiction, let us assume $b = v_1$ but $z_v = 0$. Then by replacing $b = v_1$ with v_0 on P_i' we obtain a shorter path from s to t in H, contradicting the fact that P_i' is a shortest path from s to t. Thus $z_v = 1$ and, consequently, $y_{v_0} = y_{v_1} = y_a + 1$, so v belongs to C_j, as claimed. ∎

Let us denote the unique vertex of $V(P_i) \cap V(C_j)$ as $w_{i,j}$. Note that $w_{i,j}$ and $w_{i',j}$ may coincide for different indices i, i'. By the complementary slackness conditions again, if $z_v = 1$ for a vertex $v \in V(G) \setminus \{s,t\}$, then there exists a flow path P_i' that passes through v_0. Consequently, for every $1 \leqslant j \leqslant p$ and every $v \in C_j$, there exists a flow path P_i' that passes through v_0. It follows that $v = w_{i,j}$, and thus $C_j = \{w_{i,j} : 1 \leqslant i \leqslant 2q\}$. In particular, we have $|C_j| \leqslant 2q$ for every $1 \leqslant j \leqslant p$. Moreover, since each vertex of C_j lies on some path P_i, and is the unique vertex of $V(P_i) \cap V(C_j)$, we infer the C_j is a minimal separator.

We are left with verifying the inclusion of reachability sets. Since P_i' is a shortest path from s to t in H, we have that the vertex $w_{i,j}$ lies before the vertex $w_{i,j'}$ on the path P_i, whenever $j < j'$. As the separators C_j and $C_{j'}$ respectively consist only of vertices $w_{i,j}$ and $w_{i,j'}$, already the reachability within paths P_i certifies that that $C_j \subseteq \mathsf{reach}(s, G - C_{j'})$ and $C_{j'} \subseteq \mathsf{reach}(t, G - C_j)$ for every $1 \leqslant j < j' \leqslant p$, as requested. This concludes the proof of the theorem. ∎

5. Proof of the main result. In this section we give a formal proof of our main result, that is, Theorem 1. Recall that we have fixed a graph class $\mathcal{C}$ from which the input graph is drawn, and we assumed that it satisfies properties (C1) and (C2): it is minor-closed, and it has locally bounded treewidth with a linear dependence of the treewidth on the radius; we argued in section 3 that these assumptions can be made. In the proof we will use the constant $\bar{\alpha}(\mathcal{C})$ yielded by Corollary 13 for $\mathcal{C}$.

Let $G_0 \in \mathcal{C}$ be the input graph, and let k be the requested upper bound on the sizes of patterns X that we need to cover. Throughout the proof we assume that $k \geqslant \max(10, 2^{\bar{\alpha}(\mathcal{C})})$, because otherwise the result is trivial: as k is bounded by a constant, we can just sample k vertices of the graph uniformly and independently at random and return them as A. Hence, we may assume that $k \geqslant 10$ and $\lg k \geqslant \bar{\alpha}(\mathcal{C})$.

The initial value of k is fixed throughout the whole proof; in particular, it will not change in recursive calls. Therefore, we will use it as a fixed parameter in various formulas in what follows.

5.1. Recursive scheme and potentials. Our algorithm will construct the set A by means of a recursive procedure that roughly partitions the graph into smaller and smaller pieces, at each point making some random decisions. For the analysis, we fix some pattern X, that is, a subset X of vertices such that $G[X]$ is connected and $|X| \leqslant k$. Recall that our goal is to construct A in such a manner that the probability that X is covered by A is at least the inverse of $2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot n^{\mathcal{O}(1)}$. The steps taken by the algorithm obviously will not depend on X, but at each random step we argue about the *success probability*: the probability that the taken decision is compliant with the target pattern X, that is, leads to its coverage.

The general problem: Definitions. As usual with recursive algorithms, we need to consider a more general problem, which will be supplied with a few potential measures. Formally, an instance $\mathcal{I}$ of the general problem is a 6-tuple consisting of the following:

(i) A connected graph G that is a minor of the original graph G_0; hence in particular $G \in \mathcal{C}$.
(ii) A specified vertex $r \in V(G)$ called the *root*.
(iii) Two disjoint vertex subsets $T^{\mathsf{li}}, T^{\mathsf{he}} \subseteq V(G)$, called *light terminals* and *heavy terminals*, respectively. We require that the root vertex is a light terminal, that is, $r \in T^{\mathsf{li}}$. By $T := T^{\mathsf{li}} \cup T^{\mathsf{he}}$ we will denote the set of *terminals*.
(iv) A subset of nonterminals $R \subseteq V(G) \setminus T$, called *relay vertices*.
(v) A nonnegative integer λ, called the *credit*.

Terminals represent the boundary via which the currently considered piece communicates with the rest of the original graph, whereas relay vertices represent maximal connected parts of the original graph lying outside of the currently considered piece, each contracted to one vertex. Intuitively, whenever we forget some connected part of the graph from the currently considered piece, we cannot just remove it, because we need to preserve the information about connectivity provided by it. A natural thing would be to apply the so-called *torso* operation, namely to turn the neighborhood of the forgotten part into a clique. However, this might not result in a minor of the original graph, so the graph could cease to belong to the class $\mathcal{C}$. Therefore, we instead contract the forgotten part into one vertex which we declare a relay vertex; its sole purpose is to remember the connectivity information.

For a graph G equipped with relay vertices R, by $G\langle R\rangle$ we define the graph obtained from G by *eliminating* each relay vertex, that is, removing it and turning its neighborhood into a clique; it is easy to see that the order of elimination does not matter for the final result. Note that $G\langle R\rangle$ does not necessarily belong to $\mathcal{C}$. For two vertices x, y in G, we define the *nonrelay-distance measure* $\text{nr-dist}_G(x, y)$ (called for brevity *nr-distance*) as the minimum possible number of nonrelay vertices on a path connecting x and y (including x and y), minus 1. In other words, when a graph is equipped with relay vertices, nonrelay vertices have cost 1 of traversing them, whereas relay vertices have cost 0. Note that if x and y are nonrelay vertices, then $\text{nr-dist}_G(x, y)$ is equal to the (normal) distance between x and y in $G\langle R\rangle$. Whenever we talk about just distances, we mean standard distances in the graph, and we use the term *nr-distance* for the nonrelay-distance measure defined above; the latter will be our main notion of distance in what follows.

In the course of the algorithm, we shall maintain the following invariants; that is, they are satisfied in each considered instance $\mathcal{I} = (G, r, T^{\mathsf{li}}, T^{\mathsf{he}}, R, \lambda)$.

(Inv.a) Every light terminal is at nr-distance at most 3 from the root r.

(Inv.b) It holds that $\lambda \leqslant \sqrt{k}/5$.

(Inv.C) It holds that $|T| \leqslant 16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + \lambda$.

We say that a subset $X \subseteq V(G) \setminus R$ is a *pattern* in instance $\mathcal{I}$ if the following conditions are satisfied:

(i) The root r is contained in X.

(ii) Every vertex of X can be reached from r by a path that traverses only vertices of $X \cup R$.

(iii) $|X| \leqslant k - 10\sqrt{k} \cdot \lambda$.

In particular, every pattern has at most k vertices, but we may consider only smaller patterns if the credit is positive. The intuition behind the credit is that it measures how much of the pattern is assumed to reside outside of the currently considered part of the whole graph. Note that the relay vertices provide free connectivity for the pattern.

Potentials. In order to measure the advancement of the algorithm, we introduce three auxiliary potentials that are intended to measure three different types of possible progress. First, for an instance $\mathcal{I} = (G, r, T^{\mathsf{li}}, T^{\mathsf{he}}, R, \lambda)$, we define the subset $\mathsf{Far}_{\mathcal{I}}(X)$ of *far* vertices as follows:

$$\mathsf{Far}_{\mathcal{I}}(X) := \{u \in X \ : \ \text{nr-dist}_G(u, r) > 1000\sqrt{k}\lg k\}.$$

If a vertex of X is not far, it is said to be *close*. Obviously, by invariant (Inv.a) we have that no far vertex is a light terminal, that is, $\mathsf{Far}_{\mathcal{I}}(X) \cap T^{\mathsf{li}} = \emptyset$. For a pattern X in $\mathcal{I}$, we define the following potentials.

Pattern potential	$\Pi_{\mathcal{I}}(X) := \lvert X \setminus T^{\mathsf{li}}\rvert$
Graph potential	$\Gamma_{\mathcal{I}} := \lvert V(G) \setminus (T^{\mathsf{li}} \cup R)\rvert$
Distance potential	$\Phi_{\mathcal{I}}(X) := \lvert\mathsf{Far}_{\mathcal{I}}(X)\rvert$

We drop the subscript $\mathcal{I}$ whenever the instance $\mathcal{I}$ is clear from the context.

The general problem: Statement of the result. With all definitions in place, we may state formally what our goal is in the general problem. This is encapsulated in the following theorem, whose proof will span the remainder of this section.

THEOREM 18. *There is a polynomial-time randomized algorithm that, given an instance* $\mathcal{I} = (G, r, T^{\mathsf{li}}, T^{\mathsf{he}}, R, \lambda)$ *of the general problem with* $G \in \mathcal{C}$ *and satisfying invariants* (Inv.a)–(Inv.c), *samples a subset* $A \subseteq V(G) \setminus R$ *of nonrelay vertices with the following properties:*

(P1) *It holds that* $T^{\mathsf{li}} \subseteq A$, *and the graph* $G[A]$ *admits a tree decomposition of width at most* $24022\sqrt{k}\lg k$, *where* $T \cap A$ *is contained in the root bag.*

(P2) *For every pattern* X *in instance* $\mathcal{I}$, *we have that*

$$\mathbb{P}(X \subseteq A) \geqslant \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n}{\sqrt{k}} \cdot (\Pi(X)\lg\Pi(X) + \Phi(X))\right] \cdot \left(1 - \frac{1}{k}\right)^{c_2\Pi(X)\lg\Gamma} \tag{1}$$

for some positive constants c_1, c_2, *where* $n := |V(G)|$.

The constants c_1, c_2 will be fixed while explaining the proof. Actually, we will fix $c_1 = c_2 = 2$, but we find it more instructive to treat them symbolically instead of putting actual values in order to show how different parameters depend on each other. For convenience, in the following we denote the right-hand side of (1) by

$$\mathsf{LB}(n, \Pi(X), \Gamma, \Phi(X)),$$

and we regard it as a function of the potentials and the number of vertices n. Recall here that k is the initial bound on the size of the pattern, which is considered fixed throughout the whole proof.

Note that in the context of Theorem 18 both X and A reside in the graph with the relay vertices removed. The intuition is that relay vertices do not belong to the piece of the graph that we are currently decomposing, but we cannot forget about them completely, because they may provide connectivity for the pattern.

Applying Theorem 18. Before we proceed to the proof of Theorem 18, we show that it implies our main result, Theorem 1. For this, we need the following simple claim.

CLAIM 19. *The following holds:*

$$2^{\sqrt{k}\lg k \lg\lg n} \leqslant 2^{\sqrt{k}\lg^2 k} \cdot n^{o(1)}.$$

Proof. The left-hand side is equal to $(\lg n)^{\sqrt{k}\lg k}$. Suppose first that $n \leqslant 2^k$. Then

$$(\lg n)^{\sqrt{k}\lg k} \leqslant k^{\sqrt{k}\lg k} = 2^{\sqrt{k}\lg^2 k}.$$

Suppose second that $n > 2^k$. Then

$$(\lg n)^{\sqrt{k}\lg k} \leqslant (\lg n)^{\sqrt{\lg n}\lg\lg n} = 2^{\sqrt{\lg n}\cdot(\lg\lg n)^2} = n^{o(1)}.$$

Hence in both cases we are done. ◻

Proof of Theorem 1 *assuming Theorem* 18. Let G be the input graph, and let $n := |V(G)|$. We first sample uniformly at random one vertex r of G. Let G' be the connected component of G that contains r. Next, we apply the algorithm for the general problem to the instance $\mathcal{I}_0 := (G', r, \{r\}, \emptyset, \emptyset, 0)$. Fix some subset $X \subseteq V(G)$ with $G[X]$ being connected and $|X| \leqslant k$. Conditioned on the event that $r \in X$, which happens with probability at least $1/n$, the algorithm for the general problem returns a suitable vertex subset A that covers X with probability lower bounded by $\mathsf{LB}(n', \Pi_{\mathcal{I}_0}(X), \Gamma_{\mathcal{I}_0}, \Phi_{\mathcal{I}_0}(X))$, where $n' := |V(G')| \leqslant n$. Observe that $\Pi_{\mathcal{I}_0}(X) \leqslant k$,

$\Phi_{\mathcal{I}_0}(X) \leqslant k$ and $\Gamma_{\mathcal{I}_0} \leqslant n'$. Therefore, using the fact that $1-x \geqslant e^{-2x}$ for $x \in [0,1/2]$ and Claim 19, we infer that

$$\begin{aligned}\mathbb{P}(X \subseteq A | r \in X) &\geqslant \exp\left[-c_1\sqrt{k}\lg k(\lg k + \lg\lg n')\right] \cdot \left(1 - \frac{1}{k}\right)^{c_2 \cdot k \lg n'} \\ &\geqslant \exp\left[-c_1\sqrt{k}\lg^2 k - c_1\sqrt{k}\lg k \lg\lg n' - \frac{2}{k}\cdot c_2 k \lg n'\right] \\ &\geqslant \exp\left[-2c_1\sqrt{k}\lg^2 k - o(\lg n') - 2c_2\lg n'\right] \\ &= \left[2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot (n')^{\mathcal{O}(1)}\right]^{-1} \leqslant \left[2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot n^{\mathcal{O}(1)}\right]^{-1}.\end{aligned}$$

By multiplying this by the $1/n$ probability that indeed $r \in X$, we obtain the success probability as required in Theorem 1. ∎

5.2. Solving the general problem: Opening moves. We now proceed to the proof of Theorem 18. The goal is to provide a suitable recursive procedure for the general problem. This procedure is roughly summarized using pseudocode as Algorithm 1, but the reader should regard this pseudocode just as a roadmap to the detailed description that follows.

Throughout the description we fix the considered instance $\mathcal{I} = (G, r, T^{\mathsf{li}}, T^{\mathsf{he}}, R, \lambda)$ and denote $n := |V(G)|$. In this subsection we will explain the opening steps of the algorithm, which more or less boil down to finding a separator that splits the instance in a balanced way. At this moment a crucial decision should be (randomly) made: whether the pattern contains some "very deep" vertex of the separator or not. The treatment of those two cases will be given in the subsequent two subsections.

Additional assumptions on the instance. We shall assume that $\mathcal{I}$ satisfies invariants (Inv.a)–(Inv.c). Moreover, we will make the following assumptions about $\mathcal{I} = (G, r, T^{\mathsf{li}}, T^{\mathsf{he}}, R, \lambda)$ and the sought pattern X in $\mathcal{I}$:

- (Inv.d) Relay vertices in $\mathcal{I}$ are pairwise nonadjacent, and no relay vertex is adjacent to the root.
- (Inv.e) It holds that $\lambda \leqslant \sqrt{k}/10$; this strengthens invariant (Inv.b).
- (Inv.f) There is at least one vertex that is neither a terminal nor a relay vertex.
- (Inv.g) We restrict our attention only to patterns X in $\mathcal{I}$ for which $T^{\mathsf{li}} \subsetneq X$, or equivalently $\Pi_{\mathcal{I}}(X) > 0$.

Note that (Inv.c) and (Inv.e) together imply that $|T| \leqslant 16015\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$. We now explain how the assumptions above can be guaranteed, as is formalized in the following lemma.

LEMMA 20. *Let $\mathcal{I} = (G, r, T^{\mathsf{li}}, T^{\mathsf{he}}, R, \lambda)$ be the input instance. Then:*

- (i) *If $\mathcal{I}$ does not satisfy assumption* (Inv.d), *that is, in G there is an edge connecting two relay vertices or a relay vertex with the root, then contracting this edge in G does not increase any of the potentials of $\mathcal{I}$ and preserves the family of patterns in $\mathcal{I}$.*
- (ii) *If $\mathcal{I}$ does not satisfy assumption* (Inv.e) *or* (Inv.f), *then $A := T$ is a valid outcome of the algorithm that covers every pattern X in $\mathcal{I}$ with probability* 1.
- (iii) *If pattern X in $\mathcal{I}$ is such that $\Pi_{\mathcal{I}}(X) = 0$, then every set A output by the algorithm, provided it satisfies $A \supseteq T^{\mathsf{li}}$ (condition* (P1) *of Theorem* 18*), covers X.*

Algorithm 1 Procedure `Solve`.

Input: An instance $(G, r, T^{\mathsf{li}}, T^{\mathsf{he}}, R, \lambda)$ satisfying invariants (Inv.a)–(Inv.c)
Output: A subset $A \subseteq V(G) \setminus R$ with $T^{\mathsf{li}} \subseteq A$

Apply cleaning steps of Lemma 20 so that invariants (Inv.d)–(Inv.g) are satisfied
$M \leftarrow \{u \in V(G) \colon \text{nr-dist}_G(r, u) \leqslant 2000\sqrt{k}\lg k\}$
Apply the algorithm of Theorem 8 to $G - M$, yielding B' as described in Step 1
$G_{\mathsf{cl}} \leftarrow$ connected component of $G[M \cup B']$ containing r
$T^{\mathsf{li}}_{\mathsf{cl}} \leftarrow T^{\mathsf{li}} \cap V(G_{\mathsf{cl}})$; $T^{\mathsf{he}}_{\mathsf{cl}} \leftarrow T^{\mathsf{he}} \cap V(G_{\mathsf{cl}})$; $R_{\mathsf{cl}} \leftarrow R \cap V(G_{\mathsf{cl}})$
$\mathcal{I}_{\mathsf{cl}} \leftarrow (G_{\mathsf{cl}}, r, T^{\mathsf{li}}_{\mathsf{cl}}, T^{\mathsf{he}}_{\mathsf{cl}}, R_{\mathsf{cl}}, \lambda)$
Apply the algorithm of Lemma 23 to the instance $\mathcal{I}_{\mathsf{cl}}$, yielding sets W_{isl} and W_{nrm}
islandsDisjointWithPattern $\leftarrow$ `true` with probability $1 - 1/k$ and `false` otherwise
if islandsDisjointWithPattern **then**
  $G' \leftarrow G_{\mathsf{cl}} - W_{\mathsf{isl}}$
  Construct instances $\mathcal{I}_D$ for $D \in \mathsf{cc}(G' - W_{\mathsf{nrm}})$ as described in section 5.3
  for all $D \in \mathsf{cc}(G' - W_{\mathsf{nrm}})$ **do**
    $A_D \leftarrow$ `Solve`$(\mathcal{I}_D)$
  end for
  Merge sets $\{A_D \colon D \in \mathsf{cc}(G' - W_{\mathsf{nrm}})\}$ into A as described before Step 3
  return A
else
  Select an island C satisfying $C \subseteq W_{\mathsf{isl}}$ uniformly at random
  Select a vertex z in C that is at nr-distance $\leqslant 9k^2 \lg n$ from every vertex of C
  Select $d \in \{0, 1, \ldots, \lfloor 9k^2 \lg n \rfloor\}$ uniformly at random
  $S \leftarrow w \in C$ with $\text{nr-dist}(z, w) < \max(d, 1)$ ($\max(d, 1) - 1$ if $w \in R$)
  $G'' \leftarrow G_{\mathsf{cl}}$ with S contracted onto z
  Apply the algorithm of Theorem 9 to $s = r$, $t = z$, $p = \lceil 120\sqrt{k}\lg k \rceil$ and $q = k$
  if obtained a sequence of (r, z)-paths $P_1, P_2, \ldots, P_k$ **then**
    Select $i \in \{1, \ldots, k\}$ uniformly at random
    $H \leftarrow G''$ with parts of P_i contracted as described before Claim 31
    $\mathcal{I}' \leftarrow (H, r, T^{\mathsf{li}}, T^{\mathsf{he}} \cap V(H), R_H, \lambda)$
    $A \leftarrow$ `Solve`$(\mathcal{I}')$
    return A
  end if
  if obtained an (r, z)-separator chain $C_1, C_2, \ldots, C_p$ **then**
    $C_1, \ldots, C'_p \leftarrow C_1, \ldots, C_p$ with the first three separators removed
    Select $i \in \{1, \ldots, p'\}$ uniformly at random
    Select $\alpha \in \{1, \ldots, |C_i|\}$ uniformly at random
    Select a subset $Q \subseteq C_i$ with $|Q| = \alpha$ uniformly at random
    Construct $\mathcal{I}_{\mathsf{out}}$ as described after Claim 34
    Construct $\mathcal{I}_{\mathsf{in}}$ as described after Claim 34
    $A_{\mathsf{out}} \leftarrow$ `Solve`$(\mathcal{I}_{\mathsf{out}})$
    $A_{\mathsf{in}} \leftarrow$ `Solve`$(\mathcal{I}_{\mathsf{in}})$
    $A \leftarrow (A_{\mathsf{out}} \setminus Q) \cup A_{\mathsf{in}}$
    return A
  end if
end if

Proof. Point (i) is obvious.

For point (ii), if assumption (Inv.e) is not satisfied, that is, the credit λ exceeds $\sqrt{k}/10$, then the upper bound $k-10\lambda\sqrt{k}$ on the sizes of considered patterns is negative, so the family of patterns in $\mathcal{I}$ is empty. Similarly, if assumption (Inv.f) is not satisfied, that is, every vertex of G is either a terminal or a relay, then every pattern X in G is contained in T. Therefore, in both cases we may return $A := T$ with a trivial tree decomposition consisting of one root node with bag equal to A, thus making sure that A covers every possible pattern X in $\mathcal{I}$. This decomposition has width at most $|T| \leqslant 16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + \lambda$, which is at most $16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + \sqrt{k}/5 < 24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ by invariant (Inv.b).

Finally, for point (iii), observe that if $X \subseteq T^{\mathsf{li}}$, then any output set A satisfying $A \supseteq T^{\mathsf{li}}$ will cover X for sure. ∎

Thus, the message of Lemma 20 is the following. For assumption (Inv.d) we may exhaustively contract any edges contradicting this assumption, and we may only decrease the potentials while not losing any pattern. For assumptions (Inv.e) and (Inv.f), if any of them is not satisfied, then we may immediately terminate the algorithm by returning $A := T$. For assumption (Inv.g), any pattern X with $\Pi_{\mathcal{I}}(X) = 0$ will be automatically covered, provided we return a set A that contains all light terminals, which we require anyway. This means that from now on we may assume that the considered instance $\mathcal{I}$ and pattern X in $\mathcal{I}$ satisfy all the assumptions (Inv.a)–(Inv.g).

Margin, islands, and clustering. As outlined in the proof overview, the following definitions are pivotal for the structural analysis of the instance.

DEFINITION 21. *We define the* margin M *of the instance* $\mathcal{I}$ *as follows:*

$$M := \{u \in V(G) \colon \text{nr-dist}_G(r,u) \leqslant 2000\sqrt{k}\lg k\}.$$

Every connected component of $G - M$ *will be called an* island.

Obviously, the margin M can be computed in linear time using a breadth-first search from r (here we need a trivial modification to traverse relay vertices at cost 0). Note that, by the definition of M, every vertex of an island that neighbors some vertex of M cannot be a relay vertex. Hence, in particular every island contains some nonrelay vertex. Observe also that every close vertex, in particular every light terminal, is within the margin M.

The first step of the algorithm is to apply the clustering procedure of Theorem 8 for parameter k to all the islands. More precisely, we apply the algorithm of Theorem 8 to the graph $K := (G - M)\langle R \setminus M\rangle$, that is, to $G - M$ with all the relay vertices eliminated. This algorithm works in randomized polynomial time and returns a subset $B' \subseteq V(K)$ with the following properties:

- each connected component of $K[B']$ has radius at most $9k^2 \lg n$, and
- with probability at least $1 - 1/k$ we have that $X \setminus M$ is contained in B'.

Now define B to be B' extended by adding all relay vertices of $R \setminus M$ that have at least one neighbor in B'. Then each connected component of $G[B]$ has radius at most $9k^2 \lg n$, computed according to the nr-distance measure nr-dist_G that treats relay vertices as traversed for free. Moreover, with probability at least $1 - 1/k$ we have that $X \setminus M \subseteq B$. We henceforth assume that this event happens, i.e., indeed $X \subseteq M \cup B$, keeping in mind the multiplicative factor of $1 - 1/k$ in the success probability. The operations described above are summarized in the following step, which corresponds to line 3 of the pseudocode of Algorithm 1.

Step 1. Apply the clustering algorithm of Theorem 8 to the graph $K := (G - M)\langle R \setminus M\rangle$ and parameter k, yielding a set B'. Extend B' to B by including all relay vertices outside of M that are adjacent to B'. From now on assume that $X \subseteq M \cup B$; this assumption is correct with probability at least $1 - 1/k$.

By the assumption made in Step 1, we can restrict our attention to the graph G_{cl} defined as the connected component of $G[M \cup B]$ that contains r. More precisely, we define a new instance $\mathcal{I}_{\mathsf{cl}}$ as

$$\mathcal{I}_{\mathsf{cl}} := (G_{\mathsf{cl}}, r, T^{\mathsf{li}}_{\mathsf{cl}}, T^{\mathsf{he}}_{\mathsf{cl}}, R_{\mathsf{cl}}, \lambda),$$

where

$$T^{\mathsf{li}}_{\mathsf{cl}} := T^{\mathsf{li}} \cap V(G_{\mathsf{cl}}), \qquad T^{\mathsf{he}}_{\mathsf{cl}} := T^{\mathsf{he}} \cap V(G_{\mathsf{cl}}), \qquad R_{\mathsf{cl}} := R \cap V(G_{\mathsf{cl}}).$$

The following simple claim summarizes the properties of $\mathcal{I}_{\mathsf{cl}}$.

CLAIM 22. *It holds that $M \subseteq V(G_{\mathsf{cl}})$ and $T^{\mathsf{li}}_{\mathsf{cl}} = T^{\mathsf{li}}$. Moreover, X remains a pattern in $\mathcal{I}_{\mathsf{cl}}$ and $\mathsf{Far}_{\mathcal{I}_{\mathsf{cl}}}(X) = \mathsf{Far}_{\mathcal{I}}(X)$.*

Proof. For the first claim, observe that $G[M]$ contains r and is connected by definition, and hence $M \subseteq V(G_{\mathsf{cl}})$. Since $T^{\mathsf{li}} \subseteq M$, it follows that $T^{\mathsf{li}}_{\mathsf{cl}} = T^{\mathsf{li}}$. For the latter claim, recall that we assumed that $X \subseteq M \cup B$. Further, by the definitions of M and B it follows that $R \cap N_G(M \cup B) = \emptyset$; that is, no relay vertex lies outside of $M \cup B$ and has neighbors in $M \cup B$. Since in G every vertex of X can be reached from r using only vertices of $X \cup R$, it follows that X is entirely contained in the connected component of r in $G[M \cup B]$, which is G_{cl} by definition. Thus X remains a pattern in $\mathcal{I}_{\mathsf{cl}}$. Finally, note that when constructing G_{cl} from G we remove only vertices at nr-distance larger than $2000\sqrt{k}\lg k$ from r in G, and hence every vertex of X that was close or far remains close or far, respectively; this implies that $\mathsf{Far}_{\mathcal{I}_{\mathsf{cl}}}(X) = \mathsf{Far}_{\mathcal{I}}(X)$. ∎

Note that Claim 22 implies the following relation between potentials in $\mathcal{I}$ and $\mathcal{I}_{\mathsf{cl}}$:

$$\Pi_{\mathcal{I}}(X) = \Pi_{\mathcal{I}_{\mathsf{cl}}}(X), \qquad \Gamma_{\mathcal{I}} \geqslant \Gamma_{\mathcal{I}_{\mathsf{cl}}}, \qquad \Phi_{\mathcal{I}}(X) = \Phi_{\mathcal{I}_{\mathsf{cl}}}(X). \tag{2}$$

Hence, restricting our attention to $\mathcal{I}_{\mathsf{cl}}$ can only decrease the potentials, resulting in obtaining a better lower bound on the success probability. A straightforward check verifies that assumptions (Inv.a)–(Inv.g) still hold for $\mathcal{I}_{\mathsf{cl}}$ and X, with a possible exception of (Inv.f). However, if in $\mathcal{I}_{\mathsf{cl}}$ there is no nonrelay, nonterminal vertex, then we may conclude by outputting $A = T$, as in the proof of Lemma 20. Hence from now on we assume that $\mathcal{I}_{\mathsf{cl}}$ and X also satisfy assumptions (Inv.a)–(Inv.g), in particular $\Gamma_{\mathcal{I}_{\mathsf{cl}}} > 0$ by (Inv.f).

By slightly abusing the notation, when analyzing the instance $\mathcal{I}_{\mathsf{cl}}$, we redefine the islands to be the connected components of $G_{\mathsf{cl}} - M$.

Finding a balanced separator. The next step of the algorithm is to construct a separator that splits both the graph and the terminal set $T_{\mathsf{cl}} := T^{\mathsf{li}}_{\mathsf{cl}} \cup T^{\mathsf{he}}_{\mathsf{cl}}$ in a balanced way. The existence of such a separator would follow from a bound on the treewidth of the current graph, which in turn would follow from a bound on its radius. Unfortunately, so far we do not have any good bound on the radius of the graph. However, the idea is to contract the islands outside of the margin M to single vertices, thus obtaining a graph of radius $\widetilde{\mathcal{O}}(\sqrt{k})$, hence also of treewidth $\widetilde{\mathcal{O}}(\sqrt{k})$, find a balanced separator in this graph, and then uncontract the islands. Thus, the obtained balanced separator will consist of $\widetilde{\mathcal{O}}(\sqrt{k})$ vertices of the margin M and $\widetilde{\mathcal{O}}(\sqrt{k})$ islands. We will not have any guarantee about its size, but we will be able to proceed nonetheless. This plan is formalized as follows.

LEMMA 23. *Given the instance $\mathcal{I}_{\mathsf{cl}}$, we can in polynomial time find sets W_{nrm} and W_{isl} satisfying the following conditions.*

(i) *The set W_{nrm} consists of at most $8007\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ vertices of M, and $r \in W_{\mathsf{nrm}}$.*
(ii) *The set W_{isl} consists of the union of vertex sets of at most $8007\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ islands (i.e., connected components of $G_{\mathsf{cl}} - M$).*
(iii) *For every connected component D of $G_{\mathsf{cl}} - (W_{\mathsf{nrm}} \cup W_{\mathsf{isl}})$, we have that*

$$|V(D) \setminus (T_{\mathsf{cl}}^{\mathsf{li}} \cup R_{\mathsf{cl}})| \leqslant |V(G_{\mathsf{cl}}) \setminus (T_{\mathsf{cl}}^{\mathsf{li}} \cup R_{\mathsf{cl}})|/2 \quad \textit{and} \quad |V(D) \cap T_{\mathsf{cl}}| \leqslant |T_{\mathsf{cl}}|/2.$$

Proof. Construct an auxiliary graph H from G_{cl} by contracting each island $C \in \mathsf{cc}(G_{\mathsf{cl}} - M)$ into a single (nonrelay) vertex u_C; let $I = \{u_C : C \in \mathsf{cc}(G_{\mathsf{cl}} - M)\}$. By $\iota : V(G_{\mathsf{cl}}) \to V(H)$ we denote the function that assigns to each vertex of G_{cl} its image under the contraction; i.e., ι is identity on M and $\iota(V(C)) = \{u_C\}$ for each island C. Obviously H is a minor of G_{cl}, which in turn is a subgraph of G, and hence $H \in \mathcal{C}$. Moreover, each vertex of H is at nr-distance at most $2000\sqrt{k}\lg k + 1 \leqslant 2001\sqrt{k}\lg k$ from r. Hence, if we measure the distance in H normally, without eliminating the relay vertices, then each vertex is at distance at most $4002\sqrt{k}\lg k$ from r; this follows from assumption (Inv.d) that no two relay vertices are adjacent. This means that the radius of H, measured normally, is at most $4002\sqrt{k}\lg k$.

Define two weight functions $\mathbf{w}_1(u)$ and $\mathbf{w}_2(u)$ on $V(H)$ as follows. For a vertex $u \notin I$, we put $\mathbf{w}_1(u) = 1$ if $u \in T_{\mathsf{cl}}$ and $\mathbf{w}_1(u) = 0$ otherwise. However, if $u = u_C$ for some island C, then we put $\mathbf{w}_1(u_C) = |V(C) \cap T_{\mathsf{cl}}|$. Similarly, for $u \notin I$, we put $\mathbf{w}_2(u) = 1$ if $u \in V(G) \setminus (T_{\mathsf{cl}}^{\mathsf{li}} \cup R_{\mathsf{cl}})$ and $\mathbf{w}_2(u) = 0$ otherwise, whereas for $u = u_C$, we put $\mathbf{w}_2(u) = |V(C) \setminus (T_{\mathsf{cl}}^{\mathsf{li}} \cup R_{\mathsf{cl}})|$. In other words, $\mathbf{w}_1$ and $\mathbf{w}_2$ are characteristic functions of T_{cl} and of $V(G_{\mathsf{cl}}) \setminus (T_{\mathsf{cl}}^{\mathsf{li}} \cup R_{\mathsf{cl}})$, where all vertices contained in one island C contribute to the weight of the collapsed vertex u_C.

By applying Corollary 13 to these weight functions, we infer that in polynomial time we can compute sets Z_1 and Z_2, each of size at most $4002\bar{\alpha}(\mathcal{C}) \cdot \sqrt{k}\lg k + 1 \leqslant 4003\bar{\alpha}(\mathcal{C}) \cdot \sqrt{k}\lg k$, such that every connected component of $H - Z_1$ has $\mathbf{w}_1$-weight at most $|T_{\mathsf{cl}}|/2$, and every connected component of $H - Z_2$ has $\mathbf{w}_2$-weight at most $|V(G_{\mathsf{cl}}) \setminus (T_{\mathsf{cl}}^{\mathsf{li}} \cup R_{\mathsf{cl}})|/2$. We define

$$\begin{aligned} Z &:= Z_1 \cup Z_2 \cup \{r\} \; ; \\ W_{\mathsf{isl}} &:= \iota^{-1}(Z \cap I); \\ W_{\mathsf{nrm}} &:= \iota^{-1}(Z \setminus I). \end{aligned}$$

Clearly, we have

$$|Z| \leqslant |Z_1| + |Z_2| + 1 \leqslant 8006\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + 1 \leqslant 8007\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k.$$

Hence, W_{nrm} consists of at most $8007\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ vertices of M, and it contains r due to adding it explicitly to Z, whereas W_{isl} is the union of the vertex sets of at most $8007\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ islands. This proves conditions (i) and (ii). Condition (iii) follows immediately from the definition of $\mathbf{w}_1$ and $\mathbf{w}_2$, and the properties of Z_1 and Z_2 as $\frac{1}{2}$-balanced separators for $\mathbf{w}_1$ and $\mathbf{w}_2$, respectively. ∎

The algorithm now branches into two cases based on a random decision as follows. With probability $1 - 1/k$ the algorithm assumes that W_{isl} is disjoint with the sought pattern X; that is, no island contained in W_{isl} intersects X. In the remaining event, which happens with probability $1/k$, the algorithm assumes that X intersects some island contained in W_{isl}. The operations described above are summarized in the following step, which corresponds to lines 7–8 of the pseudocode of Algorithm 1.

Step 2. Compute sets W_{nrm} and W_{isl} using the algorithm of Lemma 23. Randomly branch into one of two cases as follows. With probability $1 - 1/k$ assume that W_{isl} is disjoint with X, and otherwise assume that X intersects W_{isl}.

We now describe steps taken by the algorithm in each of the cases, supposing the assumption made is correct. The success probability analysis will be explained at the end of each case.

5.3. Solving the general problem: When islands are disjoint with the pattern. In this subsection we present the steps made by the algorithm, provided that in Step 2 it chose to assume that $W_{\mathsf{isl}} \cap X = \emptyset$. We proceed under the assumption that this choice is correct, i.e., indeed X does not contain any vertex of W_{isl}.

The intuition is that the assumption $W_{\mathsf{isl}} \cap X = \emptyset$ implies that it is safe for the algorithm to restrict our attention to the graph

$$G' := G_{\mathsf{cl}} - W_{\mathsf{isl}}.$$

This graph admits a small balanced separator in the form of W_{nrm}; by Lemma 23 this separator breaks the graph into components, each containing at most half of the terminals and at most half of the (nonrelay, nonlight-terminal) vertices. Therefore, the strategy is as follows: inspect the connected components of $G' - W_{\mathsf{nrm}} = G_{\mathsf{cl}} - (W_{\mathsf{isl}} \cup W_{\mathsf{nrm}})$; for each such connected component D define a simpler instance $\mathcal{I}_D$; apply the algorithm to instances $\mathcal{I}_D$ recursively, yielding sets A_D; and combine all sets A_D into one output set A. In the remainder of this section we implement this plan formally. Unfortunately, we need to be very careful when defining and analyzing the instances $\mathcal{I}_D$, so that the amortized analysis using the potentials goes through.

Before we proceed, let us observe the following properties of G' and X.

CLAIM 24. *The graph G' is connected, X is contained in G', and every vertex of X can be reached from r in G' by a path traversing only the vertices of X and $R_{\mathsf{cl}} \cap V(G')$.*

Proof. For the first claim, observe that G' is obtained from G_{cl}—which is connected by definition—by removing some connected components of $G_{\mathsf{cl}} - M$. Since $G_{\mathsf{cl}}[M]$ is connected, it follows that G' is connected. The second claim follows directly from the assumption that $W_{\mathsf{isl}} \cap X = \emptyset$. Finally, for the third claim, observe that by the way we defined M we have that $R_{\mathsf{cl}} \cap N_{G_{\mathsf{cl}}}(M) = \emptyset$; that is, there is no relay vertex outside of M that has neighbors in M. In G_{cl}, each vertex $x \in X$ can be reached from r by a path P_x traversing only vertices of $X \cup R_{\mathsf{cl}}$. In G' we remove from G_{cl} only some connected components of $G_{\mathsf{cl}} - M$, which are disjoint from X and do not contain any relay vertices having neighbors in M. Hence, it follows that P_x in fact traverses only vertices of $X \cup (R_{\mathsf{cl}} \cap V(G'))$. ∎

Defining subinstances. Our first goal is to define, for each connected component D of $G' - W_{\mathsf{nrm}}$, a new instance $\mathcal{I}_D$ to which the algorithm will be applied

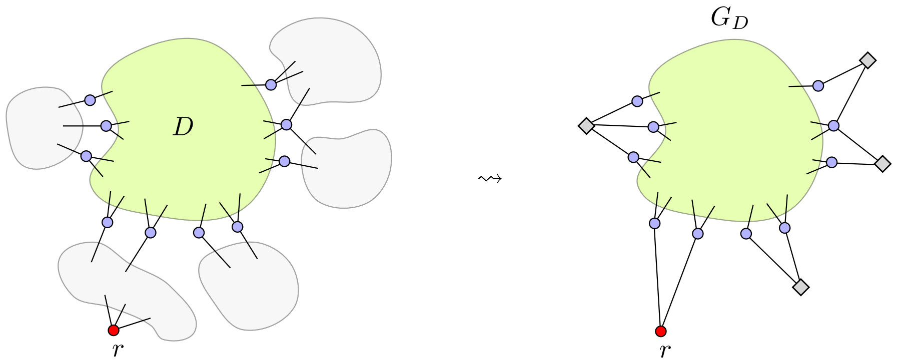

Fig. 4. *Construction of the graph G_D from a connected component $D \in \mathsf{CC}$. The blue vertices are the vertices of $N_D \setminus \{r\}$, and the gray rhombi are the new relay vertices g_Q constructed in the process.*

recursively. Let CC be the set of connected components of the graph $G' - W_{\mathsf{nrm}}$. Let us fix a component $D \in \mathsf{CC}$. We shall now construct an instance

$$\mathcal{I}_D := (G_D, r, T_D^{\mathsf{li}}, T_D^{\mathsf{he}}, R_D, \lambda)$$

corresponding to this component.

We begin by defining the graph G_D, which will be constructed from G' by a series of contractions. Define

$$N_D := N_{G'}[V(D)] \cup \{r\}.$$

We now consider the connected components of the graph $G' - N_D$. For each component $Q \in \mathsf{cc}(G' - N_D)$ that contains a neighbor of the root vertex r, contract the whole Q onto r. For each component $Q \in \mathsf{cc}(G' - N_D)$ that does not contain any neighbor of r, contract the whole Q into a new vertex g_Q, and declare it a relay vertex. We define G_D to be the graph obtained from G' by applying the contractions specified above for each connected component Q of $G' - N_D$; see Figure 4. In particular, the set R_D of relay vertices in G_D is defined as

$$R_D := (R_{\mathsf{cl}} \cap N_D) \cup \{g_Q \colon Q \in \mathsf{cc}(G' - N_D) \text{ and no vertex of } Q \text{ is adjacent to } r\}.$$

In other words, R_D consists of $R_{\mathsf{cl}} \cap N_D$, plus we add g_Q to R_D for each $Q \in \mathsf{cc}(G' - N_D)$ that is not adjacent to r. Obviously, G_D is still connected and contains the root vertex r.

Next, we need to define terminals in the instance $\mathcal{I}_D$. For this, construct an auxiliary graph L from G' by contracting every connected component $D \in \mathsf{CC}$ into a single vertex w_D. Observe that, thus, each vertex of L is of one of two kinds: it is either an original vertex of G' that belongs to W_{nrm} (these include the root r), or it is of the form w_D for some $D \in \mathsf{CC}$. Note that the vertices of the second kind are pairwise nonadjacent in L, as they originate from the connected components of $G' - W_{\mathsf{nrm}}$. Obviously L is connected, as G' was, and r persists in L. We treat L as a graph without any relay vertices, so all distances are computed normally.

Run a breadth-first search (BFS) in L starting from r. Let S be the tree of this BFS, rooted at r. For every vertex $v \in W_{\mathsf{nrm}} \setminus \{r\}$, inspect the parent of v in S. If this parent is equal to w_D for some component $D \in \mathsf{CC}$, then we shall say that v is *charged* to D. Note that a vertex $v \in W_{\mathsf{nrm}}$ is not charged to any component $D \in \mathsf{CC}$ if and only if $v = r$ or the parent of v in S is not of the form w_D for some $D \in \mathsf{CC}$.

The intuition is that once vertices of $W_{\mathsf{nrm}} \setminus \{r\}$ are turned into terminals in instances $\mathcal{I}_D$ for $D \in \mathsf{CC}$, we want that for every vertex $v \in W_{\mathsf{nrm}} \setminus \{r\}$ only at most instance is responsible for handling v as a heavy terminal; this is the instance $\mathcal{I}_D$ constructed for $D \in \mathsf{CC}$ to which v is charged. This is necessary for ensuring a correct split of the pattern potential among the instances $\mathcal{I}_D$.

With this charging scheme, we are ready to define terminals in instance $\mathcal{I}_D$, for each $D \in \mathsf{CC}$. We put the following:

$$T_D^{\mathsf{li}} := (T_{\mathsf{cl}}^{\mathsf{li}} \cap N_D) \cup \{v \in N_{G'}(V(D)) \setminus (T_{\mathsf{cl}}^{\mathsf{li}} \cup R_{\mathsf{cl}}) \colon\ v \text{ is not charged to } D\};$$
$$T_D^{\mathsf{he}} := (T_{\mathsf{cl}}^{\mathsf{he}} \cap V(D)) \cup \{v \in N_{G'}(V(D)) \setminus (T_{\mathsf{cl}}^{\mathsf{li}} \cup R_{\mathsf{cl}}) \colon\ v \text{ is charged to } D\}.$$

In other words, the terminals in the instance $\mathcal{I}_D$ comprise terminals that were originally in component D, plus we add also all nonrelay vertices from the boundary $N_{G'}(V(D))$ as terminals. These new terminals are partitioned into light and heavy depending on whether they are charged to D. However, any vertex that was originally a light terminal in $\mathcal{I}_{\mathsf{cl}}$ remains a light terminal; this is in order to force the subinstances to cover them in recursive calls. From the definition it directly follows that T_D^{li} and T_D^{he} are disjoint, as required. Note that each vertex $v \in W_{\mathsf{nrm}}$ is charged to at most one component $D \in \mathsf{CC}$, so it can be declared a heavy terminal in at most one instance $\mathcal{I}_D$.

Analyzing the subinstances. To be able to apply the algorithm recursively on instances $\{\mathcal{I}_D\}_{D \in \mathsf{CC}}$, we need to make sure that they all satisfy invariants (Inv.a)–(Inv.c). Since each instance $\mathcal{I}_D$ has the same credit λ as $\mathcal{I}_{\mathsf{cl}}$, invariant (Inv.b) holds for each instance $\mathcal{I}_D$. The satisfaction of invariants (Inv.c) and (Inv.a) is respectively verified in the following two claims.

CLAIM 25. *For each* $D \in \mathsf{CC}$, *we have* $|T_D| \leqslant 16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + \lambda$.

Proof. From Lemma 23, assertions (i) and (iii), the satisfaction of invariant (Inv.c) in $\mathcal{I}_{\mathsf{cl}}$, and the fact that $N_{G'}(V(D)) \subseteq W_{\mathsf{nrm}}$, we have that

$$|T_D| \leqslant |T_{\mathsf{cl}} \cap V(D)| + |W_{\mathsf{nrm}}| \leqslant |T_{\mathsf{cl}}|/2 + 8007\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k \leqslant 16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + \lambda,$$

as claimed. ◻

CLAIM 26. *For each* $D \in \mathsf{CC}$, *in graph* G_D *every vertex of* T_D^{li} *is at nr-distance at most* 3 *from* r.

Proof. Fix some $v \in T_D^{\mathsf{li}}$. Suppose first that $v \in T_{\mathsf{cl}}^{\mathsf{li}} \cap N_D$. As we observed earlier, the invariant (Inv.a) still holds in the instance $\mathcal{I}_{\mathsf{cl}}$, so v was already at nr-distance at most 3 from r in the graph G_{cl}. During the construction of G' from G_{cl} we removed only vertices at nr-distance more than $2000\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ from r, and hence v is still at nr-distance at most 3 from r in G'. Graph G_D was obtained from G' by means of edge contractions, which can only decrease the nr-distances. Hence, v is at nr-distance at most 3 from r also in G_D.

Consider now the remaining case when $v \in N_{G'}(V(D)) \setminus (T_{\mathsf{cl}}^{\mathsf{li}} \cup R_{\mathsf{cl}})$ and v is not charged to D. If $v = r$, then we are done, so assume otherwise. Let P be the path from r to v in S; recall that S was the tree of a BFS from r in the graph L that was used to define charging. Since v is not charged to D, the parent of v in S tree is not equal to w_D. As w_D is a neighbor of v in L, due to $v \in N_{G'}(V(D))$, this implies that

w_D cannot lie on path P. Indeed, otherwise we could shortcut P by using the edge $w_D v$, which contradicts the fact that P is a shortest path from r to v in L.

Let P' be the path P lifted to the graph G' in the following manner: for every vertex of the form $w_{D'} \in V(L)$ for some $D' \in \mathsf{CC}$ visited on P, we replace the visit of this vertex by the traversal of any path within the connected component D'. Then P' is a path from r to v in G' that does not traverse any vertex of D. Let v' be the first vertex on P' that belongs to $N_{G'}(V(D))$; clearly, such a vertex exists, because $v \in N_{G'}(V(D))$. Observe that in the construction of G_D the prefix of P from r to v', excluding v', gets entirely contracted onto r. Hence, v' is a neighbor of r in G_D.

Finally, consider the suffix of P from v' to v. Observe that both v and v' are neighbors of w_D in the graph L, so since P is a shortest path, we infer that the distance between v' and v on P is at most 2. This implies that the suffix of P' between v and v' gets contracted to a path of length at most 2 in G_D. In this manner we have uncovered a walk of length at most 3 from r to v in G_D, which concludes the proof. ☐

Partitioning the pattern. To apply recursion, we also need to partition the pattern X among the instances $\{\mathcal{I}_D\}_{D\in\mathsf{CC}}$. More precisely, for each instance $\mathcal{I}_D$ we would like to define the *projection* X_D of X onto $\mathcal{I}_D$ so that, intuitively, the following conditions are satisfied:

- each X_D should be a pattern in $\mathcal{I}_D$;
- if applying the algorithm to each instance $\mathcal{I}_D$ yields a set A_D covering X_D, then the combination of A_D-s (to be defined later) should cover X; and
- the pattern and distance potentials are split between subinstances, so that the lower bounds on the probabilities of covering the pattern multiply correctly.

Formally, for the pattern X in $\mathcal{I}_{\mathsf{cl}}$, we define its *projection* X_D onto instance $\mathcal{I}_D$ by simply putting

$$X_D := X \cap N_D.$$

Let us verify that X_D is indeed a pattern in $\mathcal{I}_D$.

CLAIM 27. *Each vertex of X_D can be reached from r in G_D by a path that traverses only vertices from $X_D \cup R_D$. Consequently, X_D is a pattern in $\mathcal{I}_D$.*

Proof. By Claim 24, every vertex $x \in X$ can be reached from r in G' by a path P_x traversing only vertices of $X \cup (R_{\mathsf{cl}} \cap V(G'))$. In the construction of G_D from G', every connected component of $G' - N_D$ has been either contracted onto r or contracted into a single relay vertex. By applying these contractions to P, we obtain a walk in G_D from r to x that uses only vertices of $X_D \cup R_D$. This, together with the observation that $|X_D| \leqslant |X| \leqslant k - 10\sqrt{k} \cdot \lambda$, certifies that X_D is a pattern in $\mathcal{I}_D$. ☐

Analyzing the split of potentials. It is crucial for our amortized analysis that the potentials of the original instance $\mathcal{I}_{\mathsf{cl}}$ and pattern X are split among the potentials in the defined instances $\{\mathcal{I}_D\}_{D\in\mathsf{CC}}$ and projections $\{X_D\}_{D\in\mathsf{CC}}$ of X onto them. This is because when we apply the algorithm recursively to each instance $\mathcal{I}_D$ we will obtain some lower bound on the probability of covering X_D. The product of these lower bounds should be not smaller than the requested lower bound on the probability of covering X; for this condition to hold, the potentials need to be split. The following claim verifies this formally.

CLAIM 28. *The following hold:*

$$\Pi_{\mathcal{I}_{\mathsf{cl}}}(X) \geqslant \sum_{D\in\mathsf{CC}} \Pi_{\mathcal{I}_D}(X_D), \tag{3}$$

$$\Gamma_{\mathcal{I}_{\mathsf{cl}}} \geqslant \sum_{D \in \mathsf{CC}} \Gamma_{\mathcal{I}_D}, \tag{4}$$

$$\Gamma_{\mathcal{I}_{\mathsf{cl}}}/2 \geqslant \Gamma_{\mathcal{I}_D} \qquad \textit{for each } D \in \mathsf{CC}, \tag{5}$$

$$\Phi_{\mathcal{I}_{\mathsf{cl}}}(X) \geqslant \sum_{D \in \mathsf{CC}} \Phi_{\mathcal{I}_D}(X_D). \tag{6}$$

Proof. Take any vertex $u \in V(G_{\mathsf{cl}}) \setminus (T^{\mathsf{li}}_{\mathsf{cl}} \cup R_{\mathsf{cl}})$. By the construction of instances $\mathcal{I}_D$, and in particular the fact that each vertex of W_{nrm} can be declared a heavy terminal in at most one instance $\mathcal{I}_D$, it follows that there is at most one instance $\mathcal{I}_D$ for which $u \in V(G_D) \setminus (T^{\mathsf{li}}_D \cup R_D)$. From this observation we immediately obtain statements (3) and (4). Statement (6) follows similarly, but one needs to additionally observe the following: G_D is obtained from G' by means of edge contractions that can only decrease the nr-distances from r, so if a vertex $u \in X_D$ is far in the instance $\mathcal{I}_D$, then it was also far in the original instance $\mathcal{I}_{\mathsf{cl}}$. Finally, (5) follows directly from Lemma 23(iii). ∎

Thus, (5) in Claim 28 certifies that the graph potential drops significantly in each new instance. Intuitively, this drop will be responsible for amortizing the $(1-1/k)^2$ multiplicative factor in the success probability incurred by the preliminary clustering step and by the random choice of the assumption on the considered case.

Recursive application. Note that, by assumption (Inv.f) (there is at least one nonrelay, nonterminal vertex), we have that $\Gamma_{\mathcal{I}_{\mathsf{cl}}} > 0$. Hence, by Claim 28, (5) in particular, each of the instances $\mathcal{I}_D$ has strictly fewer vertices than $\mathcal{I}_{\mathsf{cl}}$.

Therefore, we may apply the algorithm recursively to each instance $\mathcal{I}_D$. This yields subsets of vertices $\{A_D\}_{D \in \mathsf{CC}}$ with the following properties satisfied for each $D \in \mathsf{CC}$:

- It holds that $A_D \supseteq T^{\mathsf{li}}_D$ and $G_D[A_D]$ admits a tree decomposition $\mathcal{T}_D$ of width at most $24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ with $T_D \cap A_D$ contained in the root bag.
- The probability that $X_D \subseteq A_D$ is at least $\mathsf{LB}(n_D, \Pi_{\mathcal{I}_D}(X_D), \Gamma_{\mathcal{I}_D}, \Phi_{\mathcal{I}_D}(X_D))$, where we denote $n_D := |V(G_D)|$.

Let us define how sets $\{A_D\}_{D \in \mathsf{CC}}$ are combined to yield the final set A.

- First, for every $D \in \mathsf{CC}$, we put $V(D) \cap A_D$ into A.
- Second, for every $v \in W_{\mathsf{nrm}} \setminus R_{\mathsf{cl}}$, we put v into A if *for every* $D \in \mathsf{CC}$ such that $v \in N_{G'}(V(D))$ we have $v \in A_D$. In particular, if there is no $D \in \mathsf{CC}$ for which $v \in N_{G'}(V(D))$, we also include v in A.

Note that for every $D \in \mathsf{CC}$ we have $A \cap N_D \subseteq A_D$, but not necessarily $A \cap N_D = A_D$. The above operations are summarized in the following step, which corresponds to lines 11–16 of the pseudocode of Algorithm 1.

Step 3. Having constructed the instances $\{\mathcal{I}_D\}_{D \in \mathsf{CC}}$, apply the algorithm recursively to each of them. Suppose the application in instance $\mathcal{I}_D$ yields a vertex subset A_D. Combine $\{A_D\}_{D \in \mathsf{CC}}$ into A according to the construction above, and output A as the outcome of the algorithm.

To get more intuition about the construction of A, in particular the universal quantification in the second point, let us observe the following. Consider a vertex $v \in W_{\mathsf{nrm}} \setminus R_{\mathsf{cl}}$. This vertex is a terminal in every instance $\mathcal{I}_D$ for which $v \in N_{G'}(V(D))$; it is a heavy terminal in at most one such instance and a light terminal in all other such instances. If $v \in T^{\mathsf{li}}_D$, then we have $v \in A_D$ by property (P1) of the algorithm. Thus, we have $v \notin A$ if and only if the (unique) instance $\mathcal{I}_D$ where $v \in T^{\mathsf{he}}_D$ exists and, furthermore, $v \notin A_D$ for this instance. Hence, intuitively, we allow this particular instance to exclude the vertex v from A_D if it is deemed necessary for some reason; all other instances are required to keep it in the set A_D. On the other hand, note that the vertex v in the instance $\mathcal{I}_D$ where $v \in T^{\mathsf{he}}_D$ contributes to the potential $\Pi_{\mathcal{I}_D}(X_D)$, and does not contribute to this potential in the other instances to which it belongs.

We now formally verify that A defined in this way has the required properties.

CLAIM 29. *The following assertions hold:*

(i) $T^{\mathsf{li}}_{\mathsf{cl}} \subseteq A$;
(ii) *if* $X_D \subseteq A_D$ *for each* $D \in \mathsf{CC}$, *then* $X \subseteq A$; *and*
(iii) $G_{\mathsf{cl}}[A]$ *admits a tree decomposition of width at most* $24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ *with* $T_{\mathsf{cl}} \cap A$ *contained in the root bag.*

Proof. For condition (i), take any $v \in T^{\mathsf{li}}_{\mathsf{cl}}$. Observe that by the construction of instances $\mathcal{I}_D$, in particular the sets T^{li}_D of light terminals in them, we have that whenever $v \in N_{G'}[V(D)]$ for some $D \in \mathsf{CC}$, then v is automatically included in T^{li}_D. By property (P1) of the algorithm (see Theorem 18) this implies that $v \in A_D$, so in the second point of the construction of A we include v in A.

For condition (ii), suppose that indeed $X_D \subseteq A_D$ for each $D \in \mathsf{CC}$. By definition, we have $X_D = X \cap N_D$. Take any $x \in X$. If $x \in V(D)$ for some $D \in \mathsf{CC}$, then also $x \in X_D \cap V(D) \subseteq A_D \cap V(D)$, and hence x is included in A in the first point of its construction. On the other hand, if $x \in W_{\mathsf{nrm}}$, then $x \in N_D$ for all D such that $x \in N_{G'}(V(D))$. Therefore $x \in X \cap N_D = X_D \subseteq A_D$ for all such D, and x is included in A in the second point of its construction.

Finally, for condition (iii), we construct a suitable tree decomposition of $G_{\mathsf{cl}}[A]$ as follows. Create the root node with bag $A \cap (T_{\mathsf{cl}} \cup (W_{\mathsf{nrm}} \setminus R_{\mathsf{cl}}))$ associated with it. Next, for each $D \in \mathsf{CC}$, restrict the decomposition $\mathcal{T}_D$ to the vertices of $A \cap N_D$; that is, remove all the vertices of $A_D \setminus (A \cap N_D)$ from all the bags, thus constructing a tree decomposition $\mathcal{T}'_D$ of $G_D[A \cap N_D]$. Then, for each $D \in \mathsf{CC}$, attach $\mathcal{T}'_D$ below the root node by making its root a child of the root node. Let $\mathcal{T}$ be the obtained decomposition.

To verify that $\mathcal{T}$ is a tree decomposition of $G_{\mathsf{cl}}[A]$, observe that whenever a vertex $v \notin R_{\mathsf{cl}}$ is shared between multiple instances $\mathcal{I}_D$ we have that v is a vertex of $v \in W_{\mathsf{nrm}} \setminus R_{\mathsf{cl}}$ that is a terminal in all of them (i.e., belongs to T_D), and moreover $v \in A$ only if $v \in A_D$ for every instance $\mathcal{I}_D$ where v is present. Since the root bag of each $\mathcal{T}'_D$ contains $A \cap T_D$, it is now easy to see that $\mathcal{T}$ is indeed a tree decomposition of $G_{\mathsf{cl}}[A]$.

We are left with upper bounding the width of $\mathcal{T}$. Observe that

$$\begin{aligned}|A \cap (T_{\mathsf{cl}} \cup (W_{\mathsf{nrm}} \setminus R_{\mathsf{cl}}))| &\leqslant |T_{\mathsf{cl}}| + |W_{\mathsf{nrm}}| \\ &\leqslant 16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + \lambda + 8007\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k \\ &= 24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k;\end{aligned}$$

here, the last inequality follows from assumption (Inv.e) (that $\lambda \leqslant \sqrt{k}/10$). Since each $\mathcal{T}_D$ has width at most $24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$, due to each recursive satisfying property (P1) (see Theorem 18), it follows that $\mathcal{T}$ has width at most $24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$. ∎

Finally, we are left with analyzing the success probability. For this, we fix $c_2 := 2$.

CLAIM 30. *Supposing $X \cap W_{\mathsf{isl}} = \emptyset$, the algorithm outputs a set A satisfying $X \subseteq A$ with probability at least $\mathsf{LB}(n_{\mathsf{cl}}, \Pi_{\mathcal{I}_{\mathsf{cl}}}(X), \Gamma_{\mathcal{I}_{\mathsf{cl}}}, \Phi_{\mathcal{I}_{\mathsf{cl}}}(X))$, where $n_{\mathsf{cl}} := |V(G_{\mathsf{cl}})|$. This includes the $(1-1/k)$ probability of success of the preliminary clustering step and the $(1-1/k)$ probability that the algorithm correctly assumes that $X \cap W_{\mathsf{isl}} = \emptyset$.*

Proof. The preliminary clustering step is correct (that is, the set of removed vertices is disjoint with X) with probability at least $1 - 1/k$. Then, the algorithm makes the correct assumption that $X \cap W_{\mathsf{isl}} = \emptyset$ with probability $1 - 1/k$. By Claim 29(ii), to have that $X \subseteq A$ it suffices to have $X_D \subseteq A_D$ for each $D \in \mathsf{CC}$. The event $X_D \subseteq A_D$ holds with probability at least $\mathsf{LB}(n_D, \Pi_{\mathcal{I}_D}(X_D), \Gamma_{\mathcal{I}_D}, \Phi_{\mathcal{I}_D}(X_D))$, where $n_D \leqslant n_{\mathsf{cl}}$ is the number of vertices in instance $\mathcal{I}_D$. Note that these events are independent. Hence, we have

$$\mathbb{P}(X \subseteq A) \geqslant \left(1 - \frac{1}{k}\right)^2 \cdot \prod_{D \in \mathsf{CC}} \mathsf{LB}(n_D, \Pi_{\mathcal{I}_D}(X_D), \Gamma_{\mathcal{I}_D}, \Phi_{\mathcal{I}_D}(X_D)). \tag{7}$$

From Claim 28, equation (3), we have that $\Pi_{\mathcal{I}_{\mathsf{cl}}}(X) \geqslant \sum_{D \in \mathsf{CC}} \Pi_{\mathcal{I}_D}(X_D)$. Hence, by the convexity of the function $t \mapsto t \lg t$, we obtain

$$\Pi_{\mathcal{I}_{\mathsf{cl}}}(X) \lg \Pi_{\mathcal{I}_{\mathsf{cl}}}(X) \geqslant \sum_{D \in \mathsf{CC}} \Pi_{\mathcal{I}_D}(X_D) \lg \Pi_{\mathcal{I}_D}(X_D). \tag{8}$$

By (8) and the fact that $n_D \leqslant n_{\mathsf{cl}}$ for all $D \in \mathsf{CC}$, we infer that

$$\begin{aligned} &\exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_{\mathsf{cl}}}{\sqrt{k}} \cdot \Pi_{\mathcal{I}}(X) \lg \Pi_{\mathcal{I}}(X)\right] \\ &\leqslant \prod_{D \in \mathsf{CC}} \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_{\mathsf{cl}}}{\sqrt{k}} \cdot \Pi_{\mathcal{I}_D}(X_D) \lg \Pi_{\mathcal{I}_D}(X_D)\right] \\ &\leqslant \prod_{D \in \mathsf{CC}} \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_D}{\sqrt{k}} \cdot \Pi_{\mathcal{I}_D}(X_D) \lg \Pi_{\mathcal{I}_D}(X_D)\right]. \end{aligned} \tag{9}$$

By Claim 28, equation (6), we similarly conclude the following:

$$\exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_{\mathsf{cl}}}{\sqrt{k}} \cdot \Phi(X)\right] \leqslant \prod_{D \in \mathsf{CC}} \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_D}{\sqrt{k}} \cdot \Phi(X_D)\right]. \tag{10}$$

To estimate the last factor in each term $\mathsf{LB}(n_D, \Pi_{\mathcal{I}_D}(X_D), \Gamma_{\mathcal{I}_D}, \Phi_{\mathcal{I}_D}(X_D))$, we use Claim 28, equations (3) and (5). Recall that for any $D \in \mathsf{CC}$ we have

$$\Gamma_{\mathcal{I}_{\mathsf{cl}}}/2 \geqslant \Gamma_{\mathcal{I}_D}.$$

Hence, if we pick $c_2 := 2$, then we obtain the following:

$$\begin{aligned} (1-1/k)^{c_2 \Pi_{\mathcal{I}_{\mathsf{cl}}}(X) \lg \Gamma_{\mathcal{I}_{\mathsf{cl}}}} &= (1-1/k)^{c_2 \Pi_{\mathcal{I}_{\mathsf{cl}}}(X)} \cdot (1-1/k)^{c_2 \Pi_{\mathcal{I}_{\mathsf{cl}}}(X) \lg(\Gamma_{\mathcal{I}_{\mathsf{cl}}}/2)} \\ &\leqslant (1-1/k)^{c_2 \Pi_{\mathcal{I}_{\mathsf{cl}}}(X)} \cdot \prod_{D \in \mathsf{CC}} (1-1/k)^{c_2 \Pi_{\mathcal{I}_D}(X_D) \lg(\Gamma_{\mathcal{I}_{\mathsf{cl}}}/2)} \\ &\leqslant (1-1/k)^{c_2 \Pi_{\mathcal{I}_{\mathsf{cl}}}(X)} \cdot \prod_{D \in \mathsf{CC}} (1-1/k)^{c_2 \Pi_{\mathcal{I}_D}(X_D) \lg \Gamma_{\mathcal{I}_D}}. \end{aligned}$$

Recall that we assumed that $\Pi_{\mathcal{I}_{\mathsf{cl}}}(X) > 0$ (assumption (Inv.g)). Therefore, this yields

$$(11) \qquad (1-1/k)^{c_2 \Pi_{\mathcal{I}_{\mathsf{cl}}}(X) \lg \Gamma_{\mathcal{I}_{\mathsf{cl}}}} \leqslant (1-1/k)^2 \cdot \prod_{D \in \mathsf{CC}} (1-1/k)^{c_2 \Pi_{\mathcal{I}_D}(X_D) \lg \Gamma_{\mathcal{I}_D}}.$$

Putting (7), (9), (10), and (11) together, we obtain that

$$\begin{aligned} \mathbb{P}(X \subseteq A) &\geqslant (1-1/k)^2 \cdot \prod_{D \in \mathsf{CC}} \mathsf{LB}(n_D, \Pi_{\mathcal{I}_D}(X_D), \Gamma_{\mathcal{I}_D}, \Phi_{\mathcal{I}_D}(X_D)) \\ &\geqslant \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_{\mathsf{cl}}}{\sqrt{k}} \cdot (\Pi_{\mathcal{I}_{\mathsf{cl}}}(X) \lg \Pi_{\mathcal{I}_{\mathsf{cl}}}(X) + \Phi_{\mathcal{I}_{\mathsf{cl}}}(X))\right] \\ &\quad \cdot (1-1/k)^{c_2 \Pi_{\mathcal{I}_{\mathsf{cl}}}(X) \lg \Gamma_{\mathcal{I}_{\mathsf{cl}}}} \\ &= \mathsf{LB}(n_{\mathsf{cl}}, \Pi_{\mathcal{I}_{\mathsf{cl}}}(X), \Gamma_{\mathcal{I}_{\mathsf{cl}}}, \Phi_{\mathcal{I}_{\mathsf{cl}}}(X)). \end{aligned}$$

This concludes the proof. ∎

Claim 29, assertions (i) and (iii), and Claim 30 verify that the set A is a correct output for the instance $\mathcal{I}_{\mathsf{cl}}$, provided $X \cap W_{\mathsf{isl}} = \emptyset$. By equations (2) we have

$$\mathsf{LB}(n_{\mathsf{cl}}, \Pi_{\mathcal{I}_{\mathsf{cl}}}(X), \Gamma_{\mathcal{I}_{\mathsf{cl}}}, \Phi_{\mathcal{I}_{\mathsf{cl}}}(X)) \geqslant \mathsf{LB}(n, \Pi_{\mathcal{I}}(X), \Gamma_{\mathcal{I}}, \Phi_{\mathcal{I}}(X)),$$

and hence, by Claim 22, A is also a valid output for the instance $\mathcal{I}$. This concludes the description and the proof of correctness of the algorithm in the case $X \cap W_{\mathsf{isl}} = \emptyset$.

5.4. Solving the general problem: When islands intersect the pattern. We are left with describing the steps taken by the algorithm after taking the assumption that W_{isl} intersects the pattern X; recall that the algorithm takes this decision with probability $\frac{1}{k}$. Henceforth, we assume that this assumption is correct.

First, the algorithm chooses uniformly at random an island C among islands whose vertex sets are included in W_{isl}, and from now on it assumes that X intersects the island C. By Lemma 23, C is sampled from a set of at most $8007\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ islands, among which at least one intersects X, and hence the algorithm correctly fixes an island C intersecting X with probability at least $(8007\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k)^{-1} \geqslant k^{-7}$. Keeping this success probability in mind, from now on we assume that the choice of C was indeed correct.

Recall that, due to the preliminary clustering step, island C has radius bounded by $9k^2 \lg n$, where the radius is evaluated w.r.t. the nr-distance measure nr-dist_C that regards relay vertices as traversed for free (we write C in the subscript to indicate that the metric is restricted to the vertices of C). Select a nonrelay vertex z of C such that $\text{nr-dist}_C(u,z) \leqslant 9k^2 \lg n$ for each $u \in V(C)$. Let

$$d := \min\{\text{nr-dist}_C(z,u) \colon u \in V(C) \cap X\}.$$

Since $V(C) \cap X \neq \emptyset$, we have that $0 \leqslant d \leqslant d_{\max}$, where $d_{\max} = \max\{\text{nr-dist}_C(z,u) \colon u \in V(C)\}$, which is not larger than $1 + 9k^2 \lg n$. The algorithm now samples an integer value between 0 and $d_{\max}$ and assumes henceforth that the sampled value is equal to d. This assumption holds with probability at least $1/(1+d_{\max}) \geqslant (10k^2 \lg n)^{-1}$. Keeping this success probability in mind, we proceed with the assumption that the algorithm knows the correct value of d.

Let

$$S := \{u \in V(C) \setminus R\colon \text{nr-dist}_C(u,z) < \max(d,1)\} \cup \{u \in V(C) \cap R\colon \text{nr-dist}_C(u,z) < \max(d,1)-1\}.$$

That is, S contains all the vertices of C that are at nr-distance less than $\max(d,1)$ from z, but we exclude relay vertices at nr-distance exactly $\max(d,1)-1$. From the definition it readily follows that the induced subgraph $C[S]$ contains z and is connected. Construct graph G'' from G_{cl} by contracting the whole subgraph $G_{\mathsf{cl}}[S]$ onto z; the contracted vertex of G'' will be also denoted as z. Note that if $d \leqslant 1$, then in fact no contraction has been made and $G'' = G_{\mathsf{cl}}$. Observe that, provided the sampled value of d is correct, the set S is disjoint with X (unless $d=0$ when $S=\{z\}$ and no contraction is made). Thus $X \subseteq V(G'')$. Moreover, observe that each vertex of X can be still reached from r in G'' by a path that uses only relay vertices and vertices of X. Indeed, by the definition of S, if S is disjoint with X, then S also does not contain any relay vertex traversed by the aforementioned path. Hence, X can be still regarded as a pattern in G'', where the relay vertices in G'' are inherited from G'. On the other hand, by the definition of d it follows that some vertex of X is at nr-distance at most 1 from z in G'' (so either z, in case $d=0$ and $z \in X$, or a neighbor of z, or a neighbor via one relay vertex). Also, observe that since C is disjoint with M, that is, all vertices of C are at nr-distance larger than $2000\sqrt{k}\lg k$ from r in G'', we have the following:

$$\text{nr-dist}_{G''}(r,z) > 2000\sqrt{k}\lg k. \tag{12}$$

Now we would like to apply the duality theorem, i.e., Theorem 9. Consider graph $L := G''\langle R \cap V(G'')\rangle$ (this is a different graph than L considered in the previous section), a pair of vertices $(s,t)=(r,z)$ of L, and the following parameters:

$$p := \lceil 120\sqrt{k}\lg k\rceil \qquad \text{and} \qquad q := k.$$

By applying Theorem 9 to these, in polynomial time we can compute one of the following structures:

(a) An (r,z)-separator chain $(C_1,\ldots,C_p)$ with $|C_j| \leqslant 2k$ for each $j \in [p]$.
(b) A sequence $(Q_1,\ldots,Q_k)$ of (r,z)-paths with

$$\left|\left(V(Q_i) \cap \bigcup_{i' \neq i} V(Q_{i'})\right) \setminus \{r,z\}\right| \leqslant 4p \qquad \text{for each } i \in [k].$$

The behavior of the algorithm now differs depending on which structure has been uncovered. We start with the simpler case when the algorithm of Theorem 9 yielded a sequence of paths.

Subcase: A sequence of paths. Suppose that the algorithm of Theorem 9 returned a sequence $(Q_1,\ldots,Q_k)$ of (r,z)-paths, where each path contains only at most $4p$ vertices that also belong to other paths, not including z and r. These are paths in graph $G''\langle R \cap V(G'')\rangle$, but we can lift them to (r,z)-paths $P_1,\ldots,P_k$ in G'' in a natural manner as follows: whenever some Q_i traverses an edge obtained from eliminating some relay vertex g, we replace the usage of this edge by a path of length

2 traversing g. If we obtain a walk in this manner, i.e., some relay vertex is used more than once, we shortcut the segment of the walk between the visits of this relay vertex; thus we obtain again a simple path. All in all, we obtain (r,z)-paths $P_1,\ldots,P_k$ in G'' with the following property: for each $i \in [k]$, there can be at most $4p$ nonrelay vertices on P_i that are traversed by other paths $P_{i'}$ for $i' \neq i$. Note that every relay vertex can be used by many paths.

By (12), we have that each P_i has length larger than $2000\sqrt{k}\lg k$ (measured according to the nr-distance, i.e., with relay vertices contributing 0 to the length). For $i \in [k]$, define

$$\mathsf{Pub}(P_i) := \Big(V(P_i) \cap \bigcup_{i' \neq i} V(P_{i'})\Big) \setminus (R \cup \{r,z\}),$$

and

$$\mathsf{Prv}(P_i) := \Big(V(P_i) \setminus \bigcup_{i' \neq i} V(P_{i'})\Big) \setminus (R \cup \{r,z\}).$$

We have that $|\mathsf{Pub}(P_i)| \leqslant 4p$ for all $i \in [k]$ and, by definition, sets $\mathsf{Prv}(P_i)$ are pairwise disjoint. Pattern X has at most k vertices, out of which one is equal to r. Hence, there is at least one index $i \in [k]$ for which $\mathsf{Prv}(P_i)$ is disjoint with X. The algorithm samples one index i from $[k]$ and assumes henceforth that the sampled index has the property stated above. Note that this holds with probability at least $1/k$; keeping this success probability in mind, we proceed with the assumption that the algorithm made a correct choice of i.

Now that $\mathsf{Prv}(P_i)$ is assumed to be disjoint with the sought pattern X, we can get rid of it in the following manner. Consider the consecutive vertices of P_i, traversed in the direction from r to z. Let v_0 be the last light terminal on P_i; vertex v_0 is well defined because r is a light terminal itself. Let P' be the suffix of P_i from v_0 to z (both inclusive). Observe that since v_0, being a light terminal, is at nr-distance at most 3 from r, whereas z is at nr-distance more than $2000\sqrt{k}\lg k$ from r, we have that P' traverses at least $2000\sqrt{k}\lg k - 3 \geqslant 1997\sqrt{k}\lg k$ nonrelay vertices.

Let $v_0, v_1, \ldots, v_\ell = z$ be the vertices of $(\mathsf{Pub}(P_i) \cap V(P')) \cup \{v_0, z\}$ in the order of their appearance on P'. Then clearly $\ell \leqslant |\mathsf{Pub}(P_i)| + 1 \leqslant 485\sqrt{k}\lg k$. For each $j \in \{0,1,\ldots,\ell-1\}$, inspect the segment of P' lying between v_j and v_{j+1} (both exclusive). If this segment contains some relay vertex g_j, then contract it entirely onto g_j; in case there are multiple relay vertices in the segment, select any of them as g_j. Otherwise, if there are no relay vertices within the segment, contract this whole segment onto vertex v_j; see Figure 5 for a visualization. Observe that, by the definition of v_0, no light terminal gets contracted in this manner.

Denote the obtained graph by H; this graph is equipped with a set $R_H := R \cap V(H)$ of relay vertices naturally inherited from G''. Since we assume that X is disjoint with $\mathsf{Prv}(P_i)$, no contracted vertex belonged to X. Hence, it can be easily seen that X is still a pattern in H.

Observe that H has strictly fewer vertices than G for the following reason: path P' traversed at least $1997\sqrt{k}\lg k$ nonrelay vertices at the beginning, but after the contraction it got shortened to at most $485\sqrt{k}\lg k$ nonrelay vertices. Observe also that, since v_0 is at nr-distance at most 3 from r as a light terminal, we have that the nr-distance between r and z in H is at most $3 + 485\sqrt{k}\lg k \leqslant 488\sqrt{k}\lg k$.

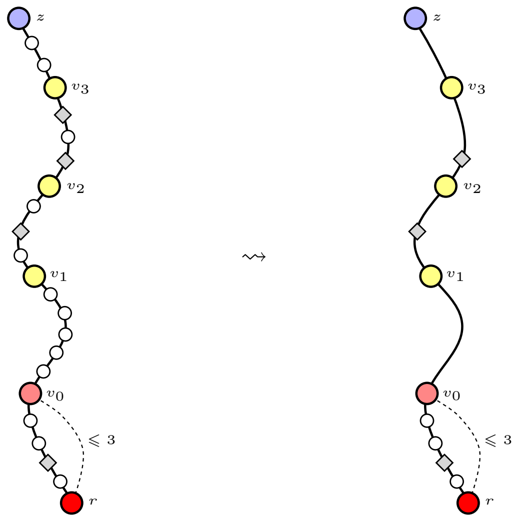

FIG. 5. *The contraction procedure applied on the path P_i. Vertices $v_1, v_2, \ldots, v_{\ell-1}$ are depicted in yellow, and small gray rhombi depict relay vertices.*

Define a new instance $\mathcal{I}' := (H, r, T_H^{\mathsf{li}}, T_H^{\mathsf{he}}, R_H, \lambda)$ by giving it the same credit λ and taking

$$T_H^{\mathsf{li}} := T^{\mathsf{li}} \qquad \text{and} \qquad T_H^{\mathsf{he}} := T^{\mathsf{he}} \cap V(H).$$

Observe here that we use the fact that during the construction of H we did not contract any light terminal, and hence all vertices of T^{li} persist in H. Clearly T_H^{li} and T_H^{he} are disjoint, and observe that invariants (Inv.a) and (Inv.c) trivially hold for $\mathcal{I}'$, because H was obtained from G'' by means of edge contractions. Since no contracted vertex belongs to X, we have that

$$\Pi_{\mathcal{I}'}(X) = \Pi_{\mathcal{I}}(X). \tag{13}$$

Observe that during the construction of H we contracted at least $1511\sqrt{k}\lg k$ nonrelay vertices. This means that the total number of vertices that are not relay or light terminals strictly decreases, so

$$\Gamma_{\mathcal{I}'} < \Gamma_{\mathcal{I}}. \tag{14}$$

This also implies that $n_H < n$, where n_H is the number of vertices of H.

We are left with analyzing the distance potential, which is factor on which we gain in this step. More precisely, the crucial observation is that the performed contraction significantly reduces the number of far vertices.

CLAIM 31. *The following holds:*

$$\mathsf{Far}_{\mathcal{I}'}(X) \subseteq \mathsf{Far}_{\mathcal{I}}(X) \qquad \textit{and} \qquad |\mathsf{Far}_{\mathcal{I}}(X) \setminus \mathsf{Far}_{\mathcal{I}'}(X)| \geqslant 511\sqrt{k}\lg k.$$

Proof. Observe that each close vertex in $\mathcal{I}$ is contained in M, which remains intact in G''. Moreover, H is obtained from G'' only by means of edge contractions, which can only decrease the nr-distances. Hence, every vertex of X that is close in $\mathcal{I}$ remains close in $\mathcal{I}'$. It follows that $\mathsf{Far}_{\mathcal{I}'}(X) \subseteq \mathsf{Far}_{\mathcal{I}}(X)$.

For the second claim, recall that in G'' there is some vertex z' that belongs to X and is at nr-distance at most 1 from z (possibly $z' = z$). On the other hand, by (12) we have $\text{nr-dist}_{G''}(r, z) > 2000\sqrt{k}\lg k$, so $\text{nr-dist}_{G''}(r, z') > 1999\sqrt{k}\lg k$. Let P be a path from r to z' whose vertices belong to X or R. Since $\text{nr-dist}_{G''}(r, z') > 1999\sqrt{k}\lg k$, P contains at least $1999\sqrt{k}\lg k$ vertices of X, among which the last $511\sqrt{k}\lg k$ vertices have to belong to $\mathsf{Far}_{\mathcal{I}}(X)$ for the following reason: their nr-distance from r is larger than $1999\sqrt{k}\lg k - 511\sqrt{k}\lg k > 1000\sqrt{k}\lg k$ by the triangle inequality. However, in $\mathcal{I}'$ we have that the nr-distance between r and z is shortened to at most $488\sqrt{k}\lg k$, and hence all of them become close. ∎

It follows that

$$\Phi_{\mathcal{I}'}(X) \leqslant \Phi_{\mathcal{I}}(X) - 511\sqrt{k}\lg k. \tag{15}$$

Having analyzed the decrease in all the potentials, we are ready to finalize the case.

Apply the algorithm recursively to the instance $\mathcal{I}' = (H, r, T_H^{\mathsf{li}}, T_H^{\mathsf{he}}, R_H, \lambda)$. As discussed earlier, $\mathcal{I}'$ satisfies the requested invariants and has strictly fewer vertices, so this recursive call is correctly defined. The application of the algorithm yields a subset A' of vertices of H with the following properties:

- $A' \supseteq T_H^{\mathsf{li}} = T^{\mathsf{li}}$ and $H[A']$ admits a tree decomposition $\mathcal{T}'$ of width at most $24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ with $T_H \cap A'$ contained in the root bag;
- the probability that $X \subseteq A'$ is at least $\mathsf{LB}(n_H, \Pi_{\mathcal{I}'}(X), \Gamma_{\mathcal{I}'}, \Phi_{\mathcal{I}'}(X))$.

The algorithm returns the set $A := A'$; we now verify that A has all the required properties. Clearly we already have that $A = A' \supseteq T_H^{\mathsf{li}} = T^{\mathsf{li}}$, so let us check that $G[A]$ admits a suitable tree decomposition.

CLAIM 32. *The subgraph $G[A]$ admits a tree decomposition of width at most $24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ with $A \cap T$ contained in the root bag.*

Proof. We observe that decomposition $\mathcal{T} := \mathcal{T}'$ is suitable. First, it can be easily verified that it is also a tree decomposition of $G[A]$, not only $H[A]$, because $G[A]$ is a graph on the same vertex set as $H[A]$ and every edge of $G[A]$ is also present in $H[A]$. Second, from the recursive call we have that the root bag of $\mathcal{T}'$ contains $A \cap T_H$, but $A \cap T_H = A \cap T$, because A contains only vertices that are present in H. Consequently, the root bag of $\mathcal{T}$ contains $A \cap T$. Finally, from the recursive call we obtain that the width of $\mathcal{T} = \mathcal{T}'$ is at most $24022\sqrt{k}\lg k$. ∎

We are left with analyzing the success probability. For this, we assume $c_1 \geqslant 1$.

CLAIM 33. *Assume $c_1 \geqslant 1$. Supposing that $X \cap W_{\mathsf{isl}} \neq \emptyset$ and the subroutine of Theorem 9 returned a sequence of paths, the algorithm outputs a set A satisfying $X \subseteq A$ with probability at least $\mathsf{LB}(n, \Pi(X), \Gamma, \Phi(X))$. This includes the $(1 - 1/k)$ probability of success of the preliminary clustering step, the $1/k$ probability that the algorithm correctly assumes that $X \cap W_{\mathsf{isl}} \neq \emptyset$, the k^{-7} probability of correctly choosing the island C that intersects the pattern, the $(10k^2 \lg n)^{-1}$ probability of choosing the right distance d, and the $1/k$ probability of choosing the right path index i.*

Proof. By the bound on the success probability of the recursive call and the assumption that $k \geqslant 10$, we have that

$$\begin{aligned} \mathbb{P}(X \subseteq A) &\geqslant \left(1 - \frac{1}{k}\right) \cdot k^{-8} \cdot (10k^2 \lg n)^{-1} \cdot \mathsf{LB}(n', \Pi_{\mathcal{I}'}(X), \Gamma_{\mathcal{I}'}, \Phi_{\mathcal{I}'}(X)) \\ &\geqslant k^{-12} \cdot (\lg n)^{-1} \cdot \mathsf{LB}(n', \Pi_{\mathcal{I}'}(X), \Gamma_{\mathcal{I}'}, \Phi_{\mathcal{I}'}(X)). \end{aligned} \tag{16}$$

By (13) and the fact that $n_H < n$, we have

$$\exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n}{\sqrt{k}} \cdot \Pi_{\mathcal{I}}(X) \lg \Pi_{\mathcal{I}}(X)\right] \leqslant \tag{17}$$
$$\exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_H}{\sqrt{k}} \cdot \Pi_{\mathcal{I}'}(X) \lg \Pi_{\mathcal{I}'}(X)\right].$$

By (15) and the facts that $c_1 \geqslant 1$ and $n_H < n$, we have

(18)
$$\exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n}{\sqrt{k}} \cdot \Phi_{\mathcal{I}}(X)\right] \leqslant \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_H}{\sqrt{k}} \cdot \Phi_{\mathcal{I}'}(X)\right] \cdot \exp\left[-c_1 \cdot 511 \lg k(\lg k + \lg\lg n)\right]$$
$$\leqslant \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_H}{\sqrt{k}} \cdot \Phi_{\mathcal{I}'}(X)\right] \cdot \left(k^{12} \cdot \lg n\right)^{-1}.$$

Finally, by (13) and (14) we infer that

$$\left(1 - \frac{1}{k}\right)^{c_2 \Pi_{\mathcal{I}}(X) \lg \Gamma_{\mathcal{I}}} \leqslant \left(1 - \frac{1}{k}\right)^{c_2 \Pi_{\mathcal{I}'}(X) \lg \Gamma_{\mathcal{I}'}}. \tag{19}$$

By multiplying (17), (18), and (19) and applying the obtained bound in (16), we infer that

$$\mathbb{P}(X \subseteq A) \geqslant k^{-12} \cdot (\lg n)^{-1} \cdot \mathsf{LB}(n, \Pi_{\mathcal{I}}(X), \Gamma_{\mathcal{I}}, \Phi_{\mathcal{I}}(X)) \cdot k^{12} \cdot \lg n$$
$$= \mathsf{LB}(n, \Pi_{\mathcal{I}}(X), \Gamma_{\mathcal{I}}, \Phi_{\mathcal{I}}(X)).$$

This concludes the proof. ☐

Claim 32 ensures that the output of the algorithm has the required properties, whereas Claim 33 yields the sought lower bound on the success probability.

Subcase: A separator chain. In this case, the algorithm of Theorem 9 returned an (r, z)-separator chain $(C_1, \ldots, C_p)$ in $L = G''\langle R \cap V(G'')\rangle$, where $p = \lceil 120\sqrt{k} \lg k\rceil$ and $|C_i| \leqslant 2k$ for each $i \in [p]$. Obviously, by the definition of L we have that $(C_1, \ldots, C_p)$ is also an (r, z)-separator chain in G'', and no vertex of any C_i is a relay vertex. Recall that this means that all separators C_i are pairwise disjoint and $\mathsf{reach}(r, G''-C_i) \subseteq \mathsf{reach}(r, G''-C_j)$ whenever $1 \leqslant i < j \leqslant p$. By invariant (Inv.a), at most 3 first separators may include a light terminal, and hence after excluding them we are left with at least $\lceil 117\sqrt{k} \lg k\rceil$ separators without any light terminals. We restrict our attention to these separators. Thus, by slightly abusing the notation, from now on we work with an (r, z)-separation chain $(C_1, \ldots, C_p')$, where $p' = \lceil 117\sqrt{k} \lg k\rceil$ such that each C_i is disjoint with $T^{\mathsf{li}} \cup R$ and, in fact, $T^{\mathsf{li}} \subseteq \mathsf{reach}(r, G'' - C_1)$.

For $i \in [p']$, we define the following sets:

$$V_i^{\mathsf{in}} := \mathsf{reach}(r, G'' - C_i) \qquad \text{and} \qquad V_i^{\mathsf{out}} := V(G'') \setminus (C_i \cup V_i^{\mathsf{in}}).$$

Thus, $(V_i^{\mathsf{in}}, C_i, V_i^{\mathsf{out}})$ is a partition of $V(G'')$. Without loss of generality we can assume that each separator C_i is inclusion-wise minimal, which implies that each vertex of C_i has a neighbor in V_i^{in} and a neighbor in V_i^{out}.

We now prove that one of the separators C_i has the property that it splits X in a balanced way, relatively to the number of vertices of X it contains.

CLAIM 34. *There is an index $i \in [p']$ for which the following holds:*

$$10\sqrt{k} \cdot |X \cap C_i| \leqslant \min(|(X \cap V_i^{\mathsf{in}}) \setminus T^{\mathsf{li}}|, |(X \cap V_i^{\mathsf{out}}) \setminus T^{\mathsf{li}}|).$$

Proof. For $i \in [p']$, let

$$a_i := |(X \cap V_i^{\mathsf{in}}) \setminus T^{\mathsf{li}}|, \quad \text{and} \quad b_i := |(X \cap V_i^{\mathsf{out}}) \setminus T^{\mathsf{li}}|, \quad \text{and} \quad c_i := |X \cap C_i|.$$

Observe that since all light terminals are within $\mathsf{reach}(r, G'' - C_1)$, for each $i \in [p']$ the following hold:

$$a_i \geqslant \sum_{j<i} c_j; \tag{20}$$

$$b_i \geqslant \sum_{j>i} c_j. \tag{21}$$

Observe that each separator C_i has to contain a vertex of X, because X contains a nonrelay vertex at nr-distance at most 1 from z, and this vertex can be reached from r by a path that uses only relay vertices and vertices of X. We conclude that $c_i \geqslant 1$ for each $i \in [p']$. Consequently, $a_i \geqslant 1$ for each $i \geqslant 2$, and $b_i \geqslant 1$ for each $i \leqslant p' - 1$.

Supposing that the assertion stated in the claim does not hold, we have

$$c_i > \frac{\min(a_i, b_i)}{10\sqrt{k}} \qquad \text{for all } i \in [p']. \tag{22}$$

Obviously, the sequence $(a_i)_{i\in[p']}$ is nondecreasing and the sequence $(b_i)_{i\in[p']}$ is nonincreasing, Let i_0 be the smallest index such that $a_{i_0} > b_{i_0}$; possibly $i_0 = p' + 1$ if the condition $a_i \leqslant b_i$ is satisfied for all $i \in [p']$. We claim that in fact $i_0 \leqslant 53\sqrt{k}\lg k$; suppose otherwise. By assumption (22) and the definition of i_0, we have that $c_i > a_i/(10\sqrt{k})$ for all $i < i_0$. Therefore, by combining this with (20), we obtain that

$$a_i > \frac{1}{10\sqrt{k}} \sum_{j<i} a_j$$

for all $i < i_0$. Equivalently,

$$\sum_{j\leqslant i} a_j > \left(1 + \frac{1}{10\sqrt{k}}\right) \cdot \sum_{j<i} a_j.$$

Since $a_2 \geqslant 1$, we infer by a trivial induction that

$$\sum_{j<i} a_j \geqslant \left(1 + \frac{1}{10\sqrt{k}}\right)^{i-2}$$

for all $2 \leqslant i < i_0$. Therefore, we conclude that

$$\begin{aligned} a_{i_0-1} &> \frac{1}{10\sqrt{k}} \cdot \left(1 + \frac{1}{10\sqrt{k}}\right)^{53\sqrt{k}\lg k - 3} \geqslant \frac{1}{10\sqrt{k}} \cdot \left(1 + \frac{1}{10\sqrt{k}}\right)^{50\sqrt{k}\lg k} \\ &\geqslant \frac{1}{10\sqrt{k}} \cdot e^{5\lg k} > k. \end{aligned}$$

This is a contradiction with $|X| \leqslant k$. Hence, we have that indeed $i_0 \leqslant 53\sqrt{k}\lg k$.

By applying a symmetric reasoning for the last separators and numbers b_i, instead of the first and numbers a_i, we obtain that if i_1 is the largest index such that $a_{i_1} < b_{i_1}$, then $i_1 \geqslant 64\sqrt{k}\lg k$. However, this means that $i_0 < i_1$, which is a contradiction with the fact that sequences $(a_i)_{i\in[p']}$ and $(b_i)_{i\in[p']}$ are nonincreasing and nondecreasing, respectively. ∎

Observe that if an index i satisfies the property given by Claim 34, then $|X \cap C_i| \leqslant \sqrt{k}/10$. Indeed, otherwise we would have that $\min(|X \cap V_i^{\mathsf{in}}|, |X \cap V_i^{\mathsf{out}}|) > k$, which is a contradiction with $|X| \leqslant k$.

The algorithm performs random sampling as follows:

- First, it samples an index $i \in [p']$ uniformly at random and assumes that this index i satisfies the property given by Claim 34.
- Then, it samples an integer α between 1 and $\sqrt{k}/10$ and assumes that the sampled number α is equal to $|X \cap C_i|$.
- Finally, the algorithm samples a subset $Q \subseteq C_i$ of size α uniformly at random and assumes Q to be equal to $X \cap C_i$.

As $|C_i| \leqslant 2k$, we observe that the assumptions stated above are correct with probability at least

$$\frac{1}{p'} \cdot \frac{10}{\sqrt{k}} \cdot \frac{1}{\binom{|C_i|}{\alpha}} \geqslant \left(k^5 \cdot \binom{2k}{\alpha}\right)^{-1} \geqslant k^{-2\alpha-5},$$

where $\alpha = |X \cap C_i|$. Keeping this success probability assumption in mind, we proceed further with the supposition that the sampled objects are indeed as assumed.

The algorithm now defines two subinstances $\mathcal{I}_{\mathsf{out}}$ and $\mathcal{I}_{\mathsf{in}}$ as follows.

First, we define $\mathcal{I}_{\mathsf{out}} := (G_{\mathsf{out}}, r, T^{\mathsf{li}}_{\mathsf{out}}, T^{\mathsf{he}}_{\mathsf{out}}, R_{\mathsf{out}}, \lambda + \alpha)$; note that the guessed size of Q is added to the credit. Note that $\lambda + \alpha \leqslant \sqrt{k}/10 + \sqrt{k}/10$, so the instance on which we shall recurse will have credit at most $\sqrt{k}/5$, meaning that invariant (Inv.b) will be satisfied. Define G_{out} to be the graph constructed as follows: take G'', and contract the whole subgraph induced by $V_i^{\mathsf{in}} \cup (C_i \setminus Q)$ onto r. Observe that since $G''[V_i^{\mathsf{in}}]$ is connected by definition, and each vertex of C_i has a neighbor in V_i^{in}, the contracted subgraph is indeed connected.

The relay vertices are just inherited from the original instance: we put $R_{\mathsf{out}} = R \cap V(G_{\mathsf{out}})$. The sets of light and heavy terminals $T^{\mathsf{li}}_{\mathsf{out}}$ and $T^{\mathsf{he}}_{\mathsf{out}}$ are defined as follows. First, heavy terminals are inherited, but we remove all heavy terminals that reside in Q: we put $T^{\mathsf{he}}_{\mathsf{out}} = T^{\mathsf{he}} \cap (V(G_{\mathsf{out}}) \setminus Q)$. Second, as light terminals we put r plus the whole set Q: $T^{\mathsf{li}}_{\mathsf{out}} = \{r\} \cup Q$. Recall that $T^{\mathsf{li}} \subseteq V_1^{\mathsf{in}} \subseteq V_i^{\mathsf{in}}$, so all the light terminals of the original instance, apart from r, got contracted onto r during the construction of G^{out}; this is why we do not need to inherit any of them in $\mathcal{I}_{\mathsf{out}}$. Clearly, $T^{\mathsf{he}}_{\mathsf{out}}$ and $T^{\mathsf{li}}_{\mathsf{out}}$ defined in this manner are disjoint. Note that we have that $|T_{\mathsf{out}}| \leqslant |T| + |Q| \leqslant 16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + \lambda + |Q|$ and $|Q| = \alpha$, so we indeed have that $|T_{\mathsf{out}}| \leqslant 16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + (\lambda + \alpha)$; that is, invariant (Inv.c) is satisfied in the new instance. Invariant (Inv.a) is also satisfied, because all new light terminals are adjacent to the root r.

Finally, we define $X_{\mathsf{out}} := X \cap V(G_{\mathsf{out}})$. Since G_{out} was obtained from G'' only by contracting some vertices onto the root, it still holds that every vertex of X_{out} can be reached from r by a path traversing only relay vertices and vertices of X_{out}. Observe also that Claim 34 implies that at least $10\sqrt{k} \cdot \alpha$ vertices of X that are not light terminals are contained in V_i^{in}. These vertices do not remain in X^{out}, and hence

$$|X_{\mathsf{out}}| \leqslant |X| - 10\sqrt{k}\cdot\alpha \leqslant k - 10\sqrt{k}\cdot\lambda - 10\sqrt{k}\cdot\alpha = k - 10\sqrt{k}\cdot(\lambda+\alpha).$$

Therefore, we conclude that X_{out} is a pattern in the instance $\mathcal{I}_{\mathsf{out}}$.

By applying the algorithm recursively to the instance $\mathcal{I}_{\mathsf{out}}$, we obtain a subset of vertices A_{out} with the following properties:

- $A_{\mathsf{out}} \supseteq T^{\mathsf{li}}_{\mathsf{out}}$ and $G_{\mathsf{out}}[A_{\mathsf{out}}]$ admits a tree decomposition $\mathcal{T}_{\mathsf{out}}$ of width at most $24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ with $A_{\mathsf{out}} \cap T_{\mathsf{out}}$ contained in the root bag.
- Denoting the number of vertices of G_{out} by n_{out}, the probability that $X_{\mathsf{out}} \subseteq A_{\mathsf{out}}$ is at least $\mathsf{LB}(n_{\mathsf{out}}, \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}), \Gamma_{\mathcal{I}_{\mathsf{out}}}, \Phi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}))$.

Second, we define $\mathcal{I}_{\mathsf{in}} := (G_{\mathsf{in}}, r, T^{\mathsf{li}}{}_{\mathsf{in}}, T^{\mathsf{he}}{}_{\mathsf{in}}, R_{\mathsf{in}}, \lambda+\alpha)$; again the guessed size of Q is added to the credit. Note that, again, $\lambda+\alpha \leqslant \sqrt{k}/10+\sqrt{k}/10$, so the instance on which we shall recurse will have credit at most $\sqrt{k}/5$, meaning that invariant (Inv.b) will be satisfied. Graph G_{in} is constructed from G'' as follows. Inspect the connected components of the graph $G'' - (V_i^{\mathsf{in}} \cup Q)$. For each such component D, contract it onto a new vertex g_D that is declared to be a relay vertex. That is, we define R_{in} to be $(R \cap V_i^{\mathsf{in}}) \cup \{g_D \colon D \in \mathsf{cc}(G'' - (V_i^{\mathsf{in}} \cup Q))\}$.

Next, we define the terminal sets $T^{\mathsf{li}}{}_{\mathsf{in}}$, $T^{\mathsf{he}}{}_{\mathsf{in}}$. Recall that $T^{\mathsf{li}} \subseteq V_i^{\mathsf{in}}$, so all the original light terminals persist in the graph G_{in}. Hence, the light terminals are defined as simply inherited from the original instance: $T^{\mathsf{li}}{}_{\mathsf{in}} := T^{\mathsf{li}}$. For the heavy terminals, we take all the ones inherited from the original instance, plus we add all vertices of Q explicitly: $T^{\mathsf{he}}{}_{\mathsf{in}} := (T^{\mathsf{he}} \cap V(G_{\mathsf{in}})) \cup Q$. Note that $T^{\mathsf{li}}{}_{\mathsf{in}}$ and $T^{\mathsf{he}}{}_{\mathsf{in}}$ are disjoint, because there was no light terminal in Q; that is, $Q \cap T^{\mathsf{li}} = \emptyset$. Again, we have that $|T_{\mathsf{in}}| \leqslant |T| + |Q| \leqslant 16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + \lambda + |Q|$ and $|Q| = \alpha$, so we indeed have that $|T_{\mathsf{in}}| \leqslant 16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + (\lambda+\alpha)$; that is, invariant (Inv.c) is satisfied in the new instance. Invariant (Inv.a) is also satisfied, since all light terminals remain at the same nr-distance from r.

Finally, we take $X_{\mathsf{in}} := X \cap V(G_{\mathsf{in}})$. We observe that each vertex of X_{in} can be reached from r in G_{in} by a path that uses only relay vertices and vertices of X_{in}. Indeed, there is such a path in G'', and its parts that lie outside of $V(G_{\mathsf{in}})$ must be contained in the connected components of $G'' - (V_i^{\mathsf{in}} \cup Q)$, so they can be replaced by the traversal of the relay vertices into which these connected components are collapsed. Next, from Claim 34 we infer that

$$|X^{\mathsf{in}}| \leqslant |X| - 10\sqrt{k}\cdot\alpha \leqslant k - 10\sqrt{k}\cdot\lambda - 10\sqrt{k}\cdot\alpha = k - 10\sqrt{k}\cdot(\lambda+\alpha).$$

Hence, we conclude that X^{in} is a pattern in $\mathcal{I}_{\mathsf{in}}$.

Again, we apply the algorithm recursively to instance $\mathcal{I}_{\mathsf{in}}$, thus obtaining a subset of vertices A_{in} with the following properties:

- $A_{\mathsf{in}} \supseteq T^{\mathsf{li}}{}_{\mathsf{in}}$ and $G_{\mathsf{in}}[A_{\mathsf{in}}]$ admits a tree decomposition $\mathcal{T}_{\mathsf{in}}$ of width at most $24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ with $A_{\mathsf{in}} \cap T_{\mathsf{in}}$ contained in the root bag.
- The probability that $X_{\mathsf{in}} \subseteq A_{\mathsf{in}}$ is at least $\mathsf{LB}(n_{\mathsf{in}}, \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}), \Gamma_{\mathcal{I}_{\mathsf{in}}}, \Phi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}))$, where n_{in} is the number of vertices in G_{in}.

Observe that the sets of nonrelay vertices of G_{out} and G_{in} are contained in the vertex set of G'', and hence we can treat A_{out} and A_{in} also as subsets of nonrelay vertices of G''. Hence, let us define $A := (A_{\mathsf{out}} \setminus Q) \cup A_{\mathsf{in}}$ and declare that the algorithm returns A as the answer. Note that here, as in section 5.3, we formally allow the instance $\mathcal{I}_{\mathsf{in}}$ to exclude the vertices of Q from A_{in}, since they are heavy terminals there.

We now verify that A has the required properties. First, since we have that ${T^{\mathsf{li}}}_{\mathsf{in}} = T^{\mathsf{li}}$, then $A \supseteq A_{\mathsf{in}} \supseteq {T^{\mathsf{li}}}_{\mathsf{in}} = T^{\mathsf{li}}$, so A indeed covers all light terminals. We now check that $G[A]$ admits a suitable tree decomposition.

Claim 35. *The subgraph $G[A]$ admits a tree decomposition of width at most $24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ with $A \cap T$ contained in the root bag.*

Proof. Construct the root node and associate with it the bag $(T \cup Q) \cap A$. Since $|T| \leqslant 16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + \lambda$ and $\lambda, |Q| \leqslant \sqrt{k}/10$, it follows that this bag has size at most $16015\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$. Let us restrict decomposition $\mathcal{T}_{\mathsf{out}}$ to the vertex set $A \cap A_{\mathsf{out}}$; that is, remove all vertices of $A_{\mathsf{out}} \setminus A$ from all bags of $\mathcal{T}_{\mathsf{out}}$, thus obtaining a tree decomposition ${\mathcal{T}'}_{\mathsf{out}}$ of $G_{\mathsf{out}}[A_{\mathsf{out}} \cap A]$. We have that $A_{\mathsf{in}} \subseteq A$, so there is no need of restricting decomposition $\mathcal{T}_{\mathsf{in}}$. Finally, attach decompositions $\mathcal{T}_{\mathsf{in}}$ and ${\mathcal{T}'}_{\mathsf{out}}$ as children of the root bag. It can be easily verified that in this manner we obtain a tree decomposition of $G[A]$, and its width is clearly at most $24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$. Finally, the root bag contains $A \cap T$ by its definition. ☐

We are left with estimating the success probability. Before we proceed with the final calculation, let us analyze each of the potentials. Note that graphs G_{in} and G_{out} intersect only at $Q \cup \{r\}$, and each vertex of Q is a heavy terminal in $\mathcal{I}_{\mathsf{in}}$ and a light terminal in $\mathcal{I}_{\mathsf{out}}$. This observation will be crucial in the forthcoming analysis. Recall also that Q, as a subset of C_i, contains no original light terminal, i.e., $Q \cap T^{\mathsf{li}} = \emptyset$.

First, in $\mathcal{I}_{\mathsf{out}}$ we contracted all vertices of $V_i^{\mathsf{in}} \cup (C_i \setminus Q)$, and in $\mathcal{I}_{\mathsf{in}}$ we contracted all vertices of $V_i^{\mathsf{out}} \cup (C_i \setminus Q)$. Among the vertices shared by the instances, being $\{r\} \cup Q$, r is a light terminal in both instances, whereas the vertices of Q are heavy terminals only in $\mathcal{I}_{\mathsf{in}}$. From this it immediately follows that

$$\Gamma_{\mathcal{I}} \geqslant \Gamma_{\mathcal{I}_{\mathsf{out}}} + \Gamma_{\mathcal{I}_{\mathsf{in}}}. \tag{23}$$

Observe that both G_{out} and G_{in} are constructed from G by means of edge contractions only, which can only decrease the nr-distances from r. Hence, a vertex of X that was close in the original instance $\mathcal{I}$ remains close in the instance $\mathcal{I}_{\mathsf{in}}$ or $\mathcal{I}_{\mathsf{out}}$ to which it belongs. The vertices of Q are adjacent to the root in $\mathcal{I}_{\mathsf{out}}$, and hence none of them can be a far vertex of X_{out}. Hence it follows that

$$\Phi_{\mathcal{I}}(X) \geqslant \Phi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) + \Phi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}). \tag{24}$$

Finally, the vertices of Q—shared among the instances—are declared light terminals in $\mathcal{I}_{\mathsf{in}}$, and hence the same analysis yields that

$$\Pi_{\mathcal{I}}(X) \geqslant \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) + \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}). \tag{25}$$

We now perform a finer analysis of the behavior of potential Π. For this, we use Claim 34 as follows.

Claim 36. *The following holds:*

$$\Pi_{\mathcal{I}}(X)\lg\Pi_{\mathcal{I}}(X) \geqslant \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}})\lg\Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) + \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}})\lg\Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) + 10\sqrt{k}\cdot\alpha. \tag{26}$$

Proof. Observe that

$$\Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) = |(X \cap V_i^{\mathsf{out}}) \setminus T^{\mathsf{li}}|;$$
$$\Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) = |(X \cap V_i^{\mathsf{in}}) \setminus T^{\mathsf{li}}| + |X \cap C_i|.$$

Thus we have

$$\begin{aligned}\Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) + \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) &= |(X \cap V_i^{\mathsf{out}}) \setminus T^{\mathsf{li}}| + |(X \cap V_i^{\mathsf{in}}) \setminus T^{\mathsf{li}}| + |X \cap C_i| \\ &= |X \setminus T^{\mathsf{li}}| = \Pi_{\mathcal{I}}(X).\end{aligned} \tag{27}$$

Suppose first that $\Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) \leqslant \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}})$; the second case is symmetric. Combining this with (27) yields the following:

$$\Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) \leqslant \Pi_{\mathcal{I}}(X)/2 \qquad \text{and} \qquad \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) \leqslant \Pi_{\mathcal{I}}(X).$$

By Claim 34, we infer that

$$10\sqrt{k} \cdot \alpha \leqslant |(X \cap V_i^{\mathsf{out}}) \setminus T^{\mathsf{li}}| \leqslant \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}).$$

Putting all these together, we observe that

$$\begin{aligned}\Pi_{\mathcal{I}}(X) \lg \Pi_{\mathcal{I}}(X) &= \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) \lg \Pi_{\mathcal{I}}(X) + \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) \lg \Pi_{\mathcal{I}}(X) \\ &\geqslant \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) \cdot (1 + \lg \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}})) + \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) \lg \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) \\ &\geqslant \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) \lg \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) + \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) \lg \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) + \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) \\ &\geqslant \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) \lg \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) + \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) \lg \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) + 10\sqrt{k} \cdot \alpha.\end{aligned}$$

This is exactly the claimed inequality. As mentioned before, the case when $\Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) \geqslant \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}})$ is symmetric. ∎

Finally, we can proceed with the final success probability analysis. For this, we take any $c_1 \geqslant 2$.

CLAIM 37. *Assume $c_1 \geqslant 2$. Supposing $X \cap W_{\mathsf{isl}} \neq \emptyset$ and the subroutine of Theorem 9 returned a separator chain, the algorithm outputs a set A with $X \subseteq A$ with probability at least $\mathsf{LB}(n, \Pi(X), \Gamma, \Phi(X))$. This includes the $(1 - 1/k)$ probability of success of the preliminary clustering step, the $1/k$ probability that the algorithm correctly assumes that $X \cap W_{\mathsf{isl}} \neq \emptyset$, the k^{-7} probability of correctly choosing the island C that intersects the pattern, the $(10k^2 \lg n)^{-1}$ probability of choosing the right distance d, and $k^{-2\alpha-5}$ probability of correctly sampling the i, α, and set Q.*

Proof. We denote by n_{in} and n_{out} the numbers of vertices in G_{in} and G_{out}, respectively; note that $n_{\mathsf{in}}, n_{\mathsf{out}} \leqslant n$. From the probability of the success of recursive calls, we infer that

$$\begin{aligned}\mathbb{P}(X \subseteq A) &\geqslant \left(1 - \frac{1}{k}\right) \cdot k^{-8} \cdot (10k^2 \lg n)^{-1} \cdot k^{-2\alpha-5} \cdot \\ &\qquad \mathsf{LB}(n_{\mathsf{out}}, \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}), \Gamma_{\mathcal{I}_{\mathsf{out}}}, \Phi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}})) \cdot \mathsf{LB}(n_{\mathsf{in}}, \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}), \Gamma_{\mathcal{I}_{\mathsf{in}}}, \Phi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}})) \\ &\geqslant k^{-2\alpha-17} \cdot (\lg n)^{-1} \cdot \\ &\qquad \mathsf{LB}(n_{\mathsf{out}}, \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}), \Gamma_{\mathcal{I}_{\mathsf{out}}}, \Phi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}})) \cdot \mathsf{LB}(n_{\mathsf{in}}, \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}), \Gamma_{\mathcal{I}_{\mathsf{in}}}, \Phi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}})).\end{aligned} \tag{28}$$

From (23) and (25) we infer that

$$\left(1 - \frac{1}{k}\right)^{c_2 \Pi_{\mathcal{I}}(X) \lg \Gamma_{\mathcal{I}}} \leqslant \left(1 - \frac{1}{k}\right)^{c_2 \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) \lg \Gamma_{\mathcal{I}_{\mathsf{out}}}} \cdot \left(1 - \frac{1}{k}\right)^{c_2 \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) \lg \Gamma_{\mathcal{I}_{\mathsf{in}}}}. \tag{29}$$

Similarly, from (24) and the fact that $n_{\mathsf{in}}, n_{\mathsf{out}} \leqslant n$, we infer that

$$\exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n}{\sqrt{k}} \cdot \Phi_{\mathcal{I}}(X)\right] \leqslant \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_{\mathsf{out}}}{\sqrt{k}} \cdot \Phi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}})\right] \cdot \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_{\mathsf{in}}}{\sqrt{k}} \cdot \Phi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}})\right]. \tag{30}$$

Finally, from Claim 36 we have that

$$\begin{aligned}
&\exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n}{\sqrt{k}} \cdot \Pi_{\mathcal{I}}(X) \lg \Pi_{\mathcal{I}}(X)\right] \\
&\leqslant \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_{\mathsf{out}}}{\sqrt{k}} \cdot \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}) \lg \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}})\right] \cdot \\
&\quad \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n_{\mathsf{in}}}{\sqrt{k}} \cdot \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) \lg \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}})\right] \cdot \\
&\quad \exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n}{\sqrt{k}} \cdot 10\alpha\sqrt{k}\right].
\end{aligned} \tag{31}$$

Let us analyze the last factor of the right-hand side of (31), keeping in mind that $c_1 \geqslant 2$.

$$\begin{aligned}
\exp\left[-c_1 \cdot \frac{\lg k + \lg\lg n}{\sqrt{k}} \cdot 10\alpha\sqrt{k}\right] &= \exp\left[-c_1 \cdot 10\alpha \cdot \lg k - c_1 \cdot 10\alpha \cdot \lg\lg n\right] \\
&\leqslant k^{-20\alpha} \cdot (\lg n)^{-1} \leqslant k^{-2\alpha-17} \cdot (\lg n)^{-1}.
\end{aligned} \tag{32}$$

Finally, by plugging (29), (30), (31), and (32) into (28) with LB function expanded, and recognizing the expression $\mathsf{LB}(n, \Pi(X), \Gamma, \Phi(X))$, we obtain

$$\mathbb{P}(X \subseteq A) \geqslant \mathsf{LB}(n, \Pi(X), \Gamma, \Phi(X)),$$

which is exactly what we needed to prove. ◻

Claim 35 ensures that the output of the algorithm has the required properties, whereas Claim 37 yields the sought lower bound on the success probability.

6. Extensions.

In this section we develop the following extension of Theorem 1 for graph classes excluding a fixed minor, at the cost of a bound on the maximum degree of the pattern. By a *proper minor-closed graph class* we mean a graph class that is minor-closed and does not contain all graphs.

THEOREM 38. *Let $\mathcal{C}$ be a proper minor-closed graph class, and let Δ be a fixed constant. Then there exists a randomized polynomial-time algorithm that, given an n-vertex graph G from $\mathcal{C}$ and an integer k, samples a vertex subset $A \subseteq V(G)$ with the following properties:*

- (P1) *The induced subgraph $G[A]$ has treewidth $\mathcal{O}(\sqrt{k}\log k)$.*
- (P2) *For every vertex subset $X \subseteq V(G)$ with $|X| \leqslant k$ such that $G[X]$ is connected and has a spanning tree of maximum degree Δ, the probability that X is covered by A, that is $X \subseteq A$, is at least $(2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot n^{\mathcal{O}(1)})^{-1}$.*

To see why the above assumption seems necessary with our techniques, let us look at the following example. Let G be a graph that contains a universal vertex v_0 (i.e., adjacent to all vertices of $V(G) \setminus \{v_0\}$) such that $G - v_0$ is planar. It is easy to see that, since $G - v_0$ is K_5-minor-free, we have that G is K_6-minor-free. Let H be a connected pattern in G: a connected subgraph on k vertices. If H contains v_0, then $H - v_0$ is a not necessarily connected pattern (subgraph) of $G - v_0$. Hence, finding a connected k-vertex pattern in G boils down to finding a not necessarily connected $(k-1)$-vertex pattern in $G - v_0$. However, if we bound the maximum degree of the pattern, the $(k-1)$-vertex pattern $H - v_0$ in the graph $G - v_0$ has bounded number of connected components, making the situation much more similar to the connected planar (or apex-minor-free) case.

We do not know how to handle arbitrary disconnected patterns (subgraphs) with our techniques. As we show in this section, we are able to do it in a limited fashion; namely, we can handle up to roughly $\sqrt{k}/\lg k$ connected components without increasing the asymptotic bound in the exponential factor in the success probability. The proof of the following generalization of Theorem 1 is described in section 6.1.

THEOREM 39. *Let $\mathcal{C}$ be a class of graphs that exclude a fixed apex graph as a minor. Then there exists a randomized polynomial-time algorithm that, given an n-vertex graph G from $\mathcal{C}$ and an integer k, samples a vertex subset $A \subseteq V(G)$ with the following properties:*

- (P1) *The induced subgraph $G[A]$ has treewidth $\mathcal{O}(\sqrt{k}\log k)$.*
- (P2) *For every vertex subset $X \subseteq V(G)$ with $|X| \leqslant k$ such that $G[X]$ has at most $\mathcal{O}(\sqrt{k}/\log k)$ connected components, the probability that X is covered by A, that is $X \subseteq A$, is at least $(2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot n^{\mathcal{O}(1)})^{-1}$.*

After proving Theorem 39, in section 6.2 we show how to use this extension for a bounded number of connected components in order to handle connected patterns in graph classes excluding a fixed minor. To this end, we use the Robertson–Seymour decomposition theorem that provides a tree decomposition for any graph that exclude a fixed minor. Roughly speaking, in this decomposition every bag corresponds to a graph almost embeddable into a fixed surface, and every adhesion (intersection of neighboring bags) has bounded size. By a result of Grohe [26], one can delete a bounded number of vertices from an almost embeddable graph to get an apex-minor-free graph. If the pattern we are looking for is connected and has bounded degree, deleting a bounded number of vertices can split it only into a bounded number of connected components. Thus, the algorithm for graph classes excluding a fixed minor boils down to an application of either Theorem 39 or a simple Baker-style argument to every bag, after turning it into an apex-minor-free graph.

6.1. Extension to bounded number of components. In this section we prove Theorem 39; that is, we extend Theorem 1 to handle a bounded number of connected components of the pattern. We describe the reasoning as a series of modifications to the proof of Theorem 1 from section 5.

As in section 5, in a recursive step we are given an instance $\mathcal{I}$ consisting of a minor G of the input graph G_0, a root $r \in V(G)$, two disjoint sets of light and heavy terminals $T^{\mathsf{li}}, T^{\mathsf{he}} \subseteq V(G)$ with $r \in T^{\mathsf{li}}$ (we denote $T = T^{\mathsf{li}} \cup T^{\mathsf{he}}$ a set $R \subseteq V(G) \setminus T$ of relay vertices representing connectivity in other parts of the input graph), and credit λ. The nonrelay-distance measure $\text{nr-dist}_G(\cdot,\cdot)$ is defined in the same way: the cost of traversing a relay vertex is 0. We maintain the same invariants regarding terminals: every light terminal is within nr-distance at most 3 from the root, and the number of terminals is bounded by $16014\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k + \lambda$.

The first significant difference concerns the notion of a pattern, as we need to generalize this concept. We start with the following definition.

DEFINITION 40. *Let $X \subseteq V(G)\setminus R$ be a set of vertices. Two vertices $x, y \in X$ are* connected *if they belong to the same connected component of $G[X \cup R]$. A* component *of the set X is an equivalence class in the relation of being connected (i.e., a set of vertices from X from a connected component of $G[X \cup R]$ that contains at least one vertex of X). A component Y is* rooted *if it contains a vertex within nr-distance at most* 3 *from the root and free otherwise.*

For a set $X \subseteq V(G) \setminus R$ in an instance $\mathcal{I}$, we introduce the following *component potential* as the fourth potential:

$$\textit{Component potential} \quad \Lambda_{\mathcal{I}}(X) := \text{number of free components of } X.$$

We can now formally define a pattern. A set $X \subseteq V(G) \setminus R$ is a *pattern* in $\mathcal{I}$ if $r \in X$ and

$$|X| \leqslant k - 10\sqrt{k} \cdot \lambda - 486\sqrt{k} \lg k \cdot \Lambda_{\mathcal{I}}(X).$$

That is, we drop the assumption of the connectivity of X (possibly with the help of some relay vertices), but every free component imposes a penalty on the allowed size of the pattern. Note that every pattern contains at least one rooted component (the one containing the root r) and an arbitrary number of free components.

6.1.1. Potentials. Let us now proceed to the description of the potentials. Apart from introducing the component potential, we extend the set of *far* vertices: every vertex in a free component is far, regardless of its nr-distance from the root r.

$$\mathsf{Far}_{\mathcal{I}}(X) := \{u \in X \ : \ \text{nr-dist}_G(u, r) > 1000\sqrt{k} \lg k \quad \text{or} \quad u \text{ is in a free component}\}.$$

As before, every vertex of the pattern that is not far is called *close*.

Intuitively, every free component of the pattern decreases the success probability of the algorithm by a factor inverse-quasipolynomial in k and $\lg n$. Formally, we define

$$\begin{aligned}&\widehat{\mathsf{LB}}(n, \Pi_{\mathcal{I}}(X), \Gamma_{\mathcal{I}}, \Phi_{\mathcal{I}}(X), \Lambda_{\mathcal{I}}(X)) := \\ &\mathsf{LB}(n, \Pi_{\mathcal{I}}(X), \Gamma_{\mathcal{I}}, \Phi_{\mathcal{I}}(X)) \cdot \exp\left[-c_3 \cdot \Lambda_{\mathcal{I}}(X) \cdot \left(\lg^2 k(\lg k + \lg\lg n) + \frac{\lg n \lg k}{\sqrt{k}}\right)\right]\end{aligned}$$

for some positive constant c_3 that will be determined later. Here, again, n denotes the total number of vertices of the graph, and we omit the subscript $\mathcal{I}$ whenever the instance is clear from the context.

Our goal is to sample a subset of nonrelay vertices $A \subseteq V(G)\setminus R$ with the following properties:

(1) It holds that $T^{\mathsf{li}} \subseteq A$, and the graph $G[A]$ admits a tree decomposition of width at most $24022\bar{\alpha}(\mathcal{C})\sqrt{k} \lg k$, where $T \cap A$ is contained in the root bag.

(2) For every pattern X in instance $\mathcal{I}$, we have

$$\mathbb{P}(X \subseteq A) \geqslant \widehat{\mathsf{LB}}(n, \Pi(X), \Gamma, \Phi(X), \Lambda(X)). \tag{33}$$

6.1.2. Operations on the instance. One of the crucial properties of the algorithm of section 5 is that it modifies the input graph in a limited fashion. Namely, every subinstance is created by means of the following operations:

(1) Edge contraction. Furthermore, if one of the contracted vertices is a relay vertex, the new vertex remains a relay vertex or the contraction is made onto the root.

(2) Other modifications such as vertex/edge deletion/addition, but only involving vertices within nr-distance larger than $2000\sqrt{k}\lg k$ from r, and not involving vertices in the pattern X or relay vertices essential for the connectivity relation within the pattern (assuming that the algorithm made correct random choices).

The analysis of section 5 used the above properties to ensure that the algorithm never turns a close vertex into a far vertex, assuming that the algorithm makes correct random choices. Here, we observe that neither of the above modifications can create a new component. Furthermore, a component that is rooted remains rooted, and a vertex belonging to a rooted component remains in a rooted component. As a result, these modifications cannot turn a close vertex into a far vertex under the new definition of the far vertices, nor can they create a new free component. In particular, whenever we construct a pattern in a subinstance by projecting the original pattern in the natural way, the projection remains a pattern in the new instance. This is because the $486\sqrt{k}\lg k \cdot \Lambda(X)$ penalty in the upper bound on the size of the pattern does not increase.

6.1.3. Solving the general problem. First, note that we can make the same assumptions (Inv.d)–(Inv.g) as in section 5, as the reasoning of Lemma 20 still applies.

The general structure and the main steps of the algorithm are the same as in section 5: we define the margin M to be the set of vertices of G within nr-distance at most $2000\sqrt{k}\lg k$ from the root r and apply the clustering procedure to the graph $(G-M)\langle R \setminus M\rangle$. Note that the clustering procedure does not use the assumption of the connectivity of the pattern. Thus, we can assume that every island—every connected component of $G-M$—has radius at most $9k^2 \lg n$ (where relay vertices are traversed for free), at the cost of a $(1-1/k)$ multiplicative factor in the success probability. By slightly abusing the notation, we redefine G to be the graph obtained from the clustering procedure; this graph was named G_{cl} in section 5. We trim the sets of terminals and of relay vertices as in section 5, obtaining sets $T^{\mathsf{li}}_{\mathsf{cl}}, T^{\mathsf{he}}_{\mathsf{cl}}, R_{\mathsf{cl}}$, respectively.

By the same arguments as in section 5, by locally bounded treewidth we obtain sets W_{nrm} and W_{isl} with the following properties:

(1) W_{nrm} consists of at most $8007\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ vertices of M, and $r \in W_{\mathsf{nrm}}$;
(2) W_{isl} consists of the union of vertex sets of at most $8007\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ islands of $G-M$;
(3) every connected component D of $G_{\mathsf{cl}} - (W_{\mathsf{nrm}} \cup W_{\mathsf{isl}})$ contains at most $|T_{\mathsf{cl}}|/2$ terminals and at most $|V(G_{\mathsf{cl}}) \setminus (T^{\mathsf{li}}_{\mathsf{cl}} \cup R_{\mathsf{cl}})|/2$ vertices that are neither light terminals nor relay vertices.

As before, we randomly select a branch we pursue: with probability $(1-1/k)$ we assume that the pattern is disjoint with W_{isl}, and with the remaining probability $1/k$ we assume otherwise. Thus, we have two cases: when W_{isl} is assumed to be disjoint with the pattern, and when we suppose that W_{isl} intersects the pattern.

6.1.4. The case when W_{isl} is disjoint with the pattern. The crux in this case is to observe that nothing new happens, mostly because the argumentation of section 5.3 does not rely on the connectivity of the pattern. That is, we argue that the algorithm as described in section 5 works also in our setting.

Recall that in this case we first delete W_{isl} from G_{cl}; let the obtained graph be named G', as in section 5. Then recurse into instances $\mathcal{I}_D$ created for every connected component D of $G_{\mathsf{cl}} - (W_{\mathsf{isl}} \cup W_{\mathsf{nrm}}) = G' - W_{\mathsf{nrm}}$, defined as in section 5. In the instance

$\mathcal{I}_D$ we look for pattern $X_D := X \cap N_D$, where $N_D := N_{G'}[D] \cup \{r\}$. We denote by CC the set of connected components of $G' - W_{\mathsf{nrm}}$.

We now need to analyze the behavior of the free components in the recursion. We start with the following observation that follows directly from the discussion of section 6.1.2.

CLAIM 41. *Let Y be a component of X in $\mathcal{I}$, and let $D \in \mathsf{CC}$ be such that $Y \cap N_D \neq \emptyset$. Then $Y \cap N_D$ is contained in a single component of X_D in $\mathcal{I}_D$.*

Second, we observe that the rooted components cannot give rise to any new free components.

CLAIM 42. *Let Y be a rooted component of X in $\mathcal{I}$, and let $D \in \mathsf{CC}$ be such that $Y \cap N_D \neq \emptyset$. Then $Y \cap N_D$ is contained in a rooted component of X_D in $\mathcal{I}_D$.*

Proof. Let w be a vertex of Y that is within nr-distance at most 3 from the root r in G_{cl} (equivalently, in G'). If $w \in N_D$, then we are done; hence assume otherwise. By the definition of a component, there exists a path P in $G'[Y \cup R]$ between w and some vertex $v \in Y \cap N_D$ such that no vertex of $P - \{v\}$ belongs to N_D, except for possibly a neighbor v' of v on P if v' is a relay vertex in $N_{G'}(D)$. Consequently, in the process of construction of $\mathcal{I}_D$, the path $P - \{v\}$ is contracted either onto the root or onto a relay vertex. As the nr-distance between w and r is at most three in G', and relay vertices are traversed for free in our nr-distance measure, we have that v is within nr-distance at most 3 from the root in $\mathcal{I}_D$. Consequently, $Y \cap N_D$ is contained in a rooted component of X_D in $\mathcal{I}_D$. ∎

Third, we observe that a free component cannot split into multiple free components.

CLAIM 43. *Let Y be a free component of X in $\mathcal{I}$. Then there exists a component $D_0 \in \mathsf{CC}$ such that for every $D \in \mathsf{CC}$ such that $D \neq D_0$ and $Y \cap N_D \neq \emptyset$ the set $Y \cap N_D$ is contained in a rooted component of X_D in $\mathcal{I}_D$.*

Proof. We say that a component $D \in \mathsf{CC}$ is *touched* if $Y \cap N_D \neq \emptyset$. We consider all paths in G' between the root r and a vertex $w \in N_{G'}[D]$ for a touched component D and pick Q_0 to be a shortest such path. Let $w \in N_{G'}[D_0]$ be the second endpoint of Q_0, where D_0 is a touched component. By the minimality of Q_0, no vertex of $Q_0 - \{w\}$ belongs to $N_{G'}[D]$ for a touched component D. Let Q_1 be a shortest path between w and a vertex $v \in Y \cap N_{D_0}$ with all internal vertices in D_0; such a path exists by the connectivity of $G'[D_0]$, and by the minimality of Q_1 no vertex of $Q_1 - \{v\}$ belongs to Y.

Consider a touched component $D \in \mathsf{CC}$ different than D_0. By the definition of a component, there exists a path Q_2 in $G'[Y \cup R]$ between v and a vertex $u \in Y \cap N_D$ such that no vertex of $Q_2 - \{u\}$ belongs to N_D, except for possibly a neighbor u' of u on Q_2 that is a relay vertex in $N_{G'}(D)$. Observe that from the walk being the concatenation of the paths Q_0, Q_1, and Q_2 only the root r and the vertices w, u, and u' may potentially belong to N_D. Consequently, in $\mathcal{I}_D$, the vertex u is within nr-distance at most 2 from the root (recall that u' is a relay vertex if it belongs to N_D). We infer that $Y \cap N_D$ is contained in a rooted component of X_D in $\mathcal{I}_D$. ∎

Claims 41–43 justify the following.

CLAIM 44. *The following holds:*

$$\Lambda_{\mathcal{I}}(X) \geqslant \sum_{D \in \mathsf{CC}} \Lambda_{\mathcal{I}_D}(X_D). \tag{34}$$

Furthermore, Claims 41–43 ensure that a close vertex of X in $\mathcal{I}$ cannot become a far vertex in any of the instances $\mathcal{I}_D$. Consequently, the potential analysis of Claim 28 holds also in our case.

Using Claim 44, the following claim follows along the same lines as Claim 30, finishing the analysis of this subcase.

CLAIM 45. *Supposing* $X \cap W_{\mathsf{isl}} = \emptyset$, *the algorithm outputs a set* A *with* $X \subseteq A$ *with probability at least* $\widehat{\mathsf{LB}}(n, \Pi_{\mathcal{I}}(X), \Gamma_{\mathcal{I}}, \Phi_{\mathcal{I}}(X), \Lambda_{\mathcal{I}}(X))$. *This includes the* $(1-1/k)$ *probability of success of the preliminary clustering step and the* $(1-1/k)$ *probability that the algorithm correctly assumes that* $X \cap W_{\mathsf{isl}} = \emptyset$.

We note that the bound on the treewidth of $G[A]$ follows in exactly the same manner as in section 5.

6.1.5. The case when W_{isl} intersects the pattern. Just as in section 5, in the second case we

(a) guess (by sampling at random) an island C with $V(C) \subseteq W_{\mathsf{isl}}$ such that C intersects the pattern;
(b) take z to be a nonrelay vertex of C that is at nr-distance at most $9k^2 \lg n$ from all vertices of C within C;
(c) guess (by sampling at random) the distance d from z to the pattern within the island C; and
(d) contract the vertices of C within nr-distance less than d from z onto z.

In step (d), we perform the same distinction as in section 5 between nonrelay vertices (for which we use nr-distance less than $\max(d,1)$) and relay vertices (for which we use nr-distance less than $\max(d,1)-1$).

As a result, by incurring a multiplicative factor

$$\frac{1}{k} \cdot |W_{\mathsf{isl}}|^{-1} \cdot (10k^2 \lg n)^{-1} \geqslant k^{-11} (\lg n)^{-1}$$

in the success probability we can assume that we have identified a vertex $z \notin R$ satisfying $\text{nr-dist}_G(r,z) > 2000\sqrt{k}\lg k$ such that there exists a vertex of the pattern within distance at most 1 from z. We let G'' denote the graph after the modifications explained above. As before, we now apply the duality theorem (Theorem 9) to the graph $G''\langle R \cap V(G'')\rangle$, pair of vertices $(s,t) := (r,z)$, and parameters

$$p := \lceil 120\sqrt{k}\lg k \rceil \qquad \text{and} \qquad q := k.$$

The further behavior of the algorithm, as well as its analysis, depends on the output of the duality theorem. Thus, we need to consider two subcases: the duality theorem returns either a family of paths or a separator chain.

Subcase: A sequence of paths. Following the argumentation of section 5, in this section we are working with the following objects:

- a vertex $z \in V(G'') \setminus R$ with $\text{nr-dist}_{G''}(z,r) > 2000\sqrt{k}\lg k$, such that some vertex of X is within nr-distance at most 1 from z, and
- a sequence $P_1, P_2, \ldots, P_k$ of (r,z)-paths in G'', such that for every $i \in [k]$ the set $V(P_i)$ can be partitioned as

$$V(P_i) = \{r,z\} \uplus (V(P_i) \cap R) \uplus \mathsf{Pub}(P_i) \uplus \mathsf{Prv}(P_i),$$

where the sets $\mathsf{Prv}(P_i)$ are pairwise disjoint and $|\mathsf{Pub}(P_i)| \leqslant 480\sqrt{k}\lg k$.

The success probability so far in this case is at least $k^{-11}(\lg n)^{-1}$.

As in section 5, we randomly pick an index $i \in [k]$ and assume further that $X \cap \mathsf{Prv}(P_i) = \emptyset$. Such an index i exists, because $|X| \leqslant k$, the sets $\mathsf{Prv}(P_i)$ are pairwise disjoint, and $r \in X$ but $r \notin \mathsf{Prv}(P_i)$ for every i. Hence, the success probability of this step is at least $1/k$.

We reduce P_i in the same way as in section 5. Let v_0 be the last light terminal on P_i (it exists, as r is a light terminal), let P' be the suffix of P_i from v_0 to z, and let $v_0, v_1, \ldots, v_\ell = z$ be the vertices of $(\mathsf{Pub}(P_i) \cap V(P')) \cup \{v_0, z\}$ in the order of their appearance on P'. For every $0 \leqslant j < \ell$, we inspect the segment of P' between v_j and v_{j+1}. If this segment contains some relay vertex g_j, then contract it entirely onto g_j; if there is more than one relay vertex, choose an arbitrary one as g_j. Otherwise, if there are no relay vertices on the segment, contract the whole segment onto vertex v_j.

Let H be the resulting graph, and construct the instance $\mathcal{I}'$ as in section 5. We have $\ell \leqslant |\mathsf{Pub}(P_i)| + 1 \leqslant 485\sqrt{k}\lg k$ and $\text{nr-dist}_G(r, v_0) \leqslant 3$, and thus $\text{nr-dist}_H(r, z) \leqslant 488\sqrt{k}\lg k$.

However, the proof of Claim 31 (relation between the sets of far vertices in $\mathcal{I}$ and $\mathcal{I}'$) fails if the vertex of the pattern within distance at most 1 from z belongs to a free component. Namely, in section 5 we argued that a significant number of far vertices of the pattern become close after the contraction, but now this argument does not apply anymore if they all reside in a free component—and thus are considered far automatically, no matter what is their nr-distance from the root. The crux here is that if this is the case, then we can turn the free component containing the said vertices into a rooted one by adding $\{v_0, v_1, \ldots, v_\ell\}$ to the pattern X, thus providing a gain in the potential $\Lambda(X)$. Let $X' := X \cup \{v_0, v_1, \ldots, v_\ell\}$.

CLAIM 46. *We have* $\mathsf{Far}_{\mathcal{I}'}(X) \subseteq \mathsf{Far}_{\mathcal{I}}(X)$ *and* $\Lambda_{\mathcal{I}'}(X) \leqslant \Lambda_{\mathcal{I}}(X)$. *Furthermore, one of the following holds:*

- $|\mathsf{Far}_{\mathcal{I}}(X) \setminus \mathsf{Far}_{\mathcal{I}'}(X)| \geqslant 511\sqrt{k}\lg k$, *or*
- X' *is a pattern in* $\mathcal{I}'$ *and* $\Lambda_{\mathcal{I}'}(X') < \Lambda_{\mathcal{I}}(X)$.

Proof. The first part of the claim follows directly from the discussion of section 6.1.2 and the fact that H is created from G'' by means of edge contractions, in the same manner as in section 5. For the second part, let $v \in X$ be a vertex within distance at most 1 from z in G'' (possibly $v = z$).

If v belongs to a rooted component of X in $\mathcal{I}$, the analysis of Claim 31 remains valid. That is, $G''[X \cup R]$ contains a path P from v to a vertex w that is within nr-distance at most 3 from the root, and the first $511\sqrt{k}\lg k$ vertices of X of this path belong to $\mathsf{Far}_{\mathcal{I}}(X)$. These vertices become close in H, as $\text{nr-dist}_H(r, v) \leqslant 488\sqrt{k}\lg k$.

Hence, we are left with the case when v belongs to some free component Y of X in $\mathcal{I}$. The crucial observation is that in H the vertex v belongs to a rooted component of X', as $v_0 \in T^{\mathsf{li}}$ (and thus is within nr-distance at most 3 from the root) and $v_0, v_1, \ldots, v_\ell$ is a path in H. By the discussion in section 6.1.2, no new free component is created in $\mathcal{I}'$, while Y stops to be free and becomes part of a rooted component in $\mathcal{I}'$. Hence, $\Lambda_{\mathcal{I}'}(X') < \Lambda_{\mathcal{I}}(X)$. Furthermore,

$$\begin{aligned}
|X'| &\leqslant (\ell + 1) + |X| \\
&\leqslant 486\sqrt{k}\lg k + \left(k - 10\sqrt{k} \cdot \lambda - 486\sqrt{k}\lg k \cdot \Lambda_{\mathcal{I}}(X)\right) \\
&\leqslant k - 10\sqrt{k} \cdot \lambda - 486\sqrt{k}\lg k \cdot \Lambda_{\mathcal{I}'}(X').
\end{aligned}$$

Consequently, X' is a pattern in $\mathcal{I}'$, and the claim is proven. ∎

The potential gains in Claim 46 allow us to conclude with the analogue of Claim 33. For its proof, we need the following estimate.

CLAIM 47. *For every $x, y > 0$ it holds that*

$$(x+y)\lg(x+y) - x\lg x \leqslant y(2+\lg(x+y)).$$

Proof. We have

$$(x+y)\lg(x+y) - x\lg x = y\lg(x+y) + x\lg(1+y/x) \leqslant y\lg(x+y) + 2y,$$

where in the last inequality we have used the fact that $\lg(1+t) \leqslant 2t$ for every $t > 0$. ◻

CLAIM 48. *Assume $c_1 \geqslant 1$ and c_3 is sufficiently large. Supposing $X \cap W_{\mathsf{isl}} \neq \emptyset$ and the subroutine of Theorem 9 returned a sequence of paths, the algorithm outputs a set A with $X \subseteq A$ with probability at least $\widehat{\mathsf{LB}}(n, \Pi(X), \Gamma, \Phi(X), \Lambda(X))$. This includes the $(1-1/k)$ probability of success of the preliminary clustering step, the $1/k$ probability that the algorithm correctly assumes that $X \cap W_{\mathsf{isl}} = \emptyset$, the k^{-7} probability of correctly choosing the vertex z, the $(10k^2 \lg n)^{-1}$ probability of choosing the right distance d, and the $1/k$ probability of choosing the right path index i.*

Proof. The proof follows along the same lines as the proof of Claim 33. Note that we have $\Lambda_{\mathcal{I}'}(X) \leqslant \Lambda_{\mathcal{I}}(X)$ and $\Phi_{\mathcal{I}'}(X) \leqslant \Phi_{\mathcal{I}}(X)$. If the first option of Claim 46 happens (i.e., the drop in the potential $\Phi(X)$), then the analysis is the same as in Claim 33. It remains to analyze the second case. Note that here we need to focus on all potentials, as we will analyze pattern X' in the instance $\mathcal{I}'$.

Since $|X' \setminus X| \leqslant 486\sqrt{k}\lg k$, we have (using Claim 47 and $|X'| \leqslant k$ for the first inequality)

$$\begin{aligned}
\Pi_{\mathcal{I}'}(X')\lg\Pi_{\mathcal{I}'}(X') - \Pi_{\mathcal{I}}(X)\lg\Pi_{\mathcal{I}}(X) &\leqslant 486\sqrt{k}\lg k\cdot(2+\lg k) \leqslant 972\sqrt{k}\lg^2 k,\\
\Phi_{\mathcal{I}'}(X') - \Phi_{\mathcal{I}}(X) &\leqslant 486\sqrt{k}\lg k,\\
\Pi_{\mathcal{I}'}(X')\lg\Gamma_{\mathcal{I}'} - \Pi_{\mathcal{I}}(X)\lg\Gamma_{\mathcal{I}} &\leqslant 486\sqrt{k}\lg k\lg n,\\
\Lambda_{\mathcal{I}'}(X') - \Lambda_{\mathcal{I}}(X) &\leqslant -1.
\end{aligned}$$

Thus, a straightforward computation shows that

$$\begin{aligned}
\frac{\widehat{\mathsf{LB}}(n', \Pi_{\mathcal{I}'}(X'), \Gamma_{\mathcal{I}'}, \Phi_{\mathcal{I}'}(X'), \Lambda_{\mathcal{I}'}(X'))}{\widehat{\mathsf{LB}}(n, \Pi_{\mathcal{I}}(X), \Gamma_{\mathcal{I}}, \Phi_{\mathcal{I}}(X), \Lambda_{\mathcal{I}}(X))} \geqslant{}& \exp\left[-1458\cdot c_1\cdot\lg^2 k(\lg k + \lg\lg n)\right]\cdot\\
&\left(1-\frac{1}{k}\right)^{c_2\cdot 486\sqrt{k}\lg k\lg n}\cdot\\
&\exp\left[c_3\left(\lg^2 k(\lg k+\lg\lg n) + \frac{\lg k\lg n}{\sqrt{k}}\right)\right]. \qquad (35)
\end{aligned}$$

Note that since $k \geqslant 10$, we have $1 - 1/k \geqslant \exp(-2/k)$, and hence

$$\left(1-\frac{1}{k}\right)^{c_2\cdot 486\sqrt{k}\lg k\lg n} \geqslant \exp\left[-972c_2\cdot\frac{\lg k\lg n}{\sqrt{k}}\right].$$

Therefore, for sufficiently large c_3, for instance $c_3 \geqslant 1458 \cdot c_1 + 972 \cdot c_2 + 12$, the right-hand side in the inequality (35) is at least $k^{12} \lg n$, required to offset the success probability of the random choices, similarly as in the proof of Claim 33. This finishes the proof of the claim and the analysis of this subcase. ∎

Subcase: A separator chain. Following the argumentation of section 5, in this section we are working with the following objects:

- a vertex $z \in V(G'') \setminus R$ with $\text{nr-dist}_{G''}(z, r) > 2000\sqrt{k} \lg k$, such that some vertex of X is within nr-distance at most 1 from z, and
- an (r, z)-separator chain $(C_1, \dots, C_p)$ in $G\langle R \rangle$ with $|C_j| \leqslant 2k$ for each $j \in [p]$, where $p = \lceil 120\sqrt{k} \lg k \rceil$.

The success probability so far in this case is at least $k^{-11} (\lg n)^{-1}$.

As in section 5, we treat $(C_1, \dots, C_p)$ as a separator chain in G and drop the first three separators. In this manner, we can assume $p \geqslant \lceil 117\sqrt{k} \lg k \rceil$, every C_i is disjoint with $T^{\mathsf{li}} \cup R$, and all vertices within nr-distance at most 3 from the root, including all light terminals, are in $\mathsf{reach}(r, G - C_i)$ for every i.

Recall that in this case the algorithm of section 5 samples an index $i \in [p]$, an integer α between 1 and $\sqrt{k}/10$, and a set $Q \subseteq C_i$ of size α. The intuition is that we hope that $Q = C_i \cap X$ and that C_i is a sparse balanced separator in the sense of Claim 34.

In our case, we additionally allow the value $\alpha = 0$ in the above sampling. Moreover, whenever there exists an index $i \in [p]$ with $C_i \cap X = \emptyset$, we consider that the algorithm made a correct guess if it sampled such an index i together with $\alpha = 0$, instead of the index provided by Claim 34.

If no such index exists, the analysis of the algorithm of section 5 remains applicable: Claim 34 still holds, and only the probability of choosing the correct α drops from $(\sqrt{k}/10)^{-1}$ to $(1 + \sqrt{k}/10)^{-1}$; however, in both cases it is at least k^{-1}, and the total probability of making correct random choices is at least $k^{-2\alpha-5}$, as in section 5. Furthermore, we have the following claim concerning the potential $\Lambda(X)$.

CLAIM 49. *If for every separator C_i we have $C_i \cap X \neq \emptyset$ and the algorithm made correct random choices, then the following holds:*

$$\Lambda_{\mathcal{I}}(X) \geqslant \Lambda_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) + \Lambda_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}).$$

Proof. Let Y be a component of X in $\mathcal{I}_D$. From the discussion in section 6.1.2 and the construction of $\mathcal{I}_{\mathsf{in}}$ it follows that $Y \cap X_{\mathsf{in}}$ is either empty or contained in a single component Y_{in} of X_{in} in $\mathcal{I}_{\mathsf{in}}$. Similarly, $Y \cap X_{\mathsf{out}}$ is empty or contained in a single connected component Y_{out} of X_{out} in $\mathcal{I}_{\mathsf{out}}$. We show that if Y is a rooted component of X in $\mathcal{I}$, then both Y_{in} and Y_{out} are rooted in their respective instances if they exist, and if Y is free, then at most one of these components is free.

We first note that if $Y \cap C_i \neq \emptyset$, then Y_{out} exists and is rooted, as all vertices of $X \cap C_i$ are neighbors of the root in $\mathcal{I}_{\mathsf{out}}$.

Assume first that Y is a rooted component of $\mathcal{I}$. Then Y_{in} is a rooted component of $\mathcal{I}_{\mathsf{in}}$, as $\mathcal{I}_{\mathsf{in}}$ is created from $\mathcal{I}$ by edge contractions only. Furthermore, if $Y \cap C_i = \emptyset$, then $Y \cap X_{\mathsf{out}} = \emptyset$, and otherwise as discussed above Y_{out} is a rooted component of $\mathcal{I}_{\mathsf{out}}$.

Now assume that Y is a free component of $\mathcal{I}$. If $Y \cap C_i = \emptyset$, then $Y \cap X_{\mathsf{in}}$ or $Y \cap X_{\mathsf{out}}$ is empty. Otherwise, as already discussed, Y_{out} is a rooted component of $\mathcal{I}_{\mathsf{out}}$. This finishes the proof of the claim. ∎

Thus, we are left with the case when there exists a separator C_i that is disjoint with X, and we assume that the algorithm correctly guessed such an index i and guessed $\alpha = 0$. The probability of making a correct choice is at least

$$\frac{1}{p} \cdot (1 + \sqrt{k}/10)^{-1} \geqslant k^{-5} = k^{-2\alpha-5}.$$

Recall that for $i \in [p]$ we defined the following partition $V(G) = V_i^{\mathsf{in}} \uplus C_i \uplus V_i^{\mathsf{out}}$:

$$V_i^{\mathsf{in}} = \mathsf{reach}(r, G - C_i) \qquad \text{and} \qquad V_i^{\mathsf{out}} = V(G) \setminus (C_i \cup V_i^{\mathsf{in}}).$$

The intuition is as follows: since $C_i \cap X = \emptyset$ and C_i does not contain any relay terminal, we can independently recurse on V_i^{in} and V_i^{out}. In V_i^{in}, we need to replace the components of $G - V_i^{\mathsf{in}}$ with relay vertices to keep the potentials bounded. On the other hand, V_i^{out} does not contain any vertex within nr-distance at most 3 from the root. Thus every vertex of the pattern X that persists in V_i^{out} is in a free component, and hence it is far (in $\mathcal{I}$). Hence, we can freely choose a new root in this subcase: by proclaiming z the new root, we make the component containing a vertex of X within nr-distance 1 from z close, thus creating a gain in the $\Lambda(X)$ potential.

Let us now proceed with formal argumentation. The algorithm defines two subinstances $\mathcal{I}_{\mathsf{out}}$ and $\mathcal{I}_{\mathsf{in}}$ as follows.

The instance $\mathcal{I}_{\mathsf{in}}$ is defined in the same way as in section 5 for $\alpha = 0$ and $Q = \emptyset$. That is, we take $\mathcal{I}_{\mathsf{in}} := (G_{\mathsf{in}}, r, T^{\mathsf{li}}{}_{\mathsf{in}}, T^{\mathsf{he}}{}_{\mathsf{in}}, R_{\mathsf{in}}, \lambda)$. Graph G_{in} is constructed from G as follows. Inspect the connected components of the graph $G - V_i^{\mathsf{in}}$. For each such component D, contract it onto a new vertex g_D that is declared to be a relay vertex. That is, we define R_{in} to be $(R \cap V_i^{\mathsf{in}}) \cup \{g_D \colon D \in \mathsf{cc}(G - V_i^{\mathsf{in}})\}$.

For the terminal sets $T^{\mathsf{li}}{}_{\mathsf{in}}$, $T^{\mathsf{he}}{}_{\mathsf{in}}$, recall that $T^{\mathsf{li}} \subseteq V_i^{\mathsf{in}}$, so all the original light terminals persist in the graph G_{in}. Thus we take $T^{\mathsf{li}}{}_{\mathsf{in}} := T^{\mathsf{li}}$. For the heavy terminals, we inherit them: $T^{\mathsf{he}}{}_{\mathsf{in}} := T^{\mathsf{he}} \cap V(G_{\mathsf{in}})$. Finally, we take $X_{\mathsf{in}} := X \cap V(G_{\mathsf{in}})$. By the same arguments as in section 5, $\mathcal{I}_{\mathsf{in}}$ is a valid instance with pattern X_{in}.

Second, we define $\mathcal{I}_{\mathsf{out}} := (G_{\mathsf{out}}, z, T^{\mathsf{li}}_{\mathsf{out}}, T^{\mathsf{he}}_{\mathsf{out}}, R_{\mathsf{out}}, \lambda)$. That is, we take the vertex z to be the new root in the instance $\mathcal{I}_{\mathsf{out}}$—this is a modification of the algorithm presented in section 5 that is triggered only when $\alpha = 0$ is sampled. Recall that V_i^{out} does not contain any light terminal. We define $G_{\mathsf{out}} := G''[V_i^{\mathsf{out}}]$, $T^{\mathsf{li}}_{\mathsf{out}} := \{z\}$, $T^{\mathsf{he}}_{\mathsf{out}} := T^{\mathsf{he}} \cap V_i^{\mathsf{out}}$, and $R_{\mathsf{out}} := R \cap V_i^{\mathsf{out}}$. In other words, we inherit all terminals and relay vertices from $\mathcal{I}$, and additionally proclaim z the root and a light terminal. Since $r \in T^{\mathsf{li}}$, we have $|T| \geqslant |T_{\mathsf{out}}|$ and, consequently, $\mathcal{I}_{\mathsf{out}}$ is a valid instance.

Finally, we take $X_{\mathsf{out}} := (X \cap V_i^{\mathsf{out}}) \cup \{z\}$. Note that $|X_{\mathsf{out}}| \leqslant |X|$, as $r \in X \setminus X_{\mathsf{out}}$.

Since $C_i \cap (X \cup R) = \emptyset$, every component of X in $\mathcal{I}$ lies either entirely in V_i^{in} or entirely in V_i^{out}. Furthermore, since every vertex within nr-distance at most 3 from the root is in V_i^{in}, every component of X lying in V_i^{out} is free, and, consequently, all vertices of $V_i^{\mathsf{out}} \cap X$ are far. Let $x \in X$ be a vertex within nr-distance at most 1 from z; clearly, $x \in V_i^{\mathsf{out}}$. The component Y of X in $\mathcal{I}$ containing x is free, but $Y \cup \{z\}$ is contained in a rooted component of X_{out} in $\mathcal{I}_{\mathsf{out}}$. This, together with the discussion of section 6.1.2 applied to the instance $\mathcal{I}_{\mathsf{in}}$, proves the following claim.

CLAIM 50. *The following holds:*

$$\begin{aligned}
\Pi_{\mathcal{I}}(X) &\geqslant \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) + \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}),\\
\Gamma_{\mathcal{I}} &\geqslant \Gamma_{\mathcal{I}_{\mathsf{in}}} + \Gamma_{\mathcal{I}_{\mathsf{out}}},\\
\Phi_{\mathcal{I}}(X) &\geqslant \Phi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) + \Phi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}),\\
\Lambda_{\mathcal{I}}(X) &\geqslant 1 + \Lambda_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}) + \Lambda_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}).
\end{aligned}$$

In particular, the last inequality of Claim 50 shows that X_{out} is a pattern in $\mathcal{I}_{\mathsf{out}}$.

We apply the algorithm recursively to instances $\mathcal{I}_{\mathsf{in}}$ and $\mathcal{I}_{\mathsf{out}}$, obtaining sets A_{in} and A_{out}. We have $T^{\mathsf{li}} = T^{\mathsf{li}}{}_{\mathsf{in}} \subseteq A_{\mathsf{in}}$ and $z \in A_{\mathsf{out}}$. We take $A := A_{\mathsf{in}} \cup A_{\mathsf{out}}$. Clearly, $T^{\mathsf{li}} \subseteq A$.

The fact that $G[A]$ admits a tree decomposition of width at most $24022\bar{\alpha}(\mathcal{C})\sqrt{k}\lg k$ with $A \cap T$ in the root bag is straightforward: since A_{in} and A_{out} are separated by C_i, there are no edges between these two sets, and we can just take a root bag $A \cap T$ and attach as children the decompositions of $G[A_{\mathsf{in}}]$ and $G[A_{\mathsf{out}}]$.

We are left with analyzing the success probability. All the necessary observations have already been made in Claim 50, so we can conclude with the following claim.

Claim 51. *Assume $c_1 \geqslant 2$ and $c_3 \geqslant 17$. Supposing $X \cap W_{\mathsf{isl}} \neq \emptyset$ and the subroutine of Theorem 9 returned a separator chain, the algorithm outputs a set A with $X \subseteq A$ with probability at least $\widehat{\mathsf{LB}}(n, \Pi(X), \Gamma, \Phi(X), \Lambda(X))$. This includes the $(1 - 1/k)$ probability of success of the preliminary clustering step, the $1/k$ probability that the algorithm correctly assumes that $X \cap W_{\mathsf{isl}} = \emptyset$, the k^{-7} probability of correctly choosing the island C that intersects the pattern, the $(10k^2 \lg n)^{-1}$ probability of choosing the right distance d, and $k^{-2\alpha-5}$ probability of correctly sampling the i, α, and set Q.*

Proof. The case $\alpha > 0$ has been already discussed, and is the same as in section 5, with the help of Claim 49 to control the split of the potential $\Lambda(X)$. For $\alpha = 0$, Claim 50 ensures that

$$\begin{aligned}
&\widehat{\mathsf{LB}}(n_{\mathsf{in}}, \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}), \Gamma_{\mathcal{I}_{\mathsf{in}}}, \Phi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}), \Lambda_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}))\cdot\\
&\widehat{\mathsf{LB}}(n_{\mathsf{out}}, \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}), \Gamma_{\mathcal{I}_{\mathsf{out}}}, \Phi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}), \Lambda_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}))\\
\geqslant &\widehat{\mathsf{LB}}(n, \Pi_{\mathcal{I}}(X), \Gamma_{\mathcal{I}}, \Phi_{\mathcal{I}}(X), \Lambda_{\mathcal{I}}(X)) \cdot \exp\left[c_3 \lg^2 k(\lg k + \lg\lg n)\right].
\end{aligned}$$

Hence,

$$\begin{aligned}
\mathbb{P}(X \subseteq A) \geqslant &\left(1 - \frac{1}{k}\right) \cdot k^{-8} \cdot (10k^2 \lg n)^{-1} \cdot k^{-5} \cdot\\
&\widehat{\mathsf{LB}}(n_{\mathsf{out}}, \Pi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}), \Gamma_{\mathcal{I}_{\mathsf{out}}}, \Phi_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}}), \Lambda_{\mathcal{I}_{\mathsf{out}}}(X_{\mathsf{out}})) \cdot\\
&\widehat{\mathsf{LB}}(n_{\mathsf{in}}, \Pi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}), \Gamma_{\mathcal{I}_{\mathsf{in}}}, \Phi_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}), \Lambda_{\mathcal{I}_{\mathsf{in}}}(X_{\mathsf{in}}))\\
\geqslant &\, k^{-17} \cdot (\lg n)^{-1} \cdot \exp\left[c_3 \lg^2 k(\lg k + \lg\lg n)\right] \cdot\\
&\widehat{\mathsf{LB}}(n, \Pi_{\mathcal{I}}(X), \Gamma_{\mathcal{I}}, \Phi_{\mathcal{I}}(X), \Lambda_{\mathcal{I}}(X))\\
\geqslant &\, \widehat{\mathsf{LB}}(n, \Pi_{\mathcal{I}}(X), \Gamma_{\mathcal{I}}, \Phi_{\mathcal{I}}(X), \Lambda_{\mathcal{I}}(X)).
\end{aligned}$$

This finishes the proof of the claim and concludes the description of the third and last subcase. ∎

6.1.6. Wrap-up: A multiple-component version of Theorem 1. Let us now take a step back and use the developed recursive algorithm to obtain a multiple-component version of Theorem 1, namely Theorem 39, following along the lines of the reasoning of section 5.1.

Assume we are given an n-vertex graph G_0 from a minor-closed graph class $\mathcal{C}$ that excludes some apex graph as a minor. We are looking for a pattern $X \subseteq V(G_0)$ of size k such that $G_0[X]$ has at most d connected components. Similarly as in section 5.1, we can guess (by sampling at random) one vertex $x \in X$ and create an instance

$\mathcal{I}$ of our recursive problem with $G := G_0$, $r_0 := x$, $T^{\mathsf{li}} := \{x\}$, and $T^{\mathsf{he}} := R := \emptyset$. Since $G[X]$ has d connected components, we have $\Lambda_{\mathcal{I}}(X) \leqslant d - 1$.

However, X may not be a pattern in $\mathcal{I}$ due to the size penalty incurred by multiple connected components. Instead, provided we assume that $d \leqslant c \cdot \sqrt{k}/\lg k$ for some constant $c \geqslant 1$, we can invoke the recursive subproblem with the parameter $k' := (10^5 \cdot c)^3 k = \mathcal{O}(k)$. Then, as $\sqrt{\alpha k}\lg(\alpha k) \leqslant \alpha^{2/3}\sqrt{k}\lg k$ for every $k \geqslant 10$ and $\alpha \geqslant 10^5$, we have

$$k' - d \cdot 486 \cdot \sqrt{k'}\lg k' \geqslant (10^5 \cdot c)^3 k - 486 \cdot 10^{10} c^3 k > k \geqslant |X|.$$

Consequently, X is a pattern in $\mathcal{I}$ for the parameter $k' = \mathcal{O}(k)$.

Let us now look at the contribution of the term including the potential $\Lambda_{\mathcal{I}}(X)$ to the success probability. Clearly, we have $\Lambda_{\mathcal{I}}(X) \leqslant d \leqslant c \cdot \sqrt{k}/\lg k \leqslant c \cdot \sqrt{k'}/\lg k'$. Therefore, using the estimate on $\mathsf{LB}(n, \Pi(X), \Gamma, \Phi(X))$ from section 5.1, we obtain that

$$\begin{aligned}
&\widehat{\mathsf{LB}}(n, \Pi(X), \Gamma, \Phi(X), \Lambda(X)) \\
&\quad = \mathsf{LB}(n, \Pi(X), \Gamma, \Phi(X)) \cdot \exp\left[-c_3 \cdot \Lambda(X) \cdot \left(\lg^2 k'(\lg k' + \lg\lg n) + \frac{\lg n \lg k'}{\sqrt{k'}}\right)\right] \\
&\quad \geqslant \left(2^{\mathcal{O}(\sqrt{k}\lg^2 k)} n^{\mathcal{O}(1)}\right)^{-1} \cdot \exp\left[-c_3 \cdot c \cdot \left(\sqrt{k'}\lg k'(\lg k' + \lg\lg n) + \lg n\right)\right] \\
&\quad \geqslant \left(2^{\mathcal{O}(\sqrt{k}\lg^2 k)} n^{\mathcal{O}(1)}\right)^{-1}.
\end{aligned}$$

Note that we used Claim 19 in the last inequality. This finishes the proof of Theorem 39.

6.2. The excluded minor case. We now use Theorem 39—the multiple-components variant from the previous section—to prove Theorem 38—the variant for arbitrary proper minor-closed classes. Let us fix a graph G and an integer k as in the theorem statement. Furthermore, let X be a pattern in G such that $G[X]$ admits a spanning tree S of maximum degree Δ for a fixed constant Δ.

Robertson–Seymour decomposition theorem. As announced at the beginning of this section, we use the decomposition theorem of Robertson and Seymour for graphs excluding a fixed minor. To make use of locally bounded treewidth, we will use the variant of Grohe [26].

To formulate this decomposition, we need some notation. Recall that we use a notation $\mathcal{T} = (T, \beta)$ for a tree decomposition, where T is a tree and $\beta\colon V(T) \to 2^{V(G)}$ is the function that assigns bags to nodes of T.

The set $\beta(t) \cap \beta(t')$ for an edge $tt' \in E(T)$ is called an *adhesion* of tt'. For a node $t \in V(T)$, the *torso* of the node t, denoted by torso(t), is the graph $G[\beta(t)]$ with every adhesion $\beta(t) \cap \beta(t')$ for $t' \in N_T(t)$ turned into a clique.

Recall that we consider rooted tree decompositions; that is, the tree T is rooted in one node. For a nonroot node $t \in V(T)$, by parent(t) we denote the parent of t in T. Furthermore, we denote:

$$\sigma(t) = \begin{cases} \emptyset & \text{if t is the root of } T, \\ \beta(t) \cap \beta(\text{parent}(t)) & \text{otherwise.} \end{cases}$$

We are now ready to formulate the variant of the Robertson–Seymour decomposition of Grohe that we use.

THEOREM 52 ([26]). *For every proper minor-closed graph class $\mathcal{C}$ there exist a constant h and an apex-minor-free graph class $\mathcal{C}'$ such that the following holds. Given a graph $G \in \mathcal{C}$, one can in polynomial time compute a tree decomposition $\mathcal{T} = (T, \beta)$ of G together with a family of sets $(Z_t)_{t \in V(T)}$ such that every adhesion of $\mathcal{T}$ has size at most h, and for every $t \in V(T)$ the set Z_t is a subset of $\beta(t)$ of size at most h, and the graph* $\mathrm{torso}(t) - Z_t$ *belongs to $\mathcal{C}'$.*

Furthermore, we need the following variant of Baker's shifting technique.

THEOREM 53 ([26]). *Let $\mathcal{C}$ be an apex-minor-free graph class. Given a graph $G \in \mathcal{C}$ and an integer ℓ, one can in polynomial time compute a partition of $V(G)$ into ℓ sets $L_1, L_2, \ldots, L_\ell$, such that for every $1 \leqslant i \leqslant \ell$ the graph $G - L_i$ has treewidth $\mathcal{O}(\ell)$.*

The algorithm. Let us now proceed to the description of the algorithm. Given a graph $G \in \mathcal{C}$, we compute its tree decomposition $\mathcal{T} = (T, \beta)$ and sets $(Z_t)_{t \in V(T)}$ using Theorem 52. Paying $1/n$ in the success probability, we guess an arbitrary vertex $r \in X$ and root the decomposition $\mathcal{T}$ in a bag t_r such that $r \in \beta(t_r)$. By slightly abusing the notation, we proclaim $\sigma(t_r) = \{r\}$. By restricting our attention to the connected component of G that contains r, we may assume that G is connected. Therefore, we can henceforth assume that every adhesion of $\mathcal{T}$ is nonempty.

For every $t \in V(T)$, we create an instance $\mathcal{I}_t := (G_t, r_t, T^{\mathsf{li}}{}_t, T^{\mathsf{he}}{}_t, R_t, \lambda_t)$ of the recursive problem as follows. We first take $G_t := \mathrm{torso}(t) - Z_t$. If $\sigma(t) \not\subseteq Z_t$, we define r_t to be an arbitrary vertex of $\sigma(t) \setminus Z_t$; otherwise, we create a new vertex r_t and make it adjacent to an arbitrary vertex of G_t. We set $T^{\mathsf{li}}{}_t := \{r_t\} \cup (\sigma(t) \setminus Z_t)$, $T^{\mathsf{he}}{}_t := \emptyset$, $R_t := \emptyset$, and $\lambda_t := 0$.

Furthermore, we define $X_t^\circ := X \cap \beta(t)$ and $X_t := (X_t^\circ \setminus Z_t) \cup \{r_t\}$. Note that $\mathrm{torso}(t)[X_t^\circ]$ is connected, as $G[X]$ is connected. Furthermore, we claim that $\mathrm{torso}(t)[X_t^\circ]$ admits a spanning tree of bounded degree.

CLAIM 54. *There exists a spanning tree of* $\mathrm{torso}(t)[X_t^\circ]$ *of maximum degree* 2Δ.

Proof. We construct a connected spanning subgraph S_t of $\mathrm{torso}(t)[X_t^\circ]$ of maximum degree 2Δ as follows. First, we take $V(S_t) := X_t^\circ$ and $E(S_t) := E(S) \cap E(\mathrm{torso}(t)[X_t^\circ])$. Second, for every $t' \in N_T(t)$, we perform the following operation. Let $\mathrm{leg}(t, t')$ be the set of those vertices $v \in X_t^\circ \cap \beta(t) \cap \beta(t')$ for which v is incident to an edge $uv \in E(S)$ with $u \in \beta(t') \setminus \beta(t)$. If $|\mathrm{leg}(t, t')| \geqslant 2$, we add to $E(S_t)$ edges of an arbitrary path on vertex set $\mathrm{leg}(t, t')$; such a path exists in $\mathrm{torso}(t)$ as the adhesion $\beta(t) \cap \beta(t')$ is turned into a clique.

Let us now show that S_t is connected. To this end, consider a maximal path P in S between two vertices of X_t° such that no edge or internal vertex of P belongs to $\mathrm{torso}(t)$. Let v_1, v_2 be the endpoints of P. By the properties of a tree decomposition, there exists a node $t' \in N_T(t)$ such that $v_1, v_2 \in \beta(t) \cap \beta(t')$ and the first and last edges of P are v_1u_1 and v_2u_2 with $u_1, u_2 \in \beta(t') \setminus \beta(t)$ (possibly $u_1 = u_2$). However, then $v_1, v_2 \in \mathrm{leg}(t, t')$, and they remain connected in S_t. This shows that S_t is connected.

To bound the maximum degree of S_t, note that for every $t' \in N_T(t)$ and every $v \in \mathrm{leg}(t, t')$ at most two edges incident to v are added to S_t when considering t'. These two edges can be charged to the edge $vu \in E(S)$ with $u \in \beta(t') \setminus \beta(t)$ that certifies that $v \in \mathrm{leg}(t, t')$. We have $vu \in E(S) \setminus E(S_t)$, and every edge vu can be charged at most once. Since S has maximum degree at most Δ by assumption, the bound on the maximum degree of S_t follows. ◻

We define $k' := (10^5 \cdot \Delta h)^3 \cdot k = \mathcal{O}(k)$. We will use the machinery of section 6.1, in particular all the potentials, using the parameter k' instead of the input parameter k. The reason for this is that we need to pay the penalty in the size of the pattern for multiple connected components of X_t in $\mathcal{I}_t$. The following claim verifies that it suffices to inflate k by a constant factor.

CLAIM 55. *The graph $G_t[X_t]$ has at most $2\Delta h + 1$ connected components and $\Lambda_{\mathcal{I}_t}(X_t) \leqslant 2\Delta h$. Furthermore, the set X_t is a pattern in $\mathcal{I}_t$ with respect to the parameter k'.*

Proof. Since the maximum degree of S_t is at most 2Δ and $|Z_t| \leqslant h$, there are at most $2\Delta h$ connected components of $\operatorname{torso}(t)[X_t^\circ] - Z_t$. Consequently, $G_t[X_t]$ has at most $2\Delta h + 1$ connected components and

$$\Lambda_{\mathcal{I}_t}(X_t) \leqslant 2\Delta h.$$

Then, as $\sqrt{\alpha k}\lg(\alpha k) \leqslant \alpha^{2/3}k$ for every $k \geqslant 10$ and $\alpha \geqslant 10^5$, we have

$$k' - 486\sqrt{k'}\lg k' \cdot \Lambda_{\mathcal{I}_t}(X_t) \geqslant (10^5 \cdot \Delta h)^3 k - 486 \cdot (10^5 \cdot \Delta h)^2 \cdot 2\Delta h \cdot k > 2k \geqslant |X_t|.$$

Thus, X_t satisfies the size bound for a pattern in $\mathcal{I}_t$. ∎

For every node $t \in V(T)$, we are going to look for the pattern X_t in the instance $\mathcal{I}_t$, using k' as the parameter. Observe that the first three potentials partition well between the instances.

CLAIM 56. *The following holds if we measure the potentials with respect to the parameter k':*

$$k \geqslant \sum_{t \in V(T)} \Pi_{\mathcal{I}_t}(X_t) \qquad \text{and} \qquad n \geqslant \sum_{t \in V(T)} \Gamma_{\mathcal{I}_t} \qquad \text{and} \qquad k \geqslant \sum_{t \in V(T)} \Phi_{\mathcal{I}_t}(X_t).$$

Proof. The crucial observation is that for an edge between t and $\operatorname{parent}(t)$ in T every vertex $v \in \sigma(t) = \beta(t) \cap \beta(\operatorname{parent}(t))$ is either in Z_t or a light terminal in $\mathcal{I}_t$. Consequently, every vertex $v \in V(G)$ is present but *not* a light terminal in at most one instance $\mathcal{I}_t$. ∎

However, the potentials $\Lambda_{\mathcal{I}_t}(X_t)$ do not behave as nicely as the other potentials in Claim 56: they are bounded by $2\Delta h$ by Claim 55 but may be positive in $\Omega(k)$ instances. Thus, we cannot afford to apply the algorithm of the previous section to every instance $\mathcal{I}_t$ separately.

Instead, for every instance $\mathcal{I}_t$, we make a random choice. With probability $1/k$, we proclaim $\mathcal{I}_t$ *interesting* and apply the recursive algorithm to $\mathcal{I}_t$, obtaining a set $A_t \subseteq (\beta(t) \setminus Z_t) \cup \{r_t\}$ such that $G_t[A_t]$ is of treewidth $\mathcal{O}(\sqrt{k'}\lg k') = \mathcal{O}(\sqrt{k}\lg k)$. Furthermore, note that $\sigma(t) \subseteq A_t \cup Z_t$. Here, we use the fact that the graph G_t underlying $\mathcal{I}_t$ belongs to an apex-minor-free graph class $\mathcal{C}'$ promised by Theorem 52 (without loss of generality, this class is closed under adding degree-vertices, and thus the graph persists in $\mathcal{C}'$ after possibly adding r_t as a new vertex).

With the remaining probability, we proclaim $\mathcal{I}_t$ *standard* and proceed as follows. First, we apply the algorithm of Theorem 53 to the graph G_t with $\ell := \lceil c_3\sqrt{k'}\lg k' \rceil$ obtaining a partition $L_1^t, L_2^t, \ldots, L_\ell^t$. Second, we pick a random index $1 \leqslant i_t \leqslant \ell$. Third, we set $A_t := (V(G_t) \setminus L_{i_t}^t) \cup \{r_t\} \cup (\sigma(t) \setminus Z_t)$. Note that in this case also $G_t[A_t]$

has treewidth $\mathcal{O}(\sqrt{k}\lg k)$ as $\ell = \mathcal{O}(\sqrt{k}\lg k)$ and $|\sigma(t)| \leqslant h = \mathcal{O}(1)$. Furthermore, we have again $\sigma(t) \subseteq A_t \cup Z_t$.

We define

$$A := \{r\} \cup \bigcup_{t \in V(T)} (A_t \cup Z_t) \setminus (\sigma(t) \cup \{r_t\}).$$

We claim that A satisfies the desired properties. The treewidth bound is easy.

CLAIM 57. *$G[A]$ is of treewidth $\mathcal{O}(\sqrt{k}\lg k)$.*

Proof. Since $|Z_t| \leqslant h = \mathcal{O}(1)$ for every $t \in V(T)$, we have that $\mathrm{torso}(t)[A_t \cup Z_t]$ is of treewidth $\mathcal{O}(\sqrt{k}\lg k)$. Since $\sigma(t) \subseteq A_t \cup Z_t$ for every $t \in V(T)$, we have that $G[A]$ can be constructed from graphs $\mathrm{torso}(t)[A_t \cup Z_t]$ for $t \in V(T)$ using vertex deletions and clique sums along cliques of size at most h, and the claim follows. ◻

Finally, we check the probability that $X \subseteq A$. To this end, we need the following simple estimate.

CLAIM 58. *Let a, b be positive integers, and let $a_1, a_2, \ldots, a_p$ be integers such that $0 \leqslant a_i < a$ for every $1 \leqslant i \leqslant p$ and $\sum_{i=1}^{p} a_i \leqslant b$. Then*

$$\prod_{i=1}^{p} \left(1 - \frac{a_i}{a}\right) \geqslant a^{-2b/a-1}.$$

Proof. We use the following local improvement argument: whenever we have two indices $1 \leqslant i < j \leqslant p$ such that $a_i + a_j < a$, we can replace a_i and a_j with $a_i + a_j$, since

$$\left(1 - \frac{a_i}{a}\right)\left(1 - \frac{a_j}{a}\right) \geqslant 1 - \frac{a_i + a_j}{a}.$$

Thus, we can assume that for every $1 \leqslant i < j \leqslant r$ we have that $a_i + a_j \geqslant a$. In particular, every index i satisfies $a_i \geqslant a/2$, apart from at most one. We infer that $p \leqslant 2b/a + 1$. Since $1 - \frac{a_i}{a} \geqslant a^{-1}$ for each i due to a, a_i being integers, the claim follows. ◻

CLAIM 59. *The probability that $X \subseteq A$ is at least $(2^{\mathcal{O}(\sqrt{k}\lg^2 k)} \cdot n^{\mathcal{O}(1)})^{-1}$.*

Proof. Note that we have the following partition:

$$V(G) = \{r\} \uplus \biguplus_{t \in V(T)} \beta(t) \setminus \sigma(t).$$

By the definition of the set A, we have $r \in A$, and for every $t \in V(T)$ and for every $v \in \beta(t) \setminus \sigma(t)$ it holds that $v \in A$ if and only if $v \in A_t \cup Z_t$. Consequently, by the definition of X_t and G_t, if for every $t \in V(T)$ we have $X_t \subseteq A_t$, then we have $X \subseteq A$. In what follows we will argue that with sufficient probability it holds that for every $t \in V(T)$ we indeed have $X_t \subseteq A_t$.

First, observe that this assertion is clearly true for every node t where $X_t \subseteq \{r_t\} \cup (\sigma(t) \setminus Z_t)$, as both in standard and interesting nodes we have $\sigma(t) \subseteq A_t \cup Z_t$ and $r_t \in A_t$.

If this is not the case for a node t (i.e., $X_t \not\subseteq \{r_t\} \cup (\sigma(t) \setminus Z_t)$), we call the node t *touched*. Note that we have $\Pi_{\mathcal{I}_t}(X_t) > 0$ for a touched node t. Hence, there are at most k touched nodes. We require that a touched node t is proclaimed interesting if $\Pi_{\mathcal{I}_t}(X_t) \geqslant c_3\sqrt{k'}\lg k'$ and standard otherwise. Note that Claim 56 implies that we require at most $\sqrt{k'}/(c_3 \lg k')$ nodes to be interesting and at most k nodes to be standard. Consequently, the probability that we proclaim touched nodes as requested is lower bounded by

$$\left(\frac{1}{k}\right)^{\sqrt{k'}/(c_3\lg k')} \cdot \left(1-\frac{1}{k}\right)^k = 2^{-\mathcal{O}(\sqrt{k})}.$$

In every standard touched node t we have $X_t \subseteq A_t$ if $X_t \cap L^t_{i_t} \subseteq T^{\mathsf{li}}{}_t$, as $T^{\mathsf{li}}{}_t = \{r_t\} \cup (\sigma(t) \setminus Z_t)$. We have $X_t \cap L^t_{i_t} \subseteq T^{\mathsf{li}}{}_t$ with probability at least $1 - |X_t \setminus T^{\mathsf{li}}{}_t|/\ell = 1 - \Pi_{\mathcal{I}_t}(X_t)/\ell$. Recall that $\ell = \lceil c_3\sqrt{k'}\lg k'\rceil$ but $\Pi_{\mathcal{I}_t}(X_t) < \ell$ in a standard node t. Consequently, since $\sum_{t\in V(T)} \Pi_{\mathcal{I}_t}(X_t) \leqslant k'$, by Claim 58 we infer that the probability that in every standard node we have $X_t \subseteq A_t$ is at most

$$\ell^{-2k'/\ell-1} = 2^{-\mathcal{O}(\sqrt{k})}.$$

Let us now consider an interesting node t, that is, a node t with $\Pi_{\mathcal{I}_t}(X_t) \geqslant c_3\sqrt{k'}\lg k'$. Let Z_{int} be the set of these nodes; note that $|Z_{\mathsf{int}}| \leqslant \sqrt{k'}/(c_3\lg k')$. Since X_t is a pattern in $\mathcal{I}_t$, we infer that the probability of the event $X_t \subseteq A_t$ is at least $\widehat{\mathsf{LB}}(n, \Pi_{\mathcal{I}_t}(X_t), \Gamma_{\mathcal{I}_t}, \Phi_{\mathcal{I}_t}(X_t), \Lambda_{\mathcal{I}_t}(X_t))$ (with respect to the parameter k'). By Claims 55 and 56 we have that

$$\begin{aligned}
&\prod_{t\in Z_{\mathsf{int}}} \widehat{\mathsf{LB}}(n, \Pi_{\mathcal{I}_t}(X_t), \Gamma_{\mathcal{I}_t}, \Phi_{\mathcal{I}_t}(X_t), \Lambda_{\mathcal{I}_t}(X_t)) \\
&\quad\leqslant \prod_{t\in Z_{\mathsf{int}}} \mathsf{LB}(n, \Pi_{\mathcal{I}_t}(X_t), \Gamma_{\mathcal{I}_t}, \Phi_{\mathcal{I}_t}(X_t)) \\
&\qquad\cdot \prod_{t\in Z_{\mathsf{int}}} \exp\left[-c_3 \cdot \Lambda_{\mathcal{I}_t}(X_t)\cdot\left(\lg^2 k'(\lg k' + \lg\lg n) + \frac{\lg n \lg k'}{\sqrt{k'}}\right)\right] \\
&\quad\leqslant \left(2^{\mathcal{O}(\sqrt{k}\lg^2 k)} n^{\mathcal{O}(1)}\right)^{-1} \\
&\qquad\cdot \exp\left[-|Z_{\mathsf{int}}|\cdot c_3\cdot 2\Delta h\cdot\left(\lg^2 k'(\lg k' + \lg\lg n) + \frac{\lg n\lg k'}{\sqrt{k'}}\right)\right] \\
&\quad\leqslant \left(2^{\mathcal{O}(\sqrt{k}\lg^2 k)} n^{\mathcal{O}(1)}\right)^{-1}\cdot\exp\left[-2\Delta h\left(\sqrt{k'}\lg k'(\lg k' + \lg\lg n) + \lg n\right)\right] \\
&\quad\leqslant \left(2^{\mathcal{O}(\sqrt{k}\lg^2 k)} n^{\mathcal{O}(1)}\right)^{-1}.
\end{aligned}$$

Here, we estimated the product of the terms $\mathsf{LB}(n, \Pi_{\mathcal{I}_t}(X_t), \Gamma_{\mathcal{I}_t}, \Phi_{\mathcal{I}_t}(X_t))$ using Claim 56 as in section 5, and in the last inequality we used Claim 19. ◻

This concludes the proof of Theorem 38.

7. Conclusions. In this work we have laid foundations for a new tool for obtaining subexponential parameterized algorithms for problems on planar graphs and more generally on graphs that exclude a fixed apex graph as a minor. The technique is applicable to problems that can be expressed as searching for a small, connected pattern in a large host graph. Using the new approach, we designed, in a generic manner, a number of subexponential parameterized algorithms for problems for which the existence of such algorithms was open. We believe, however, that this work provides only the basics of a new methodology for the design of parameterized algorithms on planar and apex-minor-free graphs. This methodology goes beyond the paradigm of bidimensionality and is yet to be developed.

An immediate question raised by our work is whether the technique can be derandomized. Note that our main result, Theorem 1, immediately yields the following combinatorial statement.

THEOREM 60. *Let $\mathcal{C}$ be a class of graphs that exclude a fixed apex graph as a minor. Suppose G is an n-vertex graph from $\mathcal{C}$ and k is a positive integer. Then there exists a family $\mathcal{F}$ of subsets of vertices G satisfying the following properties:*

(P1) *For each $A \in \mathcal{F}$, the treewidth of $G[A]$ is at most $\mathcal{O}(\sqrt{k}\log k)$.*

(P2) *For each vertex subset $X \subseteq V(G)$ such that $G[X]$ is connected and $|X| \leqslant k$, there exists some $A \in \mathcal{F}$ for which $X \subseteq A$.*

(P3) *It holds that $|\mathcal{F}| \leqslant 2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot n^{\mathcal{O}(1)}$.*

Proof. Let $f(n,k) \in 2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot n^{\mathcal{O}(1)}$ be the inverse of the lower bound on the success probability of the algorithm of Theorem 1. Repeat the algorithm of Theorem 1 $f(n,k)\cdot 2k\ln n$ times, and consider the list of obtained vertex subsets as a candidate for $\mathcal{F}$. Let us fix some $X \subseteq V(G)$ such that $|X| \leqslant k$ and $G[X]$ is connected, and consider the probability that there is some $A \in \mathcal{F}$ for which $X \subseteq A$. For one particular run of the algorithm of Theorem 1, this holds with probability at least $f(n,k)^{-1}$. As the runs are independent, the probability that no element of $\mathcal{F}$ covers X is upper bounded by

$$\left(1 - \frac{1}{f(n,k)}\right)^{f(n,k)\cdot 2k\ln n} \leqslant e^{-2k\ln n} = n^{-2k}.$$

As the number of k-vertex subsets of $V(G)$ is upper bounded by n^k, we infer that the expected value of the number of sets X for which there is no element of $\mathcal{F}$ covering them is upper bounded by $n^{-k} < 1$. Hence, there is a run of the described experiment for which this number is zero; this run yields the desired family $\mathcal{F}$. ☐

The above proof of Theorem 60 gives only a randomized algorithm constructing a family $\mathcal{F}$ that indeed covers all small patterns with high probability. We conjecture that the algorithm can be derandomized; that is, that the family whose existence is asserted by Theorem 60 can be computed in time $2^{\mathcal{O}(\sqrt{k}\log^2 k)} \cdot n^{\mathcal{O}(1)}$. So far we are able to derandomize most of the components of the algorithm, primarily using standard constructions based on splitters and perfect hash families [34]. One part of the reasoning with which we still struggle is the clustering step (Theorem 8). Our optimism with derandomizing this last part stems from its resemblance to the construction of HSTs of [4], which have been subsequently derandomized [8].

Q1. Is it possible to construct a family with properties described in Theorem 60 in deterministic time $2^{\mathcal{O}(\sqrt{k}\log^c k)} \cdot n^{\mathcal{O}(1)}$ for some constant c?

In section 6 we attempted to generalize our technique to the cases when the pattern is disconnected and when the class only excludes some fixed (but arbitrary) graph H as a minor. In the case of disconnected patterns, we were able to prove a suitable generalization of Theorem 1; however, the success probability of the algorithm depends inverse-exponentially on the number of connected components of the pattern (see Theorem 39). In the case of general H-minor-free classes, we needed to assume that the pattern admits a spanning tree of constant maximum degree (see Theorem 38). So far we do not see any reason for any of these constraints to be necessary.

Q2. Is it possible to prove Theorem 1 without the assumption that the subgraph induced by X has to be connected?

Q3. Is it possible to prove Theorem 1 only under the assumption that all graphs from $\mathcal{C}$ exclude some fixed (but arbitrary) graph H as a minor?

Our next question concerns local search problems in the spirit of the LS Vertex Cover problem considered in section 1. Apart from this problem, Fellows et al. [23] designed FPT algorithms also for the local search for a number of other problems on apex-minor-free classes, including LS Dominating Set and its distance-d generalization. Here, we are given a dominating set S in a graph G from some apex-minor-free class $\mathcal{C}$, and we ask whether there exists a strictly smaller dominating set S' that is at Hamming distance at most k from S. Again, the approach of Fellows et al. [23] is based on the observation that if there is some solution, then there is also a solution S' such that $S \triangle S'$ can be connected using at most k additional vertices. Thus, we need to search for a connected pattern of size $2k$, instead of k, in which suitable sets $S \setminus S'$ and $S' \setminus S$ are to be found. Unfortunately, now the preprocessing step fails: vertices outside A may require being dominated from within A, which poses additional constraints that are not visible in the graph $G[A]$ only. Hence, we cannot just focus on the graph $G[A]$. Observe, however, that the whole reasoning would go through if A covered not just $S \triangle S'$ but also its neighborhood. More generally, if the considered problem concerns domination at distance d, then we should cover the distance-d neighborhood of $S \triangle S'$. This motivates the following question.

Q4. Fix some positive constant d. Is it possible to prove a stronger version of Theorem 1, where the sampled set A is required to cover the whole distance-d neighborhood of the set X with the same asymptotic lower bound on the success probability?

Finally, so far we do not know whether the connectivity condition in Corollary 5 is necessary.

Q5. Is it possible to solve Subgraph Isomorphism on planar graphs in time $2^{\mathcal{O}(k/\log k)} \cdot n^{\mathcal{O}(1)}$, even without the assumption that the pattern graph is connected?

Note that a positive answer to Q2 implies a positive answer here as well, as the algorithm of Theorem 4 does not require the pattern graph to be connected.

Acknowledgments. The authors thank Felix Reidl, Fernando Sánchez Villaamil, Somnath Sikdar, and Yngve Villanger for some preliminary discussions on the topic of this paper as well as an anynomous reviewer for pointing out the similarities between our clustering procedure and the tools from the theory of metric embeddings. Part of the research Is it OK to have added the research text of the first, third, and fourth authors was conducted when they were participating in the "Fine-grained complexity and algorithm design" program at the Simons Institute for Theory of Computing in Berkeley.

REFERENCES

[1] J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier, *Fixed parameter algorithms for dominating set and related problems on planar graphs*, Algorithmica, 33 (2002), pp. 461–493.

[2] N. Alon, P. Seymour, and R. Thomas, *Planar separators*, SIAM J. Discrete Math., 7 (1994), pp. 184–193, https://doi.org/10.1137/S0895480191198768.

[3] N. Alon, R. Yuster, and U. Zwick, *Color-coding*, J. ACM, 42 (1995), pp. 844–856.

[4] Y. Bartal, *On approximating arbitrary metrices by tree metrics*, in Proceedings of the 30th Annual ACM Symposium on the Theory of Computing, STOC 1998, 1998, pp. 161–168.
[5] H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof, *Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth*, Inform. Comput., 243 (2015), pp. 86–111, https://doi.org/10.1016/j.ic.2014.12.008.
[6] H. L. Bodlaender, J. Nederlof, and T. C. van der Zanden, *Subexponential time algorithms for embedding H-minor free graphs*, in Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, LIPIcs Int. Proc. Inform. 55, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2016, 9.
[7] G. Borradaile, P. Klein, D. Marx, and C. Mathieu, *Algorithms for optimization problems in planar graphs (Dagstuhl seminar* 13421*)*, Dagstuhl Rep., 3 (2014), pp. 36–57, https://doi.org/10.4230/DagRep.3.10.36.
[8] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. A. Plotkin, *Approximating a finite metric by a small number of tree metrics*, in Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, FOCS 1998, 1998, pp. 379–388, https://doi.org/10.1109/SFCS.1998.743488.
[9] M. Chimani, P. Mutzel, and B. Zey, *Improved Steiner tree algorithms for bounded treewidth*, J. Discrete Algorithms, 16 (2012), pp. 67–78.
[10] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized Algorithms*, Springer, 2015, https://doi.org/10.1007/978-3-319-21275-3.
[11] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos, *Fixed-parameter algorithms for (k, r)-center in planar graphs and map graphs*, ACM Trans. Agorithms, 1 (2005), pp. 33–47.
[12] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos, *Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs*, J. ACM, 52 (2005), pp. 866–893.
[13] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos, *Bidimensional structures: Algorithms, combinatorics and logic (Dagstuhl seminar* 13121*)*, Dagstuhl Rep., 3 (2013), pp. 51–74, https://doi.org/10.4230/DagRep.3.3.51.
[14] E. D. Demaine and M. Hajiaghayi, *The bidimensionality theory and its algorithmic applications*, Comput. J., 51 (2008), pp. 292–302.
[15] E. D. Demaine and M. Hajiaghayi, *Linearity of grid minors in treewidth with applications through bidimensionality*, Combinatorica, 28 (2008), pp. 19–36.
[16] E. D. Demaine and M. T. Hajiaghayi, *Equivalence of local treewidth and linear local treewidth and its algorithmic applications*, in Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, SIAM, 2004, pp. 840–849.
[17] F. Dorn, *Planar subgraph Isomorphism revisited*, in Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, LIPIcs Int. Proc. Inform. 5, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2010, pp. 263–274.
[18] F. Dorn, F. V. Fomin, D. Lokshtanov, V. Raman, and S. Saurabh, *Beyond bidimensionality: Parameterized subexponential algorithms on directed graphs*, Inform. Comput., 233 (2013), pp. 60–70.
[19] F. Dorn, F. V. Fomin, and D. M. Thilikos, *Catalan structures and dynamic programming in H-minor-free graphs*, J. Comput. System Sci., 78 (2012), pp. 1606–1622.
[20] D. Eppstein, *Subgraph isomorphism in planar graphs and related problems*, J. Graph Algorithms Appl., 3 (1999), pp. 1–27.
[21] D. Eppstein, *Diameter and treewidth in minor-closed graph families*, Algorithmica, 27 (2000), pp. 275–291, https://doi.org/10.1007/s004530010020.
[22] U. Feige, M. Hajiaghayi, and J. R. Lee, *Improved approximation algorithms for minimum weight vertex separators*, SIAM J. Comput., 38 (2008), pp. 629–657.
[23] M. R. Fellows, F. V. Fomin, D. Lokshtanov, F. A. Rosamond, S. Saurabh, and Y. Villanger, *Local search: Is brute-force avoidable?*, J. Comput. Syst. Sci., 78 (2012), pp. 707–719, https://doi.org/10.1016/j.jcss.2011.10.003.
[24] F. V. Fomin, D. Lokshtanov, F. Panolan, and S. Saurabh, *Efficient computation of representative families with applications in parameterized and exact algorithms*, J. ACM, 63 (2016), 29.
[25] F. V. Fomin, D. Lokshtanov, and S. Saurabh, *Bidimensionality and geometric graphs*, in Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, SIAM, 2012, pp. 1563–1575.

[26] M. Grohe, *Local tree-width, excluded minors, and approximation algorithms*, Combinatorica, 23 (2003), pp. 613–632.

[27] R. Impagliazzo and R. Paturi, *On the complexity of k-SAT*, J. Comput. System Sci., 62 (2001), pp. 367–375.

[28] P. N. Klein and D. Marx, *A subexponential parameterized algorithm for subset TSP on planar graphs*, in Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, SIAM, 2014, pp. 1812–1830.

[29] N. Linial and M. E. Saks, *Low diameter graph decompositions*, Combinatorica, 13 (1993), pp. 441–454, https://doi.org/10.1007/BF01303516.

[30] R. J. Lipton and R. E. Tarjan, *Applications of a planar separator theorem*, SIAM J. Comput., 9 (1980), pp. 615–627, https://doi.org/10.1137/0209046.

[31] D. Marx, M. Pilipczuk, and M. Pilipczuk, *On subexponential parameterized algorithms for Steiner tree and directed subset TSP on planar graphs*, in Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2018, 2018, pp. 474–484.

[32] D. Marx, M. Pilipczuk, and M. Pilipczuk, *A subexponential parameterized algorithm for directed subset traveling salesman problem on planar graphs*, SIAM J. Comput., 51 (2022), pp. 254–289, https://doi.org/10.1137/19m1304088.

[33] J. Matoušek and R. Thomas, *On the complexity of finding iso- and other morphisms for partial k-trees*, Discrete Math., 108 (1992), pp. 343–364, https://doi.org/10.1016/0012-365X(92)90687-B.

[34] M. Naor, L. J. Schulman, and A. Srinivasan, *Splitters and near-optimal derandomization*, in Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science, FOCS 1995, 1995, pp. 182–191.

[35] M. Pilipczuk, M. Pilipczuk, P. Sankowski, and E. J. van Leeuwen, *Subexponential-time parameterized algorithm for Steiner tree on planar graphs*, in Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, LIPIcs Int. Proc. Inform. 20, Schloss Dagstuhl — Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2013, pp. 353–364.

[36] M. Pilipczuk, M. Pilipczuk, P. Sankowski, and E. J. van Leeuwen, *Network sparsification for Steiner problems on planar and bounded-genus graphs*, ACM Trans. Algorithms, 14 (2018), 53.

[37] N. Robertson and P. D. Seymour, *Graph minors.* III. *Planar tree-width*, J. Combin. Theory Ser. B, 36 (1984), pp. 49–64.

[38] N. Robertson and P. D. Seymour, *Graph minors.* XIII. *The disjoint paths problem*, J. Combin. Theory Ser. B, 63 (1995), pp. 65–110.

[39] I. Sau and D. M. Thilikos, *Subexponential parameterized algorithms for degree-constrained subgraph problems on planar graphs*, J. Discrete Algorithms, 8 (2010), pp. 330–338, https://doi.org/10.1016/j.jda.2010.02.002.

[40] S. Tazari, *Algorithmic Graph Minor Theory: Approximation, Parameterized Complexity, and Practical Aspects*, Ph.D. thesis, Humboldt University of Berlin, 2010.

[41] S. Tazari, *Faster approximation schemes and parameterized algorithms on (odd-)H-minor-free graphs*, Theoret. Comput. Sci., 417 (2012), pp. 95–107, https://doi.org/10.1016/j.tcs.2011.09.014.