A Near-Optimal Planarization Algorithm*

Bart M. P. Jansen† Daniel Lokshtanov‡ Saket Saurabh§

October 31, 2013

Abstract

The problem of testing whether a graph is planar has been studied for over half a century, and is known to be solvable in $\mathcal{O}(n)$ time using a myriad of different approaches and techniques. Robertson and Seymour established the existence of a cubic algorithm for the more general problem of deciding whether an n-vertex graph can be made planar by at most k vertex deletions, for every fixed k. Of the known algorithms for k-Vertex Planarization, the algorithm of Marx and Schlotter (WG 2007, Algorithmica 2012) running in time $2^{k^{\mathcal{O}(k^3)}} \cdot n^2$ achieves the best running time dependence on k. The algorithm of Kawarabayashi (FOCS 2009), running in time $f(k)n$ for some $f(k) \in \Omega\left(2^{k^{\Omega(k^3)}}\right)$ that is not stated explicitly, achieves the best dependence on n.

In this paper we present an algorithm for k-Vertex Planarization with running time $2^{\mathcal{O}(k \log k)} \cdot n$, significantly improving the running time dependence on k without compromising the linear dependence on n. Our main technical contribution is a novel scheme to reduce the treewidth of the input graph to $\mathcal{O}(k)$ in time $2^{\mathcal{O}(k \log k)} \cdot n$. It combines new insights into the structure of graphs that become planar after contracting a matching, with a Baker-type subroutine that reduces the number of disjoint paths through planar parts of the graph that are not affected by the sought solution. To solve the reduced instances we formulate a dynamic programming algorithm for Weighted Vertex Planarization on graphs of treewidth w with running time $2^{\mathcal{O}(w \log w)} \cdot n$, thereby improving over previous double-exponential algorithms.

While Kawarabayashi's planarization algorithm relies heavily on deep results from the graph minors project, our techniques are elementary and practically self-contained. We expect them to be applicable to related edge-deletion and contraction variants of planarization problems.

*This work was supported by the European Research Council through Starting Grant 306992 "Parameterized Approximation", and by the Bergen Research Foundation and the University of Bergen through project "BeHard".

†University of Bergen, Norway.

‡University of Bergen, Norway.

§Institute of Mathematical Sciences, Chennai, Inda.

1 Introduction

Planarity testing algorithms, and the underlying characterizations of planar graphs, have been a popular subject of study for over half a century. A series of subsequent improvements has brought the running time of planarity testing all the way down to the optimal $\mathcal{O}(n)$, which was first achieved by Hopcroft and Tarjan [19].

The Vertex Planarization problem can be considered as a generalization of planarity testing. For a given graph G the goal is to find a minimum-size vertex set whose removal makes the resulting graph planar. Besides having obvious applications to graph visualization, the planarization problem plays an important role when attempting to generalize the good algorithmic properties of planar graphs to larger graph families.

We say that a graph G is k-apex if there is a vertex set S (the *apices*) of size at most k, such that $G - S$ is planar. For each fixed k, the k-apex graphs form a generalization of planar graphs that is orthogonal to the generalization into bounded genus graphs: 1-apex graphs can have arbitrarily high genus, while graphs of genus one might require arbitrarily many apices to make them planar. As observed by Kawarabayashi [24], many positive algorithmic results on planar graphs such as approximation schemes can be lifted to k-apex graphs if the set of apices is known. This forms another motivation for our interest in the planarization problem.

As planarity is a nontrivial and hereditary graph property, the general NP-completeness result for node deletion problems by Lewis and Yannakakis [27] shows that the decision version of Vertex Planarization is NP-complete. We therefore study the problem within the framework of parameterized complexity, hoping to find an algorithm that is efficient when keeping the desired size k of the solution fixed. A parameterized problem is a subset $Q \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. The second component of a tuple $(x, k) \in \Sigma^* \times \mathbb{N}$ is called the *parameter*. It measures some characteristic of the instance. The aim is to restrict the exponential behavior of the running time to a function depending solely on k. A problem is *fixed-parameter tractable* if there is an algorithm that decides membership of an instance (x, k) in $f(k)|x|^{\mathcal{O}(1)}$ time for some computable function f.

For every fixed k, the k-apex graphs form a lower ideal in the minor order [14, Theorem 6]. By the deep results of Robertson and Seymour's Graph Minor Project, this proves the existence of a nonuniform $\mathcal{O}(n^3)$-time algorithm to test whether a graph is k-apex. An explicit uniform algorithm can be constructed using the self-reduction technique of Fellows and Langston [15], but the constants involved in this approach are enormous. Simpler fixed-parameter tractable algorithms for the problem were given by Marx and Schlotter [30], and later by Kawarabayashi [24]. The algorithm of Marx and Schlotter [29, 30], running in time $2^{k^{\mathcal{O}(k^3)}} \cdot n^2$, achieves the best running time dependence on k. The algorithm of Kawarabayashi [24], running in time $f(k)n$ for some $f(k) \in \Omega\left(2^{k^{\Omega(k^3)}}\right)$ that is not stated explicitly, achieves the best dependence on n. As noted [7, Remark 1.2] by Chekuri et al. in their recent paper on approximation algorithms for planarization problems, no fixed-parameter tractable algorithm with a single-exponential parameter dependence was previously known for k-Vertex Planarization.

Our contribution. The main purpose of this work is to give a faster algorithm for k-Vertex Planarization, whose dependence on k is asymptotically almost optimal, and which is based purely on elementary techniques. We study the following construction version of the problem.

	k-Vertex Planarization
Input:	A graph G and integer k.
Parameter:	k.
Output:	A minimum-size apex set in G, or NO if G is not k-apex.

We present an algorithm for this problem that runs in $2^{\mathcal{O}(k \log k)} \cdot n$ time, thus improving the doubly-exponential dependence on k in the running time of Kawarabayashi's algorithm while preserving linearity in n. Assuming the *Exponential-Time Hypothesis* [20], there is no algorithm that solves k-Vertex Planarization in $2^{o(k)} n^{\mathcal{O}(1)}$ time, as this would imply the existence of a $2^{o(n)} n^{\mathcal{O}(1)}$-time algorithm for Vertex Cover on n-vertex graphs through a simple reduction. Hence the parameter-dependence $2^{\mathcal{O}(k \log k)}$ of our running time is near-optimal. Clearly, the linear dependence on n cannot be improved, making our overall algorithm near-optimal.

Our presentation of both the algorithm and its correctness proof is practically self-contained, not requiring anything more involved than planarity testing [19], decomposing a graph into three-connected components [18], and computing $\mathcal{O}(k)$-width tree decompositions of k-outerplanar graphs [5, 13]. Kawarabayashi's algorithm, on the other hand, relies heavily on techniques developed by Reed et al. [34, 33] in the context of the Graph Minors project to solve the k-Disjoint Paths problem in linear time for planar graphs, along with Bodlaender's algorithm [4] for computing tree decompositions in linear time.

We present the skeleton of the algorithm to discuss its novelties, postponing the details to Section 3. Our first step is to apply a simple pruning rule that removes superfluous copies of low-degree false twins from the graph G. Then we compute a maximal matching M in the remaining graph. If that graph is k-apex, then M has size at least $n/(2^k k^{\mathcal{O}(1)})$. Hence contracting the matching geometrically shrinks the instance. We contract M and recursively compute an apex set S_M for the resulting graph G/M. We lift S_M to an approximate apex set S in G by an algorithmic process. We then reduce the treewidth of the input graph by exploiting the structure that S reveals in the planar graph $G - S$. The final solution is found by dynamic programming on the reduced bounded-treewidth graph.

The global structure that we just described is exactly the same for Kawarabayashi's algorithm. Let us highlight the key differences in the implementation of the various steps.

- The extended abstract of Kawarabayashi [24] describes how to reduce the treewidth of the input graph when a recursively found set S of size $2k$ is provided. A remark is made that $G - S$ is in fact not planar, but *matching contractible* [1] — i.e., there is a matching whose contraction leaves the graph $G - S$ planar. Kawarabayashi states that adapting the treewidth reduction algorithms to work for this setting is the main technical challenge in designing an algorithm for k-Vertex Planarization. We give a graph-theoretical argument about the structure of matching-contractible graphs that allows us to lift a k-apex set in G/M to a $6k$-apex set in G, in $2^{\mathcal{O}(k \log k)} \cdot n$ time. This allows us to perform the treewidth reduction step in planar graphs, rather than matching-contractible graphs; a conceptually simpler setting.

- Kawarabayashi relies on a result by Mohar [31] concerning face covers of embedded graphs, to detect vertices in an approximate apex set that have to be part of every k-apex set; these are called *universal apices*. Our approach differs in two ways. First of all, rather than reasoning about one vertex at a time, we perform a branching step to guess how an optimal solution S^* intersects the approximate apex set S. Then we give a self-contained proof that if our guess is correct, there is a series of $\mathcal{O}(k)$ constant-radius balls in the planar graph $G - S$ that contains all of $N_G(S \setminus S^*)$. Our tight argument here is crucial for getting the treewidth down to $\mathcal{O}(k)$.

- Once the universal apex set is known, the existing algorithm applies the techniques of Reed et al. [34, 33] to quickly find irrelevant vertices that are contained deeply within large planar walls in $G - S$. Effectively, the algorithm applies a reduction rule that deletes a vertex v if it is surrounded by $2k + 2$ nested, vertex-disjoint cycles whose removal leaves the component containing v planar. Such vertices are irrelevant to the planarization problem. There are examples that show that exhaustive application of this rule, together with the universal apex identification, cannot reduce the treewidth of the graph to $o(k\sqrt{k})$. We introduce a new reduction strategy, utilizing Baker's layering technique [3], to reduce the treewidth to the optimal $\mathcal{O}(k)$. Rather than identifying irrelevant vertices that are insulated from the nonplanar parts of the graph by $2k+2$ nested cycles, we partition the outerplanarity layers of the planar graph $G-S$ into $k+1$ groups, and branch into $k+1$ ways to guess which of these $k+1$ groups does not contain vertices from the sought solution. For the guessed group, we forbid all vertices to be deleted. We then prove that the presence of 2 insulating cycles whose vertices are forbidden to be deleted, already causes a vertex to be irrelevant. Hence the number of insulating cycles is reduced from $2k+2$ to 2, allowing the treewidth to be reduced below $\mathcal{O}(k\sqrt{k})$. Consequently, we have to deal with vertices that are forbidden to be deleted by a solution, in the remainder of the algorithm.

- Kawarabayashi uses Bodlaender's algorithm [4] to construct a tree decomposition of the remaining bounded-treewidth graph. This algorithm is based on a very complex dynamic programming procedure. Our algorithm uses a much simpler approach to compute a small-width tree decomposition of the resulting graph. We use flow techniques to find small separators between distant parts of the planar graph, which leave only planar pieces of small diameter. Tree decompositions for planar graphs of small diameter are well-understood, featuring prominently in modern interpretations of Baker's approach [3].

- When it comes to solving the problem on a bounded-treewidth graph, Kawarabayashi takes the same approach as Marx and Schlotter [30]. They mention that the problem may be solved in linear-time on graphs of treewidth w by applying Courcelle's Theorem [2, 6, 8, 9], or by a dynamic programming step with running time $2^{w^{\Omega(w)}} \cdot n$. The intuition behind this dynamic programming algorithm is to keep track of possible Kuratowski minors in the graph after deleting at most k vertices. We give a $2^{\mathcal{O}(w \log w)} \cdot n$ time algorithm for (the weighted version of) Vertex Planarization on bounded treewidth graphs, thereby answering an open problem posed by Marx [28]. The states of the dynamic program are based on possible embeddings of the planar subgraph. A similar type of dynamic program is employed by Kawarabayashi, Mohar, and Reed [25] to compute the genus of a graph of bounded treewidth.

In summary, our algorithm uses elementary and more transparent methods to solve the k-Vertex Planarization problem much faster than before. Under the Exponential Time Hypothesis [20], our running time is almost optimal. We expect that the insights gained by this approach will prove to be useful in the future.

Organization. The paper is structured as follows. After presenting the necessary preliminaries, we give a more detailed outline of our algorithm in Section 3. Subsequent sections show how to implement each step. Finally, Section 7 shows how the pieces combine into a formal correctness proof for the algorithm.

2 Preliminaries

Graphs. All graphs we consider are simple, undirected and finite, unless explicitly stated otherwise. We use $[n]$ as an abbreviation for $\{1, \ldots, n\}$. For a graph G we denote its vertex set by $V(G)$ and the edge set by $E(G)$. An edge between vertices u and v is denoted by uv, and is identical to the edge vu. We use $G[V']$ to denote the subgraph of G induced by V', i.e., the graph on vertex set V' and edge set $\{uv \in E(G) \mid u, v \in V'\}$. We use $G - Z$ as an abbreviation for $G[V(G) \setminus Z]$. For singleton sets $\{v\}$ we write $G - v$ instead of $G - \{v\}$. The open neighborhood of a vertex v in graph G contains the vertices adjacent to v, and is written as $N_G(v)$. The open neighborhood of a set $S \subseteq V(G)$ is defined as $\bigcup_{v \in S} N_G(v) \setminus S$. The degree is a vertex v is $\deg_G(v)$. Two vertices u and v are *false twins* in graph G if $N_G(u) = N_G(v)$. A *twin class* is a maximal set of vertices with the same open neighborhood. A *leaf* is a vertex of degree at most one. Contracting an edge uv in graph G is the operation of identifying the endpoints and removing any resulting parallel edges or self-loops. For an edge set $Y \subseteq E(G)$ we write G/Y for the graph obtained by contracting the edges in Y one by one — note that the order in which the edges are contracted does not affect the final result. We use G/uv for the graph obtained by contracting the single edge uv.

Consider the graph $H := G/Y$ obtained from G by contracting the edges in Y. The *pre-image* of a vertex set $S \subseteq V(H)$, denoted by $S/^{-1}Y$, contains the vertices in $V(G) \cap S$ which were unaffected by the contractions, together with the endpoints of the edges in Y whose contraction resulted in a vertex in S.

Given a graph G with disjoint sets X and Y, a set $S \subseteq V(G) \setminus (X \cup Y)$ is an *XY-separator* if there is no connected component of $G - S$ that contains both a vertex of X and a vertex of Y. If $X = \{u\}$ and $Y = \{v\}$ such a set is simply called a *uv-separator*. We say that a tuple of XY-separators (S_1, S_2) is *nested* if $S_1 \cap S_2 = \emptyset$ and S_2 is an XS_1-separator.

A *cutvertex* is a vertex whose removal increases the number of connected components. A graph is *biconnected* if it is connected and does not contain any cutvertices. The *biconnected components* of a graph G are its maximal biconnected subgraphs. It is well known that the biconnected components of a graph form a partition of its edge set. Similarly, a connected graph is *triconnected* (or 3-connected) if there is no set of at most two vertices whose removal disconnects the graph.

Minors. A graph G has H as a minor if and only if H can be obtained from G by a (possibly empty) sequence of the following operations: edge deletion, edge contraction, and vertex deletion. A *minor model* of a graph H in a graph G is a mapping ϕ from $V(H)$ to subsets of $V(G)$ (called *branch sets*) which satisfies the following conditions: (a) $\phi(u) \cap \phi(v) = \emptyset$ for distinct $u, v \in V(H)$, (b) $G[\phi(v)]$ is connected for $v \in V(H)$, and (c) there is an edge between a vertex in $\phi(u)$ and a vertex in $\phi(v)$ for all $uv \in E(H)$. It is well-known that G is planar if and only if it does not contain any of the two *Kuratowski graphs* K_5 and $K_{3,3}$ as a minor.

Planarity. A graph is *planar* if it can be drawn in the plane without crossing edges. An *apex set* in G is a set of vertices S such that $G - S$ is planar. A *k-apex set* is simply an apex set of size at most k. A graph G is *k-apex* if it has a k-apex set. A *plane* graph is a planar graph together with an embedding, which can be represented combinatorially by a rotation system. The *radial graph* (or *face-vertex incidence graph*) corresponding to a plane graph G with face set F, is the planar bipartite graph on vertex set $V(G) \cup F$ with an edge between $v \in V(G)$ and $f \in F$ whenever v lies on F.

Proposition 1. *Let S be a vertex set in a graph G. If G has an apex set S' disjoint from S, and G' is a minor of G that can be obtained without contracting edges that are incident on S, then G' has an apex set S'' disjoint from S that is not bigger than S'.*

Proof. We can update the apex set S' while transforming G into G', to obtain an apex set S'' for G' that is disjoint from S and not bigger than S'. So consider a series of minor operations, not involving the contraction of edges incident on S, that take G to G'. It is obvious how to update S' when deleting vertices or edges. When contracting an edge pq into its endpoint q, do the following. If $p \in S'$ then replace it by q, otherwise do nothing. Performing these update steps for all minor operations, it is straight-forward to verify that for the resulting set S'', the graph $G' - S''$ is a minor of $G - S'$ and is therefore planar. As we do not contract edges incident on S, we never add a vertex from S to our apex set, thereby keeping it disjoint from S. As we remove a vertex from the apex set whenever we add a new vertex to it, $|S''| \leq |S'|$. $\square$

Given a plane graph G and two vertices u and v, we define the *radial distance* $d_G(u, v)$ between u and v in G to be one less than the minimum length of a sequence of vertices that starts at u, ends in v, and in which every two consecutive vertices lie on a common face. For example, the radial distance from v to itself is zero, and the distance to any vertex that shares a face with v is one. For $X, Y \subseteq V(G)$ define $d_G(X, Y) := \min_{x \in X, y \in Y} d_G(x, y)$. For a vertex set $S \subseteq V(G)$ of a plane graph, define $\mathbf{R}^r_G(S)$ as the set of vertices of G whose radial distance from some vertex of S is at most r. We will also refer to $\mathbf{R}^r_G(S)$ as the *r-ball* around S. The radial distance between u and v is exactly half the number of edges on a shortest uv-path in the radial graph of G, and therefore breadth-first search in the radial graph can be used to find an r-ball efficiently. In a connected graph, any two vertices on a common face are connected by a path over vertices incident with the same face. This implies that for a connected plane graph G, any choice of vertex v and radius r leads to a connected subgraph $G[\mathbf{R}^r_G(\{v\})]$. A *radial path* is a sequence of vertices in which successive elements lie on a common face. A plane graph is a *triangulation* if every face is a triangle.

Observation 1. Let G be a plane graph, X be a connected vertex subset and S be a vertex set such that $\mathbf{R}^i_G(X) \cap S = \emptyset$, for some non-negative integer i. Then $\mathbf{R}^i_G(X) = \mathbf{R}^i_{G-S}(X)$.

Proposition 2 ([32, Prop. 8.2.3])**.** *Let u and v be distinct vertices in a plane triangulation G. If S is a minimal uv-separator then $G[S]$ is an induced, nonfacial cycle.*

While the previous lemma was originally formulated for a minimal set of vertices whose removal increases the number of connected components, the same proof shows that it also holds for minimal uv-separators.

Proposition 3. *Let G be a plane graph with disjoint vertex sets X and Y of size at least one, such that $G[X]$ and $G[Y]$ are connected and $d_G(X, Y) = d \geq 2$. For any r with $0 < r < d$ there is a cycle C such that all vertices $u \in V(C)$ satisfy $d_G(X, \{u\}) = r$, and such that $V(C)$ is an XY-separator in G.*

Proof. Assume the stated conditions hold. It is not difficult to verify that we can construct a plane triangulated supergraph G' of G by repeatedly inserting a vertex into the interior of a face, and making it adjacent to the vertices on the boundary of that face. Observe that if G is not biconnected, then we might have to insert multiple vertices into a face. We end up with a plane triangulation G' where no edges have been inserted between vertices that were originally in G.

Now consider the vertices S_r in G which have radial distance exactly r from X in G. That is, $S_r = \{w \in V(G) \mid d(X, \{w\}) = r\}$. We will argue that S_r is an XY-separator in G'. Consider a path P in G' with start vertex $x \in X$ and the last vertex $y \in Y$ such that no internal vertices of P are in $X \cup Y$. Consider the original vertices from $V(G)$ occurring on the path in G', and assume $x_1, x_2 \in V(G)$ are visited by the path while only visiting newly inserted vertices in between. Since we have only inserted vertices inside existing faces, x_1 and x_2 lie on a common face in G and therefore $d_G(x_1, x_2) \leq 1$. Since $d_G(x, x) = 0$ and $d_G(x, y) \geq d$, the path P contains at least one vertex in S_r. Therefore S_r is an XY-separator in G'. Let $a \in X$ and $b \in Y$ then S_r is an ab-separator in G'. Now take a minimal subset S'_r of S_r which is still an ab-separator in G'. By Proposition 2 the set S'_r induces a cycle in G'. As all vertices in S'_r also exist in G, and we did not add new edges between existing vertices of G when creating G', it follows that $G[S'_r]$ is a cycle. As G is a subgraph of G', and S'_r is an ab-separator in G', it is also an ab-separator in G. Further, S'_r is disjoint from $X \cup Y$ and both X and Y are connected sets in G. Thus, all of X lies in the same component as a in $G - S'_r$ and all of Y lies in the same component as b. Hence, S'_r is an XY-separator inducing a cycle in G satisfying $d_G(X, \{u\}) = r$ for all vertices $u \in S'_r$. ∎

For a graph G and vertex $v \in V(G)$ we denote by $\mathcal{C}_G(v)$ the connected component of G containing v. If $S \subseteq V(G) \setminus \{v\}$ then we define $G^S(v)$ as $G[S \cup \mathcal{C}_{G-S}(v)]$, i.e., the separator together with the connected component of $G - S$ that contains v. We say that S *planarizes* v if $G^S(v)$ is planar. A *v-planarizing cycle* in G is a cycle such that $V(C)$ planarizes v. A tuple of v-planarizing cycles (C_1, C_2) is *nested* if $V(C_1) \cap V(C_2) = \emptyset$ and $V(C_2)$ separates v from $V(C_1)$. A *matching contractible* graph G is a graph containing a matching $M \subseteq E(G)$ such that G/M is planar.

Let C be a cycle in graph G. A *C-bridge* in G is a subgraph of G which is either a chord of C, or a connected component B of $G - V(C)$ together with all edges between B and C, and their endpoints. If B is a C-bridge, then the vertices $V(B) \cap V(C)$ are the *attachments of B*. Two C-bridges B_1, B_2 *overlap* if at least one of the following conditions is satisfied: (a) B_1 and B_2 have at least three attachments in common, or (b) the cycle C contains distinct vertices a, b, c, d (in this cyclic order) such that a and c are attachments of B_1, while b and d are attachments of B_2. For a graph G with a cycle C, the corresponding *overlap graph* $O(G, C)$ has the C-bridges in G as its vertices, with an edge between two bridges if they overlap. The following characterization of planar graphs will be essential to our arguments.

Lemma 1 ([10, Thm. 3.8])**.** *A graph G with a cycle C is planar if and only if the following two conditions hold:*

- *For each C-bridge B in G, the graph $B \cup C$ is planar.*
- *The overlap graph $O(G, C)$ is bipartite.*

Although the preceding lemma was originally stated only for biconnected graphs, it is easily seen to be true for all graphs. The following proposition shows that the overlap graph is not affected by the contraction of edges between vertices that do not lie on C.

Proposition 4. *Let G be a graph containing a cycle C. If e is an edge of G whose endpoints do not lie on C, then contraction of e does not change the overlap graph: $O(G, C) = O(G/e, C)$.*

Proof. Consider the C-bridges in G. Observe that for any C-bridge B that does not contain e, it is also a C-bridge in G/e with the same attachments. As the endpoints of e do not lie on C, the edge is contained entirely within one C-bridge B_e that consists of a connected component of $G - V(C)$ on at least two vertices, together with its edges to C. After contraction of e there is still at least one vertex left in the connected component of $G/e - V(C)$, and the attachments of B' on C are not affected by the contraction. Hence all the C-bridges of G/e correspond uniquely to C-bridges in G with the same attachments. As the overlap graph is determined only by the set of C-bridges and their attachments it follows that $O(G, C) = O(G/e, C)$. $\square$

Proposition 5 (Cf. [10, Sect. 3.3])**.** *A graph is planar if and only if all its biconnected components are planar.*

Euler's formula ([11, Theorem 4.2.9])**.** *Any plane graph with n vertices, m edges, and f faces satisfies $n - m + f \geq 2$.*

Elementary manipulation of Euler's formula gives well-known bounds that we will often use. A simple planar graph on n vertices has at most $3n$ edges, and has average degree less than six. If the planar graph is also bipartite, it has no more than $2n$ edges.

Proposition 6. *Let G be a triconnected planar graph and let $S \subseteq V(G)$ be a nonempty vertex set. Then the number of connected components in $G - S$ is at most $2|S|$.*

Proof. Consider the connected components $C_1, \ldots, C_\ell$ of $G - S$. If $\ell = 1$ then the statement is true as $2|S| \geq 2$. In the remainder we therefore assume that $\ell \geq 2$.

Build an auxiliary bipartite graph $\hat{G}$ where S is one partite set, and the other partite set has one vertex for each component C of $G \setminus S$. In $\hat{G}$ we make a vertex u in S adjacent to a vertex corresponding to a component C if $u \in N_G(C)$. Since $\hat{G}$ is a minor of G it follows that $\hat{G}$ is planar. Each component has at least three neighbors in S, as the triconnectedness of G implies that there are three vertex-disjoint paths between each pair of components. As there are ℓ components, the number of edges in $\hat{G}$ is at least 3ℓ. The number of vertices is $|S| + \ell$. As Euler's formula ensures $|E(\hat{G})| \leq 2|V(\hat{G})|$ for planar simple bipartite graphs $\hat{G}$, we find that $3\ell \leq 2(|S|+\ell)$. This yields $\ell \leq 2|S|$ as a bound on the number of components of $G-S$. $\square$

Lemma 2 (Cf. [21, Lemma 1])**.** *Let G be a planar graph and let X be a vertex cover in G. Then $|\{N_G(v) \mid v \notin X\}| \leq 6|X| + 1$.*

Proof. Assume the conditions stated in the lemma hold, and let $Y := V(G) \setminus X$ which is an independent set. We count the number of open neighborhoods in three steps. Define $Y_{\leq 1} := \{N_G(v) \mid v \notin X \text{ and } \deg_G(v) \leq 1\}$. Similarly let Y_2 be the set of open neighborhoods that have size two each, and let $Y_{\geq 3}$ be the set of open neighborhoods of size three each. The quantity we need to bound is $|Y_{\leq 1}| + |Y_2| + |Y_{\geq 3}|$. Obviously there are at most $|X| + 1$ distinct open neighborhoods of one vertex or less, so $|Y_{\leq 1}| \leq |X| + 1$.

To bound $|Y_2|$ consider the graph G_2 on vertex set X with an edge between $u, v \in X$ if there is a vertex $y \in Y$ such that $N_G(y) = \{u, v\}$. The resulting graph is a minor of G and therefore planar, and every neighborhood in Y_2 corresponds to a unique edge in G_2. By Euler's formula the number of edges in a simple planar graph is at most three times the number of vertices, and we find that $|Y_2| = |E(G_2)| \leq 3|X|$.

To bound the size of $Y_{\geq 3}$ we consider the bipartite graph G_3 that is the subgraph of G obtained by removing the edges in $G[X]$, removing the vertices in Y that have degree less than three, and keeping exactly one vertex in each twin class in the remainder of Y. The resulting bipartite graph is planar, with one partite set corresponding to X while the other partite set has a vertex for every open neighborhood in $Y_{\geq 3}$. The number of vertices in G_3 is therefore $|X| + |Y_{\geq 3}|$. Another application of Euler's formula shows that the number of edges in a simple bipartite planar graph on n vertices is at most $2n$. Since every vertex in $Y_{\geq 3}$ has degree at least 3 in G_3 we find $|E(G_3)| \geq 3|Y_{\geq 3}|$. Combining this with the upper bound on the edge count we find $3|Y_{\geq 3}| \leq |E(G_3)| \leq 2(|X| + |Y_{\geq 3}|)$, which implies that $|Y_{\geq 3}| \leq 2|X|$.

Since $|\{N_G(v) \mid v \notin X\}| = |Y_{\leq 1}| + |Y_2| + |Y_{\geq 3}|$ the bound of $6|X| + 1$ follows. $\square$

For more background on planar graphs we refer to the monograph of Mohar and Thomassen [32].

Treewidth. A *tree decomposition* of a graph G is a pair $(T, \mathcal{X})$, where T is a tree and $\mathcal{X} = \{X_i \mid i \in V(T)\}$ is a collection of subsets of $V(G)$ called *bags* such that the following conditions are satisfied:

1. $\bigcup_{i \in V(T)} X_i = V(G)$.
2. For each edge $uv \in E(G)$ there is a node $i \in V(T)$ such that $\{u, v\} \subseteq X_i$.
3. For each $v \in V(G)$ the nodes $\{i \mid v \in X_i\}$ induce a connected subtree of T.

The width of the tree decomposition is $\max_{i \in V(T)} |X_i| - 1$. The treewidth of a graph G, denoted by $\textsc{tw}(G)$, is the minimum width over all tree decompositions of G. A *nice tree decomposition* is a tree decomposition $(T, \mathcal{X})$ where T is a rooted tree and the following conditions are satisfied:

1. Every node of the tree T has at most two children.
2. If a node t has two children t_1 and t_2, then $X_t = X_{t_1} = X_{t_2}$.

3. If a node t has one child t_1, then either $|X_t| = |X_{t_1}| + 1$ and $X_{t_1} \subset X_t$ (in this case we call t_1 *insert node*) or $|X_t| = |X_{t_1}| - 1$ and $X_t \subset X_{t_1}$ (in this case we call t_1 *insert node*).

4. For the root node r of T, $X_r = \emptyset$. For every leaf node i of T, $X_i = \emptyset$.

It is possible to transform a given tree decomposition $(T, \mathcal{X})$ into a nice tree decomposition $(T', \mathcal{X}')$ of the same width in time $\mathcal{O}(n+m)$ [4]. An *adhesion point* of a tree decomposition is a vertex that occurs in more than one bag. The set of all adhesion points of a tree decomposition $(T, \mathcal{X})$ is denoted $\mathrm{Adh}(T, \mathcal{X})$.

Given an embedding of a graph G in the plane, define its *outerplanarity layers* $L_i \subseteq V(G)$ as follows. Layer L_0 contains the vertices incident with the outer face. For $i > 0$ layer L_i contains the vertices which lie on the outer face after deleting the vertices in $\bigcup_{j=0}^{i-1} L_j$ and their incident edges from the embedding. We use $\mathrm{rad}(G)$ to denote the largest index of a non-empty outerplanarity layer of a plane graph G.

An embedding of a graph G is *k-outerplanar* if it consists of at most k nonempty outerplanarity layers, which is equivalent to saying that $\mathrm{rad}(G) \leq k-1$. A graph is k-outerplanar if it has a k-outerplanar embedding. For a plane graph G we define the *outerplanarity index* of a vertex v to be index of the outerplanarity layer containing v.

Observation 2. Let u and v be vertices in a plane graph G. If u and v lie on a common face, then the outerplanarity indices of u and v differ by at most one. Hence if the radial distance between u and v is d, their outerplanarity indices differ by at most d.

Proposition 7. *Let u and v be vertices in a plane graph G, and let $S \subseteq V(G) \setminus \{u, v\}$. If we restrict the embedding of G to the subgraph $G - S$, then the resulting pair of plane graphs satisfies $d_{G-S}(u, v) \leq d_G(u, v) \leq d_{G-S}(u, v) - |S|$.*

Bodlaender proved a bound on the treewidth of k-outerplanar graphs [5, Theorem 83]. It has often been noted in the literature that his proof can actually be turned into a linear-time algorithm (cf. [22, 23]).

Proposition 8 ([5, 13, 23])**.** *Given a k-outerplanar embedding of a graph G on n vertices, a tree decomposition of width at most $3k-1$ can be constructed in $\mathcal{O}(kn)$ time.*

Proposition 9. *Let G be a graph on n vertices with a matching M. Given a width-w tree decomposition $(T_M, \mathcal{X}_M)$ of G/M, we can construct a tree decomposition $(T, \mathcal{X})$ of G having width at most $2w+1$ in $\mathcal{O}(wn)$ time.*

Proof. For each bag $\mathcal{X}_i$ in the tree decomposition, for each vertex $v \in \mathcal{X}_i$ which is the result of contracting an edge $xy \in M$, remove v from X_i and replace it by x and y. In the worst case this doubles the number of vertices in each bag from $w+1$ to $2(w+1)$, thereby resulting in a tree decomposition of width at most $2(w+1) - 1 = 2w+1$. □

Tutte Decomposition. Special types of tree decompositions can be used to elegantly describe a partition of a graph into its triconnected components. The concept of a *torso* is needed to state the decomposition theorem. For a graph G and vertex set $M \subseteq V(G)$, the graph $\mathrm{Torso}(G, M)$ has vertex set M. Two vertices u and v have an edge between each other in $\mathrm{Torso}(G, M)$ if $uv \in E(G[M])$, or there is a path from u to v in G with all interior vertices in $V(G) \setminus M$.

Definition 1. A *Tutte decomposition* of a graph G is a pair $(F, \mathcal{X})$, where F is a forest and $\mathcal{X} = \{X_i \subseteq V(G) \mid i \in V(F)\}$ is a set of bags, with the following properties:

(a) Each tree T_i in F can be associated to a unique connected component C_i in G, such that $(T_i, \mathcal{X}_i := \{X_j \mid j \in V(T)\})$ is a tree decomposition of C_i.

(b) For each pair of distinct nodes $i, j \in V(F)$ we have $|X_i \cap X_j| \leq 2$.

(c) For each $i \in V(F)$ the graph $\text{Torso}(G, X_i)$ is triconnected.

(d) There is no pair of distinct nodes $i, j \in V(F)$ such that $X_i \subseteq X_j$.

For a set $S \subseteq V(G)$ we say that *the leaves of the Tutte decomposition* $(F, \mathcal{X})$ *intersect* S if, for each leaf i of the forest F, the bag X_i has a non-empty intersection with S.

This definition implies that a leaf bag of a Tutte decomposition contains a vertex that is not an adhesion point. Also, each leaf bag contains at most two adhesion vertices. We work with a forest of tree decompositions, rather than a single tree, to maintain the special status of leaves of the decomposition. In Tutte decompositions of connected graphs G, the leaves of the decomposition tree correspond to "dead ends" in G. Allowing the Tutte decomposition to be a forest, rather than a tree, maintains this property even for disconnected graphs. In contrast, when using a single decomposition tree for a disconnected graph, some bags that would be leaves in a decomposition of the connected component, might turn into internal vertices of the decomposition tree, if the trees of various connected components are glued together at that vertex.

When representing a Tutte decomposition in memory, we store some extra information besides the forest and the bags corresponding to the nodes, to allow computations on the decomposition to be performed efficiently. For each node $i \in V(F)$ we store the associated torso $\text{Torso}(G, X_i)$. For each edge $ij \in E(F)$, we store the adhesion $X_i \cap X_j$ between the two bags in the edge record.

Proposition 10. *If* $(F, \mathcal{X} = \{X_i \mid i \in V(F)\})$ *is a Tutte decomposition of an* n*-vertex graph* G*, then* $|V(F)| \in \mathcal{O}(n)$ *and* $\sum_{i \in V(F)} |X_i| \in \mathcal{O}(n)$*.*

Proposition 11. *If* $(T, \mathcal{X} = \{X_i \mid i \in V(T)\})$ *is a tree decomposition of a graph* G*, and* u, v *are adjacent nodes of* T*, then the vertices* $X_u \cap X_v$ *separate* $X_u \setminus X_v$ *from* $X_v \setminus X_u$ *in* G*.*

Proposition 12. *Let* $(F, \mathcal{X} = \{X_i \mid i \in V(F)\})$ *be a Tutte decomposition of a graph* G*, and let* u, v *be adjacent nodes of* F *with* $X_u \cap X_v = \{x, y\}$*. Let* T_u *be the connected component of* $F - v$ *that contains node* u*. Then there is a path* P_{xy} *in* G *between* x *and* y*, that only uses vertices of* $\bigcup_{i \in V(T_u)} X_i$*. In particular, the interior of* P_{xy} *does not contain vertices of* X_v*.*

Proof. If $xy \in E(G)$ then the proposition is trivial, so assume that this is not the case. By the definition of a Tutte decomposition, the graph $\text{Torso}(G, X_v)$ is triconnected. We must have $|X_u| \geq 3$, as otherwise the bag X_u is a subset of X_v, contradicting property (d) of a Tutte decomposition. Consider a vertex $z \in X_u \setminus X_v = X_u \setminus \{x, y\}$.

As a single vertex-deletion cannot disconnect the triconnected graph $\text{Torso}(G, X_v)$, there is a path P_{zx} from z to x in $\text{Torso}(G, X_u) - y$, and similarly a path P_{yz} from z to y in $\text{Torso}(G, X_u) - x$. By the definition of torso, each edge pq on such a path corresponds to an edge pq in G, or to a pq-path in G whose interior vertices avoid X_u. By Proposition 11, all paths from z to X_v contain one of the adhesion points x, y. Hence the simple paths P_{zx} and P_{yz} do not contain vertices from X_v in their interior, as to reach such vertices they would have to traverse an adhesion point, but the single remaining adhesion point is an endpoint of the relevant path. Hence P_{zx} and P_{zy} correspond to simple paths in G whose interior vertices avoid X_u. Moreover, all vertices on P_{zx} and P_{yz} belong to $\bigcup_{i \in V(T_u)} X_i$, since to reach a vertex from $z \in X_u \setminus X_v$ that only occurs in bags of nodes in $T - T_u$, one needs to cross the separator xy by Proposition 11. Hence we can concatenate the two paths to obtain a path between x and y in G on a vertex subset of $\bigcup_{i \in V(T_u)} X_i$, and whose interior avoids X_v. □

Proposition 12 easily implies the following statement.

Proposition 13. *If $(F, \mathcal{X} = \{X_i \mid i \in V(F)\})$ is a Tutte decomposition of a graph G, then $Torso(G, X_i)$ is a minor of G for each $i \in V(F)$.*

The following proposition follows from the fact that a Tutte decomposition of a graph G can be obtained by applying an algorithm for computing triconnected components to each of the connected components of G. Each connected component results in a tree in the Tutte decomposition.

Proposition 14 ([18, 17])**.** *There is an $\mathcal{O}(n+m)$-time algorithm that, given a graph G, outputs a Tutte decomposition $(F, \mathcal{X} = \{X_i \mid i \in V(F)\})$ of G.*

Data structures and elementary operations. Suitable data structures are needed to implement our algorithm in linear time. As these are fairly standard, we briefly comment on them in these preliminaries and do not mention them much in the core description of the algorithm. Throughout the algorithm we identify vertices in an n-vertex graph by integers in the range $[n]$. When removing vertices we leave the identifiers of the remaining vertices unchanged. We consider each undirected edge as consisting of two *half-edges*. We store a graph as an array that has a doubly-linked list for each vertex. The list for vertex v has a record for each half-edge leaving v. A half-edge record for a neighbor u of v stores pointers to the previous and next half-edges leaving v, and to the *twin* half-edge going from u to v.

This structure supports the removal of an edge uv in $\mathcal{O}(1)$ time, its contraction in $\mathcal{O}(\deg(u)+\deg(v))$ time, and the deletion of a vertex v in $\mathcal{O}(\deg(v))$ time. In general we cannot determine in constant time whether there is an edge between u and v, given their identifiers. When we require efficient adjacency queries between a vertex set S of size k and the rest of the graph, we facilitate these queries in constant time by explicitly constructing the rows of the adjacency matrix corresponding to S. This costs $\mathcal{O}(kn)$ initialization time and space. For d-degenerate graphs (i.e., graphs where each induced subgraph has a vertex of degree at most d) we can use another tool. By repeatedly finding a vertex of degree at most d, orienting its edges outward, and forgetting about the vertex, we obtain a a (static) *degree-d out-orientation* of the graph in $\mathcal{O}(n+m) = \mathcal{O}(n+nd)$ time. Since each vertex has at most d outgoing edges we can check the presence of an edge uv in $\mathcal{O}(d)$ time by testing the $2d$ outgoing edges. As planar graphs are 5-degenerate, any k-apex graph is $k+5$-degenerate and this can be tested in linear time.

We represent embedded planar graphs combinatorially by keeping track of the clockwise ordering of edges around a vertex. We store them in the same structure as general graphs, interpreting the ordering of half-edges in v's adjacency list as the clockwise ordering of these edges around v. Therefore we implement the contraction operation to preserve this ordering. Sometimes we associate a half-edge from u to v with the face seen on the left side when looking from u to v, and maintain a data field in the half-edge record to store information about this face. By well-known techniques this allows us to construct the radial graph corresponding to a plane graph G in $\mathcal{O}(n+m) = \mathcal{O}(n)$ time. The radial graph is convenient when performing *radial breadth first search* from a vertex set S in a plane graph G, i.e., a traversal that visits the vertices of G in increasing order of their radial distance to the nearest member of S, and labels vertices by this distance. Breadth-first search in the radial graph can be used to identify the outerplanarity layers of a plane graph G in linear time. Starting the breadth-first search at the vertex f_0 in the radial graph R_G that represents the outer face of G. The vertices of R_G correspond to vertices and faces of G. For a vertex in $V(R_G) \cap V(G)$, if its edge-distance from f_0 in R_G is ℓ, then its outerplanarity index in G is $\lfloor l/2 \rfloor$. As a BFS in R_G can be used to label every vertex in $V(R_G)$ by its edge-distance from f_0, this computation allows the outerplanarity layers to be identified efficiently.

3 Outline of the Algorithm

The global idea behind the algorithm is to compute an approximate apex set, whose structure is subsequently used to reduce the treewidth of the graph. Afterward we solve the problem by dynamic programming on a tree decomposition of bounded width. To achieve all this in $2^{\mathcal{O}(k \log k)} \cdot n$ time, we combine novel algorithmic tricks and graph-theoretical insights in each of these three steps. In the remainder of this section we give a more detailed description of the algorithm.

3.1 Finding an approximate apex set S in G

The first phase consists of finding an approximate apex set. The structure of this phase is reminiscent of Bodlaender's [4] linear-time algorithm for k-Treewidth, as it uses recursion on a subinstance that is obtained by contracting a matching. This technique was subsequently used in several other linear-time algorithms for graph problems [25, 24].

Reduce the sizes of degree-at-most-$k+13$ twin classes in G. Given an input graph G and integer k, we first apply a reduction rule to G that removes superfluous copies of false twins. In Section 4.1 we show that we may safely reduce the size of each twin class to $k+3$, as any planar graph which has three false twins remains planar when adding more copies of these twins. To allow this step to be implemented in $\mathcal{O}(kn)$ time, without resorting to complicated algorithms for modular decomposition, we work only on the twin classes whose vertices have degree at most $k+13$. We reduce the twin classes to size $k+13$ rather than $k+3$, to prevent the reduction rule from re-triggering itself.

Contract a large maximal matching M in G. If G is an n-vertex k-apex graph whose bounded-degree twin classes have been reduced, then any maximal matching in G has at least $\frac{n}{30 \cdot 2^k (k+13)}$ edges (Section 4.2). Hence by contracting such a matching M, the size of the instance decreases geometrically. As a maximal matching can be found in linear time, we can recurse on a smaller instance G/M. If it does not have a k-apex set, then the graph G does not have such a set either. Otherwise we find a k-apex set S_M in G/M.

Lift S_M to an approximate solution S for G. To lift the k-apex set of the graph G/M to the original input G, we employ a structural insight into apex sets of matching-contractible graphs. Given the apex set S_M in G/M, we first do the following. Each vertex in S_M that is also a vertex in G, is added to the approximate apex set. The other vertices of S_M are the result of contracting an edge in M; we add both endpoints of the edge to the approximate apex set. We end up with a set S_1 of size at most $2k$, which has the following property: the graph $G - S_1$ becomes planar by contracting the edges in $M - S_1$. Hence to find an approximate apex set in G, it then suffices to find an approximate apex set for the matching-contractible graph $G - S_1$. The fact that $(G - S_1)/(M - S_1)$ is planar allows us to use its topological structure when building the approximation.

In Section 4.3 we prove a purely graph-theoretic statement concerning apex sets of matching-contractible graphs. Given a graph G that becomes planar by contracting the matching M, an embedding of G/M gives a way to cover G by graphs of small treewidth, as follows. If we take a series of $\mathcal{O}(k)$ consecutive outerplanarity layers of G/M, we can consider the subgraph of G that it induces when decontracting the matching. As decontracting a matching can only double the treewidth, and k-outerplanar graphs have treewidth $\mathcal{O}(k)$, such subgraphs of G have treewidth $\mathcal{O}(k)$. The structural statement we prove is basically that if S is a set such that, for all sequences of $\mathcal{O}(k)$ consecutive outerplanarity layers of G/M, the subgraph of G obtained by decontracting the matching becomes planar after removing S, then in fact the entire graph G becomes planar after removing S. We interpret this result as saying that matching-contractible graphs which are locally planar, are also globally planar. This local nature of nonplanarities arising from decontracting a matching is also exploited in the algorithm of Kawarabayashi, Mohar, and Reed [25] for embedding a graph into a fixed surface.

This structural property of matching-contractible graphs allows us to find an approximate apex set in $(G-S_1)/(M-S_1)$, as follows. We cover the graph by subgraphs of treewidth $\mathcal{O}(k)$, such that each vertex is contained in at most two such subgraphs. In each subgraph we compute an optimal apex set using our dynamic programming algorithm for Weighted Vertex Planarization on Bounded Treewidth Graphs. We then take the union S_2 of all the computed apex sets. As each vertex of $(G-S_1)/(M-S_1)$ occurs in at most two subgraphs, this union has size at most twice the optimum of $(G-S_1)/(M-S_1)$, which is at most $2k$ for relevant instances. For technical reasons we then add all the matching partners of vertices in S_2 to S_2, at most doubling its size. We then invoke the theorem described above to prove that $S := S_1 \cup S_2$ is a $6k$-apex set of G. The procedure is presented in Section 4.4.

3.2 Reducing the treewidth of G using S

In the second phase we use the $6k$-apex set S in G to reduce the treewidth of G.

Guess the intersection of a solution with S. We start by performing a branching step to guess how the sought solution intersects with S. In each branch, we obtain an instance of the following problem.

	Disjoint Vertex Planarization
Input:	A graph G, integers k and ℓ such that $\ell \leq k$, and a $6k$-apex set S of G.
Parameter:	k.
Output:	An ℓ-apex set of G that is disjoint from S, or no if such a set does not exist.

Concretely, the algorithm tries all subsets $A \subseteq S$ for the intersection of the desired apex set with S. For each guess A it produces an instance $(G' := G-A, k' := k, S' := S\setminus A, \ell' := k-|A|)$ of the disjoint problem. Given a solution to the disjoint problem, we can take its union with the corresponding set A to obtain a solution to the original problem. If G has a k-apex set S^*, then we find a solution for the disjoint problem in the branch where $A = S \cap S^*$. This is described in Section 5.1.

Trim the Tutte decomposition of the planar graph. We continue by performing treewidth reduction for an instance (G, k, S, ℓ) of Disjoint Vertex Planarization. It suffices to reduce the treewidth of the planar graph $G - S$, as the remaining $|S|$ vertices increase the treewidth by at most $6k$. Rather than reducing the treewidth of the entire planar graph at once, we use the *Tutte decomposition* to decompose $G - S$ into its triconnected components and work on each component individually. Triconnected planar graphs have a unique embedding, and enjoy some structural properties which will be exploited in later steps of the algorithm. In particular, the next phase of the algorithm relies on the fact that if there is an ℓ-apex set disjoint from S, then the neighbors of the approximate solution S can be covered by $\mathcal{O}(\ell)$ balls of constant radius, which is only valid for triconnected planar graphs. We continue by describing the Tutte decomposition and some simplification steps on it.

One may think of a Tutte decomposition of a connected planar graph H as a tree decomposition $(T, \mathcal{X} = \{X_i \mid i \in V(T)\})$ of H, such that the torso of each bag is triconnected, and each pair of bags has an intersection of size at most two; such intersections correspond to size-2 separators in the graph. The structure of the Tutte decomposition reveals irrelevant parts of the planar graph. If there is a leaf i in the decomposition tree T whose bag X_i does not contain neighbors of the apex set S, then we prove that the triconnected component corresponding to i does not play a significant role for Kuratowski minors in the graph G. Consequently, we may remove the triconnected component i, replacing it by a direct edge if the adhesion to its parent bag has size two. Using a tree traversal algorithm we can exhaustively trim the Tutte decomposition in linear time. This ensures that afterward, each leaf bag in the decomposition contains a neighbor of S. The linear-time trimming step allows us to zoom in on the relevant triconnected components of the graph in linear-time, which might also be useful in other algorithms.

Cover $N_G(S)$ and the adhesions by $\mathcal{O}(k)$ constant-radius balls. Having trimmed the Tutte decomposition of the planar graph, we work on the remaining graph G with its apex set S. The fact that we handle each triconnected component individually causes some technical complications, as the adhesion vertices of the decomposition form portals into other parts of the planar graph, whose connections to the apex set S are hidden when focusing on a single triconnected component. As all Kuratowski minors have to pass through the apex set S, it is exactly these connections to S that are important to the structure of the problem. To uncover the relevant parts of a triconnected component $\text{Torso}(G-S, X_i)$ for a node $i \in V(T)$ of the Tutte decomposition, we consider an embedding in the plane. The important pieces of $\text{Torso}(G-S, X_i)$ are its connections to S, which can be made through neighbors of S in X_i, or through adhesion vertices in X_i that connect to other triconnected components. We prove that, if our guess was correct and there is indeed an ℓ-apex set disjoint from S, then there is a set W_i of $\mathcal{O}(\ell)$ vertices in X_i such that the radius-seven radial balls around W_i contain all the neighbors of S in X_i, together with all the adhesion vertices in X_i. Hence vertices that can give connections to S in $\text{Torso}(G - S, X_i)$, are contained in $\mathcal{O}(\ell)$ constant-radius regions of the planar triconnected component, which will be exploited later.

Divide G into $\ell + 1$ thinning sets that avoid the covering balls. The treewidth reduction inside a triconnected component $\text{Torso}(G - S, X_i)$ is based on the notion of a *thinning set*. Roughly speaking, this is a subset of the outerplanarity layers of an embedding of the triconnected component, which can be partitioned into groups that each contain a constant number of consecutive layers, such that the radial distance between such groups is $\mathcal{O}(\ell)$. The latter property ensures that when we remove the vertices in the thinning set from the triconnected component, the remaining pieces are $\mathcal{O}(\ell)$-outerplanar, and therefore have treewidth $\mathcal{O}(\ell)$ by Proposition 8.

While such a treewidth reduction is our goal, just removing the vertices in a thinning set obviously changes the structure of the instance. We must therefore be more careful. Rather than deleting vertices of a thinning set, we do the opposite: we forbid them to be part of an apex set, to allow a reduction rule to be applied in the next phase that reduces the maximum number of vertex-disjoint paths through the thinning set. This reduction rule will make the graph induced by the thinning set "thinner", thereby reducing the treewidth of $\text{Torso}(G-S, X_i)$.

To safely forbid the vertices of a thinning set to be part of an apex set, without affecting the existence of a solution, we need to find a thinning set such that there is at least one ℓ-apex set that does not intersect it. To be able to find planarizing cycles inside thinning sets, which is a prerequisite to the reduction rule, we shall also require the thinning set to avoid the $\mathcal{O}(\ell)$ constant-radius regions that cover the portals to S. In Section 5.4 we therefore show how to find $\ell + 1$ disjoint thinning sets, none of which intersect the $\mathcal{O}(\ell)$ constant-radius covering balls around W_i. As a solution of size ℓ cannot intersect all $\ell+1$ thinning sets, we branch in $\ell+1$ ways and are guaranteed to have at least one branch whose thinning set avoids an optimal solution, if a solution of size ℓ exists.

Reduce the Flow through Undeletable Parts Given a thinning set for a triconnected component, we will mark all vertices in the layers it contains as undeletable. In a branch where the thinning set avoids some optimal solution, this clearly does not affect the existence of a solution. Making the thinning set undeletable allows two types of irrelevant vertices to be identified. First of all, observe that all subgraphs of the triconnected component that do not contain any of the $\mathcal{O}(\ell)$ balls that cover the adhesions and the neighbors of S, are planar. This causes vertices that are separated from all covering balls by two undeletable disjoint cycles to be irrelevant; such vertices are removed if they are not part of the thinning set. All such vertices can be identified in linear time by a tree traversal procedure on a tree T_i that encodes where the centers W_i of the constant-radius covering balls lie with respect to the blocks of the thinning set.

By discarding the first type of irrelevant vertices, which are not contained in thinning sets, we pave the way for the identification of the irrelevant vertices inside the thinning set. The reduction step ensures that when looking at a block of consecutive outerplanarity layers in the thinning set, there are only $\mathcal{O}(\ell)$ types of connections to be made through this block that are important to the structure of the problem. These correspond to connections between different covering balls around W_i that can be made through the layers in the block; we use here that $|W_i| \in \mathcal{O}(\ell)$. The classification of the second type of irrelevant vertices is based on the following fact: a vertex that is insulated from the non-planar parts of the graph by 2 disjoint cycles is irrelevant to the existence of apex sets that do not intersect those cycles. Hence, using the fact that we mark all vertices of the thinning set as undeletable, and that the thinning set avoids the balls around W_i and therefore induces a planar graph, we can discard vertices from the thinning set that are insulated from the outside by two cycles in the thinning set. Even though a solution is not allowed to delete such an undeletable vertex, we may just omit it from the graph, resulting in an equivalent instance. To find the second type of irrelevant vertices in linear time, we need to identify them all in one pass, rather than one at a time. Toward this end, we find connected sets inside the thinning set, that contain all the relevant connections between the vertices W_i. Taking a constant-radius ball around these connecting sets, we can classify all the vertices of the thinning set that do not belong to this ball, as irrelevant. The fact that their radial distance to the connected set is more than a constant ensures the existence of planarizing cycles, using Proposition 3. The implementation of these steps is described in Section 5.5.

Find a width-$\mathcal{O}(k)$ tree decomposition of the reduced graph. By thinning the graph induced by the thinning set, we ensure that in the remaining part of the triconnected component, each connected component of the graph minus the thinning set, can be separated from the other components of the graph minus the thinning set, by a separator of size $\mathcal{O}(\ell)$. We organize these components linearly, using the structure of the thinning set. As each such component is $\mathcal{O}(\ell)$-outerplanar it is easy to find width-$\mathcal{O}(\ell)$ tree decompositions for them. Afterward we use the mentioned path structure to combine the separators and the outerplanar parts into a tree decomposition of width $\mathcal{O}(\ell)$ for the entire triconnected component. In turn, the tree decompositions for the triconnected components are merged into a tree decomposition of the entire planar graph, by using the structure of the Tutte decomposition tree. These merging steps are described in Section 5.6. Afterward we may add the set S to all bags of the tree decomposition to obtain a decomposition of the entire graph G. Since this increases the treewidth by at most $|S| \leq 6k$, and using the fact that $\ell \leq k$, this results in a tree decomposition of G that has width $\mathcal{O}(k)$. The output of the overall treewidth reduction scheme, consists of a series of low-treewidth instances of the following problem.

	RESTRICTED DISJOINT VERTEX PLANARIZATION
Input:	A graph G, integers k and ℓ such that $\ell \leq k$, a set of undeletable vertices $D \subseteq V(G)$, and an apex set $S \subseteq D$ of G of size at most $6k$.
Parameter:	k.
Output:	An apex set for G of size at most ℓ that is disjoint from D, or NO if such a set does not exist.

3.3 Planarization on Graphs of Bounded Treewidth

The treewidth reduction algorithm outputs $\ell + 1 \leq k + 1$ instances of RESTRICTED DISJOINT VERTEX PLANARIZATION, along with tree decompositions of width $\mathcal{O}(k)$. For ease of presentation, we formulate a dynamic programming algorithm that solves a more general problem, namely the variant of VERTEX PLANARIZATION where each vertex has a positive integral weight value, and we want to find a minimum-weight apex set. By setting the weights of the forbidden vertices to $\ell + 1$, and of the other vertices to 1, this dynamic program allows us to solve the RESTRICTED DISJOINT VERTEX PLANARIZATION problem. We give a $2^{\mathcal{O}(w \log w)} \cdot n$-time algorithm for the following problem.

	WEIGHTED VERTEX PLANARIZATION ON BOUNDED TREEWIDTH GRAPHS
Input:	A graph G with a tree decomposition of width w, along with a non-negative integer cost c_v for each vertex v of G.
Parameter:	w.
Output:	A minimum-cost apex set in G.

The states of the dynamic programming algorithm are based on possible embeddings of the graph. Toward that end, we associate a characteristic function with each bag of the tree decomposition. The arguments to the characteristic function consist of an intersection X of the solution with the vertices B in the bag, along with a plane multigraph P on the remaining vertices $B \setminus X$ that is a supergraph of $G[B \setminus X]$, and a subset of allowed faces F in P. The characteristic function returns the minimum cost of an apex set whose intersection with B is exactly X, and for which the remaining graph admits an embedding that coincides with P on $B \setminus X$, and which embeds the other vertices in the subtree rooted at bag B in the faces designated by F. The number of plane graphs on w vertices is $2^{\mathcal{O}(w \log w)}$, which dominates the size of the tables in the dynamic program. Some reduction steps are used to compensate for the fact that we have to work with plane multigraphs rather than simple graphs. This results in an algorithm that solves WEIGHTED VERTEX PLANARIZATION ON BOUNDED TREEWIDTH GRAPHS in $2^{\mathcal{O}(w \log w)} \cdot n$ time. The details are given in Section 6.

3.4 Overall running time

If we use the described algorithm for bounded treewidth graphs on the instances that result from the treewidth reduction steps, we obtain an algorithm that solves k-VERTEX PLANARIZATION in $2^{\mathcal{O}(k \log k)} \cdot n$ time. Let us briefly justify this running time. Observe that there is only one point in the overall algorithm where we recurse. The target instance of the recursion has at most $(1 - \frac{1}{30 \cdot 2^k (k+13)})$ times as many vertices as the input. Then we branch in 2^{6k} ways when guessing the intersection of the approximate apex set with an optimal solution, and later we branch in $\ell + 1 \leq k + 1$ ways when trying to find a thinning set that avoids an optimal solution. Apart from the branching and the recursion, each step in the treewidth reduction scheme runs in $k^{\mathcal{O}(1)} n$ time. Ignoring the recursion, in total we solve at most $2^{6k}(k+1)$ instances of the planarization problem on graphs of treewidth $\mathcal{O}(k)$, which takes $2^{6k}(k+1)2^{\mathcal{O}(k \log k)} \cdot n \in 2^{\mathcal{O}(k \log k)} \cdot n$ time. The running time of the entire algorithm is therefore governed by the recursion $T(n,k) \leq T((1 - \frac{1}{30 \cdot 2^k (k+13)})n, k) + 2^{\mathcal{O}(k \log k)} \cdot n$, which solves to $2^{\mathcal{O}(k \log k)} \cdot n$ as the contribution of the recursive calls forms a geometric series.

4 Finding an Approximate Apex Set

In this section we develop the tools that are used in the linear-time algorithm that finds a $6k$-apex set in our input graph G, or decides that no such set exists. The presentation follows the subparts of the algorithm as described in Section 3.

4.1 Shrinking Twin Classes

Lemma 3. *If G has a vertex v such that v has $k+3$ false twins in G, then a set $S \subseteq V(G \setminus \{v\})$ of size at most k is an apex set of G if and only if it is an apex set of $G - v$.*

Proof. As $G - v$ is a subgraph of G, the "only if" direction is trivial. For the reverse, let S be an apex set of size at most k in $G - v$. There are at least three false twins of v not contained in S. If $|N_G(v) \setminus S| \geq 3$ then these three false twins together with their neighborhood form a nonplanar $K_{3,3}$ subgraph of $(G - v) - S$, so $|N_G(v) \setminus S| \leq 2$ since $(G - v) - S$ is planar. So any false twin of v is a vertex of degree at most two in $G - S$, which shows that any face of $G - S$ containing such a false twin, also contains all vertices of $N_G(v) \setminus S$ on its boundary. We may draw v inside this face to get a planar drawing of $G - S$, showing that S is an apex set in G. ☐

4.2 Large Matchings in Twin-Reduced Graphs

Lemma 4. *There is an algorithm that given a graph G and integer k, runs in $\mathcal{O}(nk)$ time and either solves k-*Vertex Planarization*, or outputs an induced subgraph G' of G with $\mathcal{O}(nk)$ edges such that (i) any apex set S in G' of size at most k is also an apex set of G, and (ii) if G' is k-apex then any maximal matching in G' has size at least $\frac{|V(G')|}{30 \cdot 2^k (k+13)}$.*

Proof. Given as input a graph G and integer k our algorithm proceeds as follows. First, if the number m of edges exceeds $n(3+k)$, then G cannot have an apex set of size at most k since removing any k vertices leaves a graph on more than $3n$ edges and less than n vertices; by Euler's formula such graphs are nonplanar. If $m > n(3+k)$ we therefore output NO.

What the algorithm aims to do in the remaining cases, is to apply the reduction rule of Lemma 3 exhaustively on G. While it is possible to use a linear-time algorithm for modular decomposition to identify all twin classes in linear time, such algorithms are fairly involved (e.g. [35]). To keep our planarization algorithm simple we will only apply the rule exhaustively on vertices of degree at most $12 + k$. This can be done as follows; we pick all of the vertices in G of degree at most $12 + k$ and sort them by their neighborhoods. By identifying each vertex with a number in the range $[1 \ldots n]$ and ignoring vertices whose adjacency list has more than $k + 12$ members, this can be done in time $\mathcal{O}(nk)$: use $12 + k$ stable bucket sorts, first on the first neighbor, then on the second, etc. After sorting, for each twin class of degree at most $12 + k$ its members appear consecutively on the sorted list. We iterate over the list and compare the adjacency lists of successive vertices if their degree is at most $12 + k$, to find the twin classes of small degree. Whenever such a twin class contains more than $k + 13$ vertices, we keep $k + 13$ of the vertices in the class and delete the remainder. It is easy to see this can be done in $\mathcal{O}(n + m) = \mathcal{O}(nk)$ time, and by Lemma 3 k-apex sets for the reduced graph are also apex sets in the original. This would even be true if we reduced the size of a twin class to $k + 3$, but the reduction to size $k + 13$ ensures that the process does not create any new vertices of degree at most twelve. Hence after a single pass of twin reduction we are guaranteed that in the resulting graph G', any twin class of vertices of degree at most $12+k$ has at most $k+13$ members.

Graph G' is the output of the procedure. It remains to prove that if G' is k-apex, then its maximal matchings contain at least $\frac{|V(G')|}{30 \cdot 2^k (k+13)}$ edges. Assume therefore that S' is an apex set in G' of size at most k, and that $M' \subseteq E(G')$ is a maximal matching. If $M' = \emptyset$ then G' is edgeless, and therefore planar, allowing us to solve the k-Vertex Planarization problem trivially. In the remainder we therefore have $M' \neq \emptyset$. Accordingly, the endpoints X' of M' form a nonempty vertex cover of G'. Let $G'' := G' - S'$ be planar, and consider the vertex cover $X'' := X' \setminus S'$ of G''. The complement $I'' := V(G'') \setminus X''$ is independent in G''. Let $I''_r \subseteq I''$ consist of those vertices of I'' having degree at most twelve in G''.

Split the vertices in I''_r into equivalence classes according to their neighborhood in G''. As X'' is a vertex cover of the planar graph G'', by Lemma 2 the number of equivalence classes is at most $6|X''|+1$. Now refine each class based on the adjacency to S' in G', splitting each class into at most $2^{|S'|} \leq 2^k$ subclasses. The refined partition of I''_r into at most $(6|X''|+1)2^k$ classes is such that vertices in the same class have the same open neighborhood in G'. Since all vertices in I''_r have degree at most twelve in G'', their degree in G' is at most $12+k$ and therefore the reduction rule has been applied to them. This ensures that no twin class in I''_r has more than $k+13$ members, implying $|I''_r| \leq 6(|X''|+1)2^k(k+13)$.

We now claim that $|I''| \leq 2|I''_r| + |X''|$. Assume for a contradiction that this is not the case; then $|I''_r| \leq |I''|/2 - |X''|/2$. As all vertices in $I'' \setminus I''_r$ have degree more than twelve in G'', the total number of edges in G'' would exceed $12(|I''| - |I''_r|) \geq 12(|I''| - |I''|/2 + |X''|/2) \geq 12(|I''|/2 + |X''|/2) = 6|V(G'')|$. Observe that we count each edge at most once, as I'' is an independent set. As a planar graph on $|V(G'')|$ vertices has less than $3|V(G'')|$ edges by Euler's formula, the fact that $|E(G'')| \geq 6|V(G'')|$ contradicts the planarity of G''. Hence we must have $|I''| \leq 2|I''_r| + |X''|$, which we use to conclude the proof.

Observe that $|V(G')| = |I''| + |X''| + |S'|$. Plugging in the obtained bounds yields $|V(G')| \leq 2 \cdot 6(|X''|+1)2^k(k+13) + |X''| + |X''| + |S'|$. As X' is not empty, this easily implies $|V(G')| \leq 15|X'| \cdot 2^k(k+13)$. As X' contains the endpoints of the maximal matching M', we have $|X'| \leq 2|M'|$ giving $|V(G')| \leq 2 \cdot 15|M'| \cdot 2^k(k+13)$. By trivial formula manipulation this implies the lemma. ☐

4.3 Locally Planar Means Globally Planar

Let G be a graph which has a matching M whose contraction makes G planar. In this section we present a purely graph-theoretic result which shows that if there is an embedding of G/M such that all balls of small radius in the embedding have a pre-image under M that induces a planar subgraph of G, then all of G is in fact planar. Before presenting the theorem and its proof, we give a lemma that will be needed later on.

Lemma 5. *Let G be a graph with a vertex set $P \subseteq V(G)$ containing an edge $xy \in G[P]$ such that $G[P]$ is planar, G/xy is planar, and for every $v \notin P$ there is a cycle C in $G[P] - \{x,y\}$ whose vertex set separates the pair xy from v in G. Then G is planar.*

Proof. Proof by induction on $|V(G) \setminus P|$. If $V(G) \setminus P = \emptyset$ then the claim clearly follows from the assumption that $G[P]$ is planar. Now consider some vertex $v \in V(G) \setminus P$. By assumption there is a cycle C in $G[P] - \{x,y\}$ that separates the pair xy from v. We will apply the recursive characterization of Lemma 1 to show G is planar.

Consider the C-bridges in G. For all C-bridges B we have to prove that $C \cup B$ is planar. For the bridges B that do not contain v we may apply induction as follows. Observe that the graph $C \cup B \cup G[P]$ satisfies all requirements for the induction hypothesis, using the same set P

and the same edge xy. As that graph has a smaller vertex set outside P (the outside does not contain v), by induction we find that $C \cup B \cup G[P]$ is planar and therefore the subgraph $C \cup B$ is planar as well. Now consider the bridge B_v containing v. As the vertex set of C separates xy from v, the graph $C \cup B_v$ does not contain xy, and is therefore a minor of G/xy; hence it is planar. So for all C-bridges B we find that $C \cup B$ is planar.

By Lemma 1 it remains to prove that the overlap graph $O(G, C)$ is bipartite. But as the endpoints x and y of the edge are explicitly not contained in C, by Proposition 4 we find that $O(G, C) = O(G/xy, C)$. Since G/xy is planar the graph $O(G/xy, C)$ is bipartite, and therefore $O(G, C)$ is bipartite. By Lemma 1 this proves that G is planar. $\square$

Now let us give the main result of this section with its proof.

Theorem 1. *Let G be a graph with a matching M such that G/M is planar. If G/M has an embedding such that for every vertex $v \in V(G/M)$, the 3-ball $\mathbf{R}^3_{G/M}(v)$ around v has a pre-image that induces a planar subgraph of G, i.e., $G[\mathbf{R}^3_{G/M}(v)/^{-1}M]$ is planar, then G is planar.*

Proof. We give a proof by minimum counterexample. Assume for a contradiction that the claim is false, and out of all counterexamples select one that minimizes the number of edges in M. So assume that G/M is embedded in the plane such that the pre-image of each 3-ball around a vertex induces a planar subgraph of G, but that G is nonplanar, and it is not possible to achieve this with a smaller matching M.

Observe first of all that M is not empty, otherwise the planarity of G/M already implies that G is planar. Throughout the rest of the proof we will use X to denote the vertices in G/M which are the result of contracting an edge in M. We establish some structural claims about the counterexample.

Claim 1. For all edges $e \in M$ the graph G/M is planar.

Proof. Assume there is an edge e in the matching whose contraction does not make G planar. Now define $G' := G/e$ and let $M' := M \setminus \{e\}$. By assumption G' is nonplanar. Observe that $G'/M' = (G/e)/(M \setminus \{e\}) = G/M$ and is therefore planar; we pick the same embedding for G'/M' as for G/M. As the plane graphs G'/M' and G/M are identical, so are the 3-balls around each vertex. Now consider a vertex $v \in G'/M'$ and compare $G[\mathbf{R}^3_{G'/M'}(v)/^{-1}Y]$ to $G[\mathbf{R}^3_{G/M}(v)/^{-1}Y]$. Either these graphs are the same, or the former is obtained from the latter by the contraction of e. Hence the former is a minor of the latter for each vertex v, and as the preconditions guarantee that the latter is planar for every choice of v, so is the former. Hence the pair G' and M' forms a counterexample with a smaller matching than M, contradicting the choice of M. $\diamondsuit$

Claim 2. There is no cutvertex v in G such that there are two components of $G - v$ that both contain an edge of M.

Proof. Assume for a contradiction that there are distinct edges e, e' in M and a cutvertex v such that e and e' are contained in different connected components C_e and $C_{e'}$ of $G - v$. Observe that this implies that v is not an endpoint of e or e'. The separation after removal of v shows that the edges e and e' are in different biconnected components of G. By Proposition 5 G is planar if and only if all its biconnected components are planar. Consider some biconnected component G_i of G. As G_i avoids at least one of e and e', it is a minor of the graph G/e or of G/e'. By Claim 1 the latter graphs are both planar, which shows that its minor G_i is also planar. Hence all G_i are planar which proves that G is planar, thereby contradicting our choice of counterexample. $\diamondsuit$

Claim 3. There is no cycle in G such that there are two components of $G - V(C)$ that both contain an edge of M.

Proof. Suppose $G - V(C)$ has components C_e and $C_{e'}$ that contain the edges $e, e' \in M$ respectively. We will derive a contradiction by using Proposition 1 to prove that G is planar.

By Claim 1 the graphs G/e and G/e' are planar, which implies that the overlap graphs $O(G/e, C)$ and $O(G/e', C)$ are bipartite. As e and e' are contained in connected components of $G - V(C)$, their endpoints do not lie on C. By Proposition 4 this shows that $O(G/e, C) = O(G, C) = O(G/e', C)$ which proves that G's overlap graph with respect to C is bipartite, and thereby establishes the second criterion for Lemma 1. Now consider a C-bridge B in G. As e and e' are contained in different connected components of $G - V(C)$, the bridge avoids at least one edge of e and e'. Hence the graph $B \cup C$ is a minor of G/e or G/e', which are both planar. As $B \cup C$ is the minor of a planar graph, it is planar. As this holds for all bridges, the second criterion of Lemma 1 also holds which proves that G is planar; a contradiction to our choice of counterexample. $\diamondsuit$

For the next claim, recall that X contains the vertices in G/M that are the result of contracting edges of M.

Claim 4. There is no vertex $z \in X$ such that $X \subseteq \mathbf{R}^2_{G/M}(z)$.

Proof. Suppose for a contradiction that there is a vertex $z \in X$ such that the 2-ball centered at z contains the result of all the contractions. Define the abbreviation $B := \mathbf{R}^3_{G/M}(z)$ as the 3-ball around z, and let $P := B/^{-1}M$ be the pre-image of this ball under M. Let xy be the edge in M whose contraction yields z. Then G/xy is planar by Claim 1, and $G[P]$ is planar by the assumption on the counterexample. We will apply Lemma 5 to prove G is planar.

Consider a vertex $v \notin P$, which implies that v is not the endpoint of an edge in M and therefore also exists in G/M. The radial distance from z to v in G/M is at least four, otherwise v would be contained in the 3-ball around z and hence in the pre-image P. By Proposition 3 there is a cycle C in G/M consisting of vertices at radial distance three from z, whose vertex set separates z from v in G/M. As the vertices resulting from edge contractions are all contained within the 2-ball around z, the vertices on C are not involved in edge contractions. Hence C is also a cycle in G, and separates v from the pair xy in G.

The preceding argument shows that for every $v \notin P$ there is a cycle C in $G - \{x, y\}$ whose vertex set separates the pair xy from v in G. Hence we may apply Lemma 5 on G with the vertex set P and pair xy to establish that G is planar; a contradiction to our choice of counterexample. $\diamondsuit$

The next structural claim is the last one we shall need. Together with the previous claim it will show that no counterexamples can exist.

Claim 5. There are no distinct vertices u and v in X such that $d_{G/M}(u, v) \geq 3$.

Proof. Assume for a contradiction that $u \in X$ is the result of contracting edge $pq \in M$, and $v \in X$ is the result of contracting $xy \in M$, and that $d_{G/M}(u, v) \geq 3$. Let G^* be obtained from G by contracting pq into u and contracting xy into v. As G^* is a minor of G/pq, which is planar by Claim 1, graph G^* is planar. Fix an arbitrary embedding of G^*. As the first step we shall prove that u and v must be embedded on the same face.

If u and v are not embedded on the same face, then $d_{G^*}(u, v) \geq 2$ which shows by Proposition 3 that there is a cycle C in G^* whose vertices all have radial distance exactly one from u in G^*, such that $V(C)$ is a uv-separator in G^*. It is easy to see that C is also a cycle in G, and the separation property ensures that the edges pq and xy which are the pre-images of u and v, are contained in different connected components of $G - V(C)$. Now observe that this contradicts

Claim 3. Hence u and v must lie on the same face of G^*. The classic balloon argument shows that given an embedding, any face can be blown up to become the outer face while preserving a correct embedding. Hence we may assume in the remainder that u and v are embedded on the outer face of G^*.

We will show how to construct a cycle C in G such that $V(C)$ separates the edge pq from the edge xy in G, thereby contradicting Claim 3. We build the cycle by combining parts of uv paths in G^* with the pre-images of cyclic uv separators in G/M. Let us first show how to construct the uv-paths.

If there is a cutvertex that separates u and v in G^*, then removing this vertex from G results in two components each containing an edge of M, thereby contradicting Claim 2. Hence u and v lie in the same biconnected component of G^*. Now let G^*_{uv} be the plane subgraph of G^* consisting of the biconnected component holding u and v. Form two internally vertex-disjoint uv paths in G^*_{uv}: let P_1 trace the boundary of the outer face clockwise, and let P_2 trace the boundary counterclockwise. As the biconnected component contains no cutvertices, and u and v both lie on the outer face, it is easy to see that this results in simple internally vertex-disjoint uv-paths in G^*_{uv}.

To construct our cycle C, we shall also need two cyclic uv-separators in G/M. As the radial distance from u to v in G/M is at least three by assumption, Proposition 3 shows that G/M contains a cycle C_1^M on vertices at radial distance exactly one from u in G/M whose vertex set forms a uv-separator in G/M, and similarly there is such a cycle C_2^M on vertices at radial distance two from u. Hence the vertex sets of the cycles C_1^M and C_2^M are disjoint uv-separators in G/M. Now consider the pre-images C_1 and C_2 under M' of C_1^M, C_2^M respectively, i.e., $C_1 := C_1^M/^{-1}M$ and $C_2 := C_2^M/^{-1}M$. As the inverse operation of edge contraction preserves connectivity, the graph $G[C_1]$ is connected and so is $G[C_2]$. As the pre-image of a separator is a separator in the pre-image of the graph, the vertex sets C_1 and C_2 separate pq from xy in G. As C_1 and C_2 are disjoint from pq and xy, the vertex sets also exist in G^*. As pq is contracted into u when creating G^*, and xy is contracted into v, it follows that C_1 and C_2 are disjoint, connected uv-separators in G^*.

Let us now show how the uv-paths P_1, P_2 in G^*_{uv} can be combined with C_1, C_2 in G^* to give the desired separating cycle in G. As P_1 and P_2 are also uv-paths in G^*, they intersect the uv-separators C_1 and C_2. As the separators are connected and vertex-disjoint, they must be nested. Relabel the separators such that any uv-path meets C_1 before it meets C_2. Let z_{P_1,C_1} be the last vertex from C_1 visited when traversing P_1 from u to v, and let z_{P_1,C_2} be the first vertex of C_2 that is visited by this path. Similarly, let z_{P_2,C_1} be the last vertex of C_1 that P_2 visits, and let z_{P_2,C_2} be the first vertex on C_2 met by P_2. As $G^*[C_1]$ is connected, there is a simple path R_{C_1} between z_{P_1,C_1} and z_{P_2,C_1} in $G^*[C_1]$. Similarly there is a simple $z_{P_1,C_2}z_{P_2,C_2}$-path R_{C_2} in $G^*[C_2]$. Path P_1 contains a simple subpath from z_{P_1,C_1} to z_{P_1,C_2}, while path P_2 contains a simple subpath from z_{P_2,C_1} to z_{P_2,C_2}. As P_1 and P_2 are internally vertex-disjoint, these subpaths are disjoint. Therefore the combination of these subpaths with R_{C_1} and R_{C_2} yield a simple cycle C in G^* whose vertices form a superset of R_{C_1}. We will finalize the proof by showing that the vertex set of R_{C_1} separates edge pq from edge xy in G, thereby establishing that removal of $V(C) \supseteq V(R_{C_1})$ from G disconnects two edges of M, contradicting Claim 3.

So it remains to prove that the vertex set of R_{C_1} separates pq from xy in G, which is obviously equivalent to showing that R_{C_1} is a uv-separator in G^*. Consider the structure of R_{C_1}. Since R_{C_1} is a simple path connecting two vertices on distinct uv paths, R_{C_1} is contained in the same biconnected component as u and v, hence R_{C_1} exists in G^*_{uv}. As all simple uv paths are contained in the same biconnected component, it suffices to show that R_{C_1} is a uv-separator in G^*_{uv}. Now consider the boundary walk of the outer face of G^*_{uv}. As G^*_{uv} is biconnected, and the endpoints of R_{C_1} are intersection points of P_1 and P_2 with the uv-separator C_1, the vertices $u, z_{P_1,C_1}, v, z_{P_2,C_1}$ must occur on the boundary walk in this cyclic order. But this shows

Algorithm 1 VERTEX PLANARIZATION OF MATCHING-CONTRACTIBLE GRAPHS

Input: A graph G with a matching M such that G/M is planar, and an integer k.
Output: An apex set of size at most $4k$ for G, or NO if G is not k-apex.

  Compute a plane embedding of G/M
  Compute the outerplanarity layers $L_0, \ldots, L_{\ell-1}$ of the embedding
  Let S be a vertex subset, initially empty
  for $j = 0$ to $\lfloor \ell/(8k+8) \rfloor$ **do**
    Let G_j^M be the plane subgraph of G/M induced by $\bigcup_{i=j\cdot(8k+8)}^{(j+2)(8k+8)-1} L_i$
    Build a tree decomposition $(T, \mathcal{X})$ of G_j^M of width $\mathcal{O}(k)$ using Proposition 8
    Let G_j be the pre-image of G_j^M before contracting the matching
    Lift the tree decomposition $(T, \mathcal{X})$ to G_j using Proposition 9
    Run the algorithm of Theorem 3 on $(G_j, (T, \mathcal{X}), k)$ and add the result to S
  if $|S| > 2k$ **then**
    Output NO
  else
    Output $S' := \{v \in V(G) \mid v \in S \text{ or there is an edge } uv \in M \text{ and } u \in S\}$

that any $z_{P_1,C_1} z_{P_2,C_1}$ path splits the biconnected component in two, one containing u and the other containing v. As R_{C_1} is a path between these two vertices, it forms a uv-separator in G^*_{uv} and therefore in G^*, which finally proves that the vertex set of R_{C_1} separates pq from xy in G. This concludes the proof of the claim. $\diamondsuit$

Armed with these structural claims we can finally complete the proof of Theorem 1. Suppose some minimum counter example exists. By Claim 4 there is no ball of radius two in G/M that contains the results of all the edge contractions. So if we choose some arbitrary $z \in X$ that is the result of a contraction, then there is some vertex $v \in X \setminus \mathbf{R}^2_{G/M}(v)$, which implies that the radial distance from z to v in G/M is at least three. But this contradicts Claim 5, thereby showing that no counterexample can exist: the theorem follows. $\square$

4.4 Vertex Planarization in Matching-Contractible Graphs

In this section we focus on the planarization problem in matching-contractible graphs. For the ease of understanding, we formulate the main result of the section in terms of an independent algorithm for the following problem.

	VERTEX PLANARIZATION OF MATCHING-CONTRACTIBLE GRAPHS
Input:	A matching-contractible graph G with a matching M such that G/M is planar, and an integer k.
Parameter:	k.
Output:	An apex set for G of size at most $4k$. If G is not k-apex, the output may instead be NO.

Our approach is summarized by the pseudo-code of Algorithm 1. Observe that it uses Theorem 3, the dynamic program for graphs of bounded treewidth, as a subroutine.

Theorem 2. *Algorithm 1 solves* VERTEX PLANARIZATION OF MATCHING-CONTRACTIBLE GRAPHS *in* $2^{\mathcal{O}(k \log k)} \cdot n$ *time.*

Proof. We first introduce some terminology to discuss the algorithm. For an input (G, M, k) to VERTEX PLANARIZATION OF MATCHING-CONTRACTIBLE GRAPHS we define bounded-diameter subgraphs of G/M called *slices* that the approximation algorithm works on. Consider the embedding of G/M that is chosen by the algorithm. Let $L_0, \dots, L_{\ell-1}$ be the outerplanarity layers of G/M. Then for $j \geq 0$ the j-th slice of G/M is the plane graph $G/M[\bigcup_{i=j(8k+8)}^{(j+2)(8k+8)-1} L_i]$. As each slice contains $2(8k+8)$ outerplanarity layers of G/M, each slice is $\mathcal{O}(k)$-outerplanar and therefore has treewidth $\mathcal{O}(k)$ by Proposition 8. It also follows from this definition that each vertex of G/M is contained in at most two slices. Roughly speaking the algorithm solves the planarization problem optimally in the pre-image of each slice before the contraction of M, and afterward takes the union of the solutions found in each slice. We will prove that if G is k-apex, then the algorithm outputs an apex set S' of size at most $4k$ in $2^{\mathcal{O}(k \log k)} \cdot n$ time.

As this correctness proof is fairly involved, let us first establish the running time of the procedure on an input (G, M, k) where G has n vertices. An embedding of G/M can be computed in $\mathcal{O}(n)$ time using, for example, the algorithm by Hopcroft and Tarjan [19]. Given the embedding, it is not difficult to compute the outerplanarity layers in linear time by using a BFS in the radial graph. Each slice G_j^M can then be extracted in $|V(G_j^M)|$ time, and a tree decomposition can be found in $\mathcal{O}(k|V(G_j^M)|)$ time by Proposition 8. Lifting the tree decomposition to G_j can be done in the same time bound by Proposition 9, and observe that the pre-image has at most twice as many vertices as G_j^M. The resulting tree decomposition of G_j has width $\mathcal{O}(k)$ so the dynamic programming algorithm of Theorem 3 takes time $2^{\mathcal{O}(k \log k)} \cdot |V(G_j)| = 2^{\mathcal{O}(k \log k)} \cdot |V(G_j^M)|$. As each vertex is contained in at most two slices, the total sum of the slice sizes is bounded by $2n$. It follows that the total running time is $2^{\mathcal{O}(k \log k)} \cdot n$.

Let us now consider the correctness of the algorithm. Suppose that G has an apex set S^* of size at most k. For each considered pre-image of a slice, which is a subgraph G_j of G, the set $S^* \cap V(G_j)$ is an apex set for G_j. Hence the smallest apex set in G_j has size at most $|S^* \cap V(G_j)|$. Since the dynamic programming algorithm finds a minimum apex set, we find that $|S| \leq \sum_j \textsc{opt}(G_j) \leq \sum_j |S^* \cap V(G_j)| \leq 2|S^*|$, where the last step comes from the observation that each vertex of G is contained in at most two slices. Hence if S^* is an apex set of size k, then $|S| \leq 2k$ which proves that the algorithm is correct when it outputs NO because S is too large. Observe that the set S' has size at most $2|S| \leq 4k$: for every vertex in S we add at most one new vertex to S' (its matching neighbor). Hence to establish the overall correctness, it suffices to prove that if the algorithm does not output NO, then S' is an apex set. We shall establish this by relating the outcome of the algorithm to the purely graph-theoretic claims of Theorem 1.

Assume the algorithm outputs a set S', and let M' be obtained from M by removing the edges that have both endpoints in S'. As no edge in M has only one endpoint in S', graph $(G-S')/M'$ is an induced subgraph of G/M and therefore planar. Consider the embedding of G/M used by the algorithm, and restrict it to $(G-S')/M'$.

Claim 6. For every vertex v in $(G-S')/M'$ its 3-ball $\mathbf{R}^3_{(G-S')/M'}(v)$ has a pre-image under M' that induces a planar subgraph of $G - S'$, i.e., $(G - S')[\mathbf{R}^3_{(G-S')/M'}(v)/^{-1}M']$ is planar.

Proof. Observe that the graph $(G - S')/M'$ can be obtained from G/M by removing the vertices in S' (or their images after contraction). Since we embedded $(G - S')/M'$ based on the embedding of G/M, it follows from Proposition 7 that every pair of vertices u and v in $(G - S')/M'$ satisfies $d_{G/M}(u,v) \leq d_{(G-S')/M'}(u,v) + |S'|$. Hence if we fix a vertex v in $(G-S')/M'$, then the radial distance in G/M between v and a vertex $u \in \mathbf{R}^3_{(G-S')/M'}(v)$ is at most $3 + |S'| = 3 + 4k$. By Observation 2 this shows that the outerplanarity indices of u and v in graph G/M differ by at most $3 + 4k$, and hence if the outerplanarity index of v in G/M is i then $\mathbf{R}^3_{(G-S')/M'}(v)$ is contained in $\bigcup_{j=i-(3+4k)}^{i+(3+4k)} L_j$, where $L_0, \dots$ are the outerplanarity layers of G/M. Hence $\mathbf{R}^3_{(G-S')/M'}(v)$ is contained in a series of $8k+7$ consecutive outerplanarity layers of G/M. We divided G/M into slices; each slice spans $2(8k+8)$ outerplanarity layers. As a slice starts at every layer whose index is a multiple of $8k + k$, it follows that there is some slice in G/M, say G^M_j, that contains $\mathbf{R}^3_{(G-S')/M'}(v)$. Therefore the pre-image of $\mathbf{R}^3_{(G-S')/M'}(v)$ under M', is contained in the pre-image G_j of G^M_j under M. As an apex set for G_j was added to S, it follows that $G_j - S$ is planar. As the pre-image of $\mathbf{R}^3_{(G-S')/M'}(v)$ does not contain S', and S' is a superset of S, we establish that the pre-image of $\mathbf{R}^3_{(G-S')/M'}(v)$ is a subgraph of $G_j - S$ and is therefore planar. This proves the claim. $\diamondsuit$

To conclude the proof of Theorem 2, observe that the previous claim shows that we may apply Theorem 1 to the matching-contractible graph $G - S'$ with the matching M' to conclude that $G - S'$ is planar: hence S' is an apex set for G of size at most $4k$. $\square$

5 Reducing Treewidth Using an Approximate Solution

In this section we develop the ingredients of the treewidth reduction scheme. We refer the reader to the outline in Section 3 for an overview of how these pieces contribute to the overall algorithm.

5.1 Reduction to the Disjoint Planarization Problem

The first step in the treewidth reduction process is the reduction to the following disjoint version of the problem.

	Disjoint Vertex Planarization
Input:	A graph G, an integer k, and a $6k$-apex set S of G.
Parameter:	k.
Output:	A k-apex set of G that is disjoint from S, or NO if such a set does not exist.

Observation 3. Let G be a graph and S be an apex set in G. Then G has a k-apex set if and only if there is a subset $S' \subseteq S$ such that graph $G - S'$ has an apex set of size at most $k - |S'|$ that is disjoint from $S \setminus S'$. Conversely, the union of S' with an apex set in $G - S'$ is an apex set in G.

5.2 Trimming the Tutte Decomposition

Here we describe how the Tutte decomposition is trimmed. The graph-theoretical underpinnings of the trimming step are given by Lemma 6. Afterward we show how to implement this algorithmically, in Lemma 7.

Lemma 6. *Let G be a graph with apex set S and let $(F, \mathcal{X} = \{X_i \mid i \in V(F)\})$ be a Tutte decomposition of $G - S$. Suppose that F has a leaf v^* such that $N_G(S) \cap X_{v^*} = \emptyset$. Obtain a graph G' from G as follows. Remove the vertices in X_{v^*} that are not adhesion points. If X_{v^*} contains two adhesion points that are not adjacent in G, then add the edge e between them. The following holds:*

1. *$(F - v^*, \mathcal{X} \setminus \{X^*_v\})$ is a Tutte decomposition of $G' - S$.*
2. *G' is a minor of G.*
3. *Any apex set in G' is also an apex set in G.*

Proof. Assume the conditions in the lemma statement hold. Observe that if F consists of a single leaf vertex v^*, then $F - v^*$ is an empty graph, and accordingly $G' = G[S]$. It is easy to see that the three statements are true in this case, as the assumption that $N_G(S) \cap X_{v^*} = \emptyset$ ensures that $G[X_{v^*}]$ is simply a planar connected component of G that does not interact with S in any way. In the remainder, we may therefore assume that F has at least two nodes. Before proving the three statements, we establish a claim.

Claim 7. For all $u \in V(F - v^*)$, the torsos $\mathrm{Torso}(G - S, X_u)$ and $\mathrm{Torso}(G' - S, X_u)$ are identical.

Proof. We prove that $\mathrm{Torso}(G - S, X_u)$ is a subgraph of $\mathrm{Torso}(G' - S, X_u)$, and vice versa.

Consider an edge pq in $\mathrm{Torso}(G - S, X_u)$. If $pq \in E(G - S)$ then, as $p, q \in V(G' - S)$, the edge is also present in $\mathrm{Torso}(G' - S, X_u)$. If there is a path in $G - S$ whose endpoints are p and q, and whose interior vertices avoid X_u, then such a path also exists in $G' - S$: if the path avoids $X_v \setminus Y_v$ then the same path exists in $G' - S$, and if it uses a vertex of $X_v \setminus Y_v$ then we may shortcut the path by the edge that was added between the two adhesion points in X_v. Hence $\mathrm{Torso}(G' - S, X_u)$ contains all edges of $\mathrm{Torso}(G - S, X_u)$.

In the other direction, consider an edge $pq \in \mathrm{Torso}(G' - S, X_u)$. If $pq \notin E(G' - S)$, then there is a path P' in $G' - S$ from p to q whose internal vertices avoid X_u. If P' avoids the edge e that is possibly added to G', then it is also a path in $G - S$. If it uses e, then we can replace e by a path between the adhesion points that are the endpoints of e, and whose interior vertices avoid X_u, whose existence is guaranteed Proposition 12. This results in a pq-path in $G - S$ whose interior avoids X_u; hence $\mathrm{Torso}(G - S, X_u)$ also has the edge pq. If $pq \in E(G' - S) \cap E(G - S)$ then the edge is trivially in $\mathrm{Torso}(G - S, X_u)$. And finally, if $pq \in E(G' - S) \setminus E(G - S)$ then it is the edge e added to G' in the construction of the lemma. By Proposition 12 there is a pq-path in $G - S$ whose interior vertices avoid X_u. Hence pq is an edge in $\mathrm{Torso}(G - S, X)$. $\diamondsuit$

With these claims we prove the three statements in the lemma.

(1) It is straight-forward to verify that each tree in $(F - v^*, \mathcal{X} \setminus \{X_{v^*}\}$ gives a tree decomposition for a unique connected component of $G' - S$. To see that it is actually a Tutte decomposition, consider the tree T in F that contains v^*. As the other trees in F are not affected, nor are their associated connected components of $G - S$, they continue to satisfy the conditions of a Tutte decomposition. We briefly show why the remainder of tree T in $F - v^*$ also satisfies the conditions of a Tutte decomposition. As deleting a node of T with its bag does not increase the size of intersections between bags, we have $|X_i \cap X_j| \leq 2$ for all distinct $i, j \in V(T - v^*)$ (b). By Claim 7, for each $u \in V(T - v^*)$ we have $\mathrm{Torso}(G, X_u) = \mathrm{Torso}(G', X_u)$. As the former are triconnected, so are the latter, establishing (c). Finally, condition (d) holds for the trimmed Tutte decomposition since we have only removed a bag from the decomposition, without altering the contents of the other bags.

(2) Graph G' is obtained from G by removing the non-adhesion vertices in X_v, and adding an edge between non-adjacent adhesion vertices in X_v if those exist. If the latter does not occur, then G' is an induced subgraph of G. If there are two adhesion points x, y in X_v that are non-adjacent, then these are the only adhesion points in X_v by condition (b) of Tutte decompositions. Let u^* be the parent of v^* in the tree $T \in F$ that contains v^*. Applying Proposition 12, where node v^* plays the role of u, and u^* plays the role of v, shows that there is a xy-path P_{xy} in $G - S$ on a vertex subset of $\bigcup_{i \in V(T_{v^*})}$. Here T_{v^*} is the connected component of $T - u^*$ that contains v^*; as u^* is the parent of leaf v^* in T, we find that in fact T_{v^*} just consists of the leaf v^*. Hence the interior vertices of P_{xy} are contained in bag X_{v^*}, and as the interior vertices are not adhesion points, they are not contained in any other bag by the connectivity condition of tree decompositions. If we repeatedly contract interior vertices on P_{xy} into vertex x, we obtain a minor of $G - S$ that contains the edge xy; it is realized by the contraction of the path. The only vertices that are lost through these contractions belong to X_{v^*} and no other bag. Removing

the remaining vertices of X_{v^*} from this minor, we get a graph isomorphic to $G' - S$. As these operations leave the neighbors of S unchanged, we find that G' is a minor of G.

(3) Finally, consider an apex set S' in G', and assume that $G - S'$ is nonplanar. Let us first deal with the case that X_{v^*} contains at most one adhesion point; this implies that no biconnected component of G contains vertices of both X_{v^*} and $V(G) \setminus X_{v^*}$. By Proposition 5 a graph is planar if and only if all its biconnected components are planar. Hence there is a biconnected component of $G - S'$ that is nonplanar. Since $G - S$ is planar by the assumptions in the lemma, $X_{v^*} \subseteq V(G) \setminus S$, and $X_{v^*} \cap N_G(S) = \emptyset$, the graph $G[X_{v^*}]$ is planar. All biconnected components of $G' - S'$ are planar by assumption on S'. Hence all biconnected components of $G - S'$ are planar, contradicting the assumption that $G - S'$ is nonplanar.

In the remaining case, X_{v^*} contains two or more adhesion points; by (b) it then contains exactly two adhesion points, say x and y. Assume that S' is not an apex set in G, and consider a minor model of K_5 or $K_{3,3}$ in $G - S'$, i.e., a mapping ϕ from $V(H)$ ($H \in \{K_5, K_{3,3}\}$) to pairwise disjoint, connected subsets of $V(G - S')$, such that for each edge $pq \in E(H)$ the branch sets $\phi(p)$ and $\phi(q)$ are adjacent in $G - S'$.

Claim 8. The set $\{x, y\}$ separates $X_{v^*} \setminus \{x, y\}$ from $V(G) \setminus X_{v^*}$ in G.

Proof. Recall that v^* is a leaf of the tree decomposition and that u^* is the parent of v^* in the decomposition tree. By Proposition 11 the adhesion $\{x, y\}$ separates $X_{v^*} \setminus X_{u^*} = X_{v^*} \setminus \{x, y\}$ from the vertices $X_{u^*} \setminus X_{v^*} = X_{u^*} \setminus \{x, y\}$ in $G - S$. As all paths to the leaf v^* in the decomposition tree pass through its parent u^*, this easily implies that $\{x, y\}$ separates $X_{v^*} \setminus \{x, y\}$ from $V(G - S) \setminus X_{v^*}$ in $G - S$. Now observe that by assumption, $N_G(S) \cap X_{v^*} = \emptyset$. No vertex in S can therefore provide a detour from X_{v^*} to $V(G) \setminus X_{v^*}$ in G, showing that $\{x, y\}$ indeed separates $X_{v^*} \setminus \{x, y\}$ from $V(G) \setminus X_{v^*}$ in G. ◇

Armed with this claim we finish the proof by distinguishing two cases.

- If there is a vertex $p \in V(H)$ such that the entire branch set $\phi(p)$ is contained in $X_{v^*} \setminus \{x, y\}$, then we reason as follows. We claim that in this case, for all vertices $q \in V(H)$, the set $\phi(q)$ intersects X_v. To see this, assume for a contradiction that $\phi(q) \cap X_v = \emptyset$ for some $q \in V(H)$. By Claim 8 the adhesion vertices $\{x, y\}$ separate $\phi(p)$ from $\phi(q)$ in G. But as both K_5 and $K_{3,3}$ are triconnected, there must be at least three internally vertex-disjoint paths between $\phi(p)$ and $\phi(q)$ in $G - S'$ to realize the minor model of H. Hence the existence of a size-2 separator $\{x, y\}$ shows that ϕ is not a minor model of $H \in \{K_{3,3}, K_5\}$ in $G - S'$, contradicting our choice of minor model.

  In the remainder of the argument for this case we may therefore assume that for each $q \in V(H)$ we have $\phi(q) \cap X_v \neq \emptyset$. As each branch set induces a connected graph, and the vertices $\{x, y\}$ separate $X_{v^*} \setminus \{x, y\}$ from $V(G) \setminus X_{v^*}$ in G, by Claim 8, there are at most two branch sets containing vertices of $V(G) \setminus X_{v^*}$. If no branch set intersects $\{x, y\}$, then the entire minor model is contained in $X_v \setminus \{x, y\}$, and is therefore also a minor model in $G - S$; this contradicts the assumption that S is an apex set in G. Now suppose that a branch set, say $\phi(r)$, contains a vertex of $\{x, y\}$. Consider what happens to the minor model if we restrict each branch set to its intersection with X_v. The branch sets contained within X_v are not affected; hence we break at most two branch sets. Restricting them to X_v might cause them to become disconnected, or we might lose an edge between two different branch sets that was needed to represent an edge of H. But if we restrict each branch set to its intersection with X_v, and then add to the branch set of r an xy-path in $G - S$ whose interior avoids X_v, then it is easy to prove that all properties of the minor model are restored. As Proposition 12 ensures that such a path exists in $G - S$, we find a minor model of K_5 or $K_{3,3}$ in $G - S$, thereby contradicting the assumption that S is an apex set.

- In the second case, no branch set $\phi(p)$ for $p \in V(H)$ is a subset of $X_{v^*} \setminus \{x, y\}$. So all branch sets that intersect X_{v^*}, contain a vertex of $\{x, y\}$, from which it is easily verified that the only role $X_{v^*} \setminus \{x, y\}$ can play for the minor model, is to connect x and y together. Now restrict each branch set to its intersection with $V(G') \setminus S'$. Effectively, we are removing the vertices of $X_{v^*} \setminus \{x, y\}$ from the branch sets. When using these branch sets in graph G, removing the vertices of $X_{v^*} \setminus \{x, y\}$ might disconnect x and y; but as the edge xy was added in the construction of G', the branch sets form a proper minor model of the same graph H when interpreted in G'. As the branch sets do not contain vertices from S', we find that $G' - S'$ contains a Kuratowski graph as a minor, and is therefore nonplanar; a contradiction.

This concludes the proof of Lemma 6. $\square$

Lemma 7. *There is an $\mathcal{O}(kn)$-algorithm that, given a graph G and a $6k$-apex set S of G, constructs a minor G' of G without contracting edges incident on S, and a Tutte decomposition $(F', \mathcal{X}' = \{X'_i \mid i \in V(F')\})$ of $G' - S$, such that:*

1. *The leaves of the Tutte decomposition $(F', \mathcal{X}')$ intersect $N_{G'}(S)$.*
2. *Any apex set in G' is also an apex set in G.*

Proof. Assume that the stated conditions hold. As G has a $6k$-apex set, it has $\mathcal{O}(kn)$ edges. The goal of the algorithm is to apply Lemma 6 exhaustively in linear time. The algorithm proceeds as follows. We first invoke Proposition 14 to compute a Tutte decomposition $(F, \mathcal{X})$ of the graph $G - S$. Let $T_1, \ldots, T_\ell$ be the trees in F, corresponding to the connected components $C_1, \ldots, C_\ell$ of $G - S$. In $\mathcal{O}(kn)$ time we mark all neighbors of S by a Boolean flag in the vertex record. If there is a connected component C_i ($i \in [\ell]$) of $G - S$ that does not contain a neighbor of S, then it forms a planar connected component of G: it is planar as S is an apex set, and it is a component of G because it is not adjacent to S. Each such component C_i is clearly irrelevant to the planarization problem, and is therefore deleted from G. The associated tree T_i is removed from F.

For each connected component C_i of $G - S$ that does contain a neighbor of S, we do the following. We scan through the bags of T_i to find a node $j \in T_i$ whose bag X_j contains a neighbor of S. We root the tree T_i at j, and perform a tree traversal, starting at the root, that transforms the tree from the leaves upward. The transformation involves possibly adding some edges to the planar graph $G - S$. To avoid the difficulty of maintaining a datastructure that allows constant-time adjacency checks under dynamic edge insertions, we will allow this operation to create parallel edges in the graph. After the traversal these duplicate edges will be eliminated in one step.

When applied to a node $j \in V(T_i)$ of the Tutte decomposition, the traversal algorithm does the following. It first recursively calls itself on the children of j in T_i. Then it scans through the vertices in X_j to see whether there is a neighbor of S. If it finds one, it stops immediately. If X_j does not intersect $N_G(S)$, it tests whether j is currently a leaf of T_i. Note that the execution of the algorithm on the children of j might have resulted in those children being deleted, causing j to have become a leaf.

- If j is not a leaf, then the execution for node j stops.
- If j is indeed a leaf, the algorithm does the following. If j has a parent in T_i, such that the adhesion between j and its parent has size two, then it adds an edge between the two adhesion points. Since the adhesion between bags is stored in the edge records of the Tutte decomposition, this can be done in constant time. Note that the inserted

edge might be parallel to some existing edge. Observe that each time an edge is inserted between two adhesion points, a node of the Tutte decomposition is removed. Then the algorithm deletes j and its associated bag X_j from the decomposition, after which the execution for node j terminates.

Note that, apart from the possible creation of parallel edges that we shall deal with later, the transformation executed by the traversal algorithm corresponds exactly to that described in Lemma 6. Hence a simple induction shows that the transformations build a minor of the input graph, and that apex sets in the reduced graph are apex sets in G. As we are only modifying $G - S$, the vertices in S remain unaffected. From the structure of the traversal algorithm it is easy to see that on completion, each leaf in the remainder of the decomposition tree T_i has a bag containing a neighbor of S. It therefore remains to show that the algorithm runs in $\mathcal{O}(kn)$ time, and that we can eliminate the parallel edges.

Using n for the number of vertices in G, we know that G initially had $\mathcal{O}(kn)$ edges as it has a $6k$-apex set. For each edge that is added to G when forming G', we removed a node of the Tutte decomposition of a connected component. As the size of a Tutte decomposition is linear in the size of the graph it represents, by Proposition 10, the number of edges added to G is $\mathcal{O}(n)$. Hence G' has $\mathcal{O}(kn)$ edges. As each edge is represented by two integers in the range $[1 \dots n]$, we can sort the list of edges in $\mathcal{O}(kn)$ time by a radix sort, consisting of two stable bucket sorts. After sorting the list of edges, we compare successive entries in the list to detect duplicate edges. As each duplicate edge can be removed in constant time, we can reduce G' to a simple graph G'' in $\mathcal{O}(kn)$ time. The Tutte decomposition is not affected by the reduction to a simple graph. The output is graph G'', with the Tutte decomposition $(F', \mathcal{X}')$ that results from the modifications. It is computed in $\mathcal{O}(kn)$ time in total. $\square$

5.3 Covering Neighbors of the Approximate Solution by Constant-Radius Balls

Lemma 8. *There is an $\mathcal{O}(kn)$-time algorithm that, given a graph G, an integer k, a $6k$-apex set S of G, and a Tutte decomposition $(F, \mathcal{X})$ of $G - S$ whose leaves intersect $N_G(S)$, either:*

- *determines that there is no k-apex set in G that is disjoint from S, or*
- *computes for each $i \in V(F)$ a set $W_i \subseteq N_G(S) \cup \mathrm{Adh}(F, \mathcal{X})$ of size at most $4k$ such that $(N_G(S) \cup \mathrm{Adh}(F, \mathcal{X})) \cap X_i \subseteq \mathbf{R}^7_{Torso(G-S,X_i)}(W_i)$, i.e., all adhesion points in X_i and all neighbors of S in X_i are contained in the radial 7-balls around W_i.*

Proof. Assume the stated conditions hold. As S is a k-apex set, the number of edges in G is $\mathcal{O}(kn)$. Let $Y := N_G(S) \cup \mathrm{Adh}(F, \mathcal{X})$, and for $i \in V(F)$ let $Y_i := Y \cap X_i$. The algorithm does the following for each node i of the Tutte decomposition.

Consider the graph $\mathrm{Torso}(G - S, X_i)$, and let $n_i := |X_i|$. We greedily compute a maximal set $W_i \subseteq Y_i$ such that for all pairs of distinct vertices $u, w \in W_i$ the radial distance $d_{\mathrm{Torso}(G-S)}(u, w)$ is at least seven. To do this in $\mathcal{O}(n_i)$ time we store a variable for each vertex in X_i, which indicates its radial distance to the nearest member of Y_i, or $+\infty$ if this distance is at least seven; initially all distance tracking variables are $+\infty$. We process the members of Y_i one at a time. When inspecting $z \in Y_i$, one of the following happens. If the distance value of z is less than seven, then we ignore z and continue. If the distance value of z is $+\infty$ (i.e., at least seven) then we add z to W and update the distance estimates using a breadth-first search in the radial graph of $G - S$. When the BFS discovers a vertex whose distance value is not lowered by the discovery, then it is not added to the BFS queue and not handled further; the BFS also stops once the radial distance from v exceeds six. It is easy to see that this BFS process correctly updates the distance values. To see that the construction of W_i takes $\mathcal{O}(n_i)$ time, observe that the distance value of a vertex can be lowered at most seven times before becoming zero. Hence each vertex of G is processed constantly many times by the BFS, and as $\mathrm{Torso}(G-S, X_i)$ is planar this results in the desired runtime.

Because the described process adds all vertices in Y_i to W_i for which the radial distance to the other members of W_i is seven or more, we find a maximal set $W_i \subseteq Y_i$ with the property that $d_{\mathrm{Torso}(G-S,X_i)}(u,w) \geq 7$ for all distinct $u, w \in W_i$. Hence any vertex in Y_i has radial distance at most seven to a vertex in the final set W_i, as required by the lemma statement. Performing this computation once for each node of F results in the desired sets W_i, for a total runtime of $\mathcal{O}(kn)+\sum_{i\in V(F)} \mathcal{O}(n_i) = \mathcal{O}(kn+n) = \mathcal{O}(kn)$ (Proposition 10). The first term $\mathcal{O}(kn)$ accounts for a single pass over the entire graph G in which the neighbors of S are identified by a Boolean flag. If all sets W_i' computed in this way have size at most $4k$, we output these sets and we are done. If there is a set whose size exceeds $4k$, then we output NO: there is no k-apex set in G disjoint from S. To establish the correctness of this step, we prove that if G has a k-apex set disjoint from S, then $|W_i| \leq 4k$ for all nodes $i \in V(F)$. We need the following claim.

Claim 9. For each $i \in V(F)$ there is a minor G_i of G on vertex set $X_i \cup S$, such that $G_i[X_i] = \mathrm{Torso}(G, X_i)$ and for every vertex $v \in Y_i$ the set $N_{G_i - S}[v]$ contains a vertex in $N_{G_i}(S)$.

Proof. Build a minor G_i of G as follows. Consider the tree T_i in F that contains i. By the definition of Tutte decomposition it corresponds to some connected component C_i of $G-S$. For each component of $G-S$ different from C_i, delete its vertices from G. Then, repeatedly do the following. As long as there is an edge for which one endpoint lies in X_i and the other endpoint is not in $X_i \cup S$, contract the edge into its neighbor in X_i. Let G_i be the resulting graph. It is obviously a minor of G, and it is made without contracting edges incident on S. To see that $G_i[X_i] = \mathrm{Torso}(G, X_i)$, note that for every edge $pq \in E(\mathrm{Torso}(G, X_i)) \setminus E(G)$, there is a pq-path in $G-S$ whose internal vertices do not lie in X_i. Hence the internal vertices on this path are represented in bags in a single component of $T_i - i$, the tree T_i after removing i. The adhesion of node i to that part of the tree has size at most two. Hence, when exhaustively contracting vertices into X_i, we eventually contract the path into either p or q, thereby creating the edge pq in the minor. Thus every edge in $E(\mathrm{Torso}(G, X_i)) \setminus E(G)$ is created by the contractions, showing that $G_i = \mathrm{Torso}(G, X_i)$. It remains to prove that for each $v \in Y_i$ we have $N_{G_i-S}[v] \cap N_{G_i}(S) \neq \emptyset$.

So consider a vertex $v \in Y_i = (N_G(S) \cup \mathrm{Adh}(F, \mathcal{X})) \cap X_i$. So vertex v is either a neighbor of S belonging to $\mathrm{Torso}(G, X_i)$, or it is an adhesion vertex in bag X_i. In the first case it is easy to see that v is a neighbor of S in G_i, and therefore that the closed neighborhood of v intersects Y_i, as the minor operations have not removed edges between X_i and S. The second case is more interesting; we have to use the fact that the leaves of F intersect $N_G(S)$.

So suppose that v is an adhesion vertex in bag X_i, which is not adjacent to S. By the definition of adhesion, there is a node j that is adjacent to i in T_i, which also contains v. When removing node i from T_i, there is exactly one tree in the resulting forest that contains node j: let it be T_j. We claim that one of the bags belonging to T_j contains a vertex in $N_G(S)$. If tree T_j has a leaf l distinct from j, then l is also a leaf of F, showing that X_l intersects $N_G(S)$. If j is the only leaf in T_j, then T_j is a single vertex and therefore it is also a leaf in F; hence X_j intersects $N_G(S)$. In both cases we find that T_j contains a bag intersecting $N_G(S)$. Let $z \in N_G(S)$ be contained in a bag in T_j. As v is not adjacent to S, we have $z \neq v$. During the contraction process, we might contract z into other vertices, but note that the adjacency to S is preserved by such contractions. As we contract all edges with one endpoint in X_i and the other outside $X_i \cup S$, and the adhesion between node i and node j forms a separator by Proposition 11, we eventually contract z into either v, or into the unique other vertex in the intersection $X_i \cap X_j$. If the adhesion has size one, then we must contract z into v, making v adjacent to S in G_i. If we do not contract z into v, then the contraction makes the other adhesion vertex between i and j

adjacent to S, and realizes the edge between z and that adhesion vertex. Hence z has a neighbor in G_i that is adjacent to S, proving the claim. $\diamondsuit$

Assume that there is a k-apex set in G disjoint from S, and let G_i be a minor of G as described in Claim 9. By Proposition 1, our assumption implies that there is a k-apex set S' in G_i that is disjoint from S. Consider the connected components $C_1, \ldots, C_\ell$ of $(G_i - S) - S'$. As $G_i - S = G_i[X_i] = \text{Torso}(G, X_i)$, it is a triconnected planar graph. As Proposition 6 shows that removing $|S'| \geq 1$ vertices from a triconnected graph results in at most $2|S'|$ connected components, we find that $\ell \leq 2k$.

For every pair of distinct vertices $u, w \in W_i$ we have that $\mathbf{R}^3_{\text{Torso}(G-S,X_i)}(u)$ and $\mathbf{R}^3_{\text{Torso}(G-S,X_i)}(w)$ are disjoint, as the radial distance between them is seven or more. Thus there are at most $|S'| \leq k$ vertices u in W_i such that $\mathbf{R}^3_{\text{Torso}(G-S,X_i)}(u) \cap S' \neq \emptyset$. Let W_i' contain the remaining vertices of W_i and observe that $|W_i'| \geq |W_i| - k$. For each $w \in W_i'$, we have $N_{G_i}[w] \cap N_{G_i}(S) \neq \emptyset$, by Claim 9. Therefore we can associate to each $w \in W_i'$ an edge $e(w) \in E(G_i)$, such that one endpoint of $e(w)$ lies in $N_{G_i}[w] \setminus S$, and the other lies in S. Now construct a bipartite multigraph H_i as follows. One partite set has a vertex for each connected component of $(G_i - S) - S'$; the other partite set is S. For each vertex $w \in W_i'$, we add an edge between the component that contains w and the S-endpoint of the edge $e(w)$. Note that in this way, we possibly add parallel edges. The number of edges in H_i equals the number of vertices in W_i'.

Claim 10. Graph H_i is a minor of $G_i - S'$.

Proof. We show how to obtain H_i by performing minor operations. As H_i is possibly a multigraph, we allow the minor operations to create parallel edges. We discard loops that might be created by edge contractions, though.

Starting with the graph $G_i - S'$, first remove all edges incident on S, except for the edges $e(w)$ for $w \in W_i'$. Then contract all edges whose endpoints belong to the same connected component of $(G_i - S) - S'$. These minor operations result in the graph H_i. It is obvious that the vertex set of the resulting minor consists of S, together with one vertex for each connected component of $(G_i - S) - S'$. To see that the adjacencies between the components and S are identical to those in H_i, observe that for each vertex $w \in W_i'$, we have $\mathbf{R}^3_{\text{Torso}(G-S,X_i)}(w) \cap S' = \emptyset = \mathbf{R}^3_{G_i - S}(w) \cap S'$. Hence the vertices of S' do not intersect the three-balls around vertices in W_i'. As each edge $e(w)$ for $w \in W_i'$ has an endpoint in $N_{G_i - S}[w]$, and the closed neighborhood of w is contained in the one-ball around w, the endpoint of the edge belongs to the same connected component of $(G_i - S) - S'$ as w. Therefore the component containing w receives the edge to the corresponding member of S, showing that indeed the connections between the components and S in the constructed minor, match the adjacencies in H_i. $\diamondsuit$

Claim 11. The graph H_i is acyclic. In particular, it does not have parallel edges.

Proof. Assume that H_i has a cycle C. We will use it to show that $G_i - S'$ is nonplanar. As H_i is a multigraph without loops, a cycle either consists of two parallel edges, or of a simple cycle of length at least three. Since H_i is bipartite and S is one partite set, the cycle contains a vertex d of H_i in the other partite set, which corresponds to a connected component D of $(G_i - S) - S'$. There are exactly two edges incident on d in the cycle C, say e_1 and e_2. The edges of H_i correspond naturally to the edges $e(w)$ for $w \in W_i'$. Let $w_1 \in W_i'$ be the vertex such that e_1 corresponds to $e(w_1)$, and similarly let $w_2 \in W_i'$ be such that e_2 corresponds to $e(w_2)$. As each vertex in W_i' corresponds to one edge in H_i, and $e_1 \neq e_2$, we find that $w_1 \neq w_2$.

Each $e(w_j)$ for $j \in [2]$ is an edge in G_i with one endpoint in S, and the other in $N_{G_i}[w_i] \setminus S = N_{G_i - S}[w_i]$, by definition of $e(\cdot)$. Let $x_j \in N_{G_i - S}[w_j]$ be the latter such endpoint of $e(w_j)$, for $j \in [2]$. As $G_i - S = G_i[X_i] = \text{Torso}(G - S, X_i)$, the graph $G_i - S$ is triconnected by the definition of Tutte decomposition. As the radial distance between x_j and w_j in $G_i - S$ is at most one (since $x_j \in N_{G_i - S}[w_j]$), we have $\mathbf{R}^2_{G_i-S}(x_j) \subseteq \mathbf{R}^3_{G_i-S}(w_j)$ for $j \in [2]$. Hence the radius-two balls around x_1 and x_2, $\mathbf{R}_{G_i-S}(x_1)$ and $\mathbf{R}_{G_i-S}(x_2)$ are disjoint from each other, and disjoint from S'. This implies that the radial distance between x_1 and x_2 in $G_i - S$ is at least three. Now observe that this still holds in the graph $(G_i - S) - S'$: as the vertices we remove are not contained in the radius-two balls around x_1 and x_2, those radius-two balls are unaffected by the removal from S', certifying that $d(G_i - S) - S'(x_1, x_2) \geq 3$. We use this to build a K_5 minor model in the graph $G_i - S'$, as follows.

By Proposition 3 there is a cycle C_1 in $(G_i - S) - S'$ that separates x_1 from x_2, such that $d_{(G_i-S)-S'}(x_1, c) = 1$ for all $c \in C_1$. Similarly, there is a cycle C_2 separating x_1 from x_2 whose vertices lie at radial distance two. Hence C_1 and C_2 are connected, vertex-disjoint separators, showing that they must be nested: C_1 separates x_1 from C_2 in $(G_i - S) - S'$. As $G_i - S$ is triconnected, and its the vertices at radial distance at most two from x_1 are also contained in $(G_i-S)-S'$, there are three paths P_1, P_2, P_3 in $\mathbf{R}^2_{(G_i-S)-S'}(x_1)$ that each have x_1 as one endpoint, their other endpoint lies on C_2, and are internally vertex-disjoint. (If such paths do not exist, then there is a size-2 separator between x_1 and C_2 in $(G' - S) - S'$, and therefore also in $G' - S$, by Menger's theorem, contradicting triconnectivity.) It is straight-forward, but tedious, to show that we may assume that the intersection of P_1, P_2, P_3 with C_1 is a subpath of the cycle C_1; see for example [24, Lemma 2.3] or [12, Lemma 5.4]. The final piece we need to construct the K_5-minor model in $G_i - S'$ is a path from x_1 to C_2, whose interior is disjoint from P_1, P_2, P_3, C_2. The key insight for this step is that we can use the presumed cycle C in graph H_i to give such a path. In H_i, we can follow the cycle from vertex d corresponding to the component we are working in, starting with the edge e_1, tracing along the cycle until finally using edge e_2 to arrive back at vertex d. As H_i is a minor of $G_i - S'$, and none of this path in H' along the cycle corresponds naturally to a path in $G_i - S'$. As we follow the cycle until traversing the edge e_2 that takes us back to d, none of the vertices visited in between belong to D. As e_2 corresponds to $e(w_2)$, whose endpoint in D is x_2, we find a path $P_{x_1x_2}$ in $G_i - S'$ whose endpoints are x_1 and x_2, and whose internal vertices do not belong to component D. Finally, observe that as x_1 and x_2 belong to the same connected component D of $G_i - S'$, there is a path $P_{x_2C_2}$ from x_2 to C_2 through the component. Taking a shortest path, it hits C_2 (at radial distance two from x_1) before visiting any internal vertex of P_1, P_2, P_3. Concatenating P_{x_1,x_2} with $P_{x_2C_2}$ gives a path $P_{x_1C_2}$ from x_1 to C_2 as desired; it is internally vertex-disjoint from P_1, P_2, P_3 and C_1. These ingredients suffice to make a K_5 minor, as follows.

As the intersections of the paths P_j for $i \in [3]$ are subpaths of cycle C_1, we can contract edges on the cycle C_1 to make all pairs among P_1, P_2, P_3 adjacent. Afterward, the vertex sets of the interiors of P_1, P_2, P_3, the set $\{x_1\}$, and $(P_{x_1C_2} \cup C_2$ form the branch sets of a K_5 minor: the paths P_j connect to eachother through the cycle, the P_j's connect to x_1, and to C_2, since they are $x_1 - C_2$ paths by construction, and finally x_1 connects to $(P_{x_1C_2} \cup C_2$ by the construction of $P_{x_1C_2}$. As the vertex sets are vertex-disjoint and connected, these indeed form a K_5 minor in $G_i - S'$, contradicting the assumption that $G_i - S'$ is planar. $\diamondsuit$

Using Claim 11 we can complete the proof. The vertex set of H_i consists of two parts: the $\ell \leq 2k$ connected components of $(G_i - S) - S'$, and the set S' of size at most k. As it is acyclic, it is a forest and its edge count is smaller than its vertex count. Hence $|E(H_i)| \leq |V(H_i) \leq 3k$. As the edges of H_i directly correspond to vertices in W'_i, we find that $|W'_i| \leq 3k$. As $|W'_i| \geq |W_i| - k$, this shows that $|W_i| \leq |W'_i| + k \leq 4k$. We have therefore shown that if G has a k-apex set disjoint from S, then all the sets W_i have size at most $4k$. This concludes the proof of Lemma 8. $\square$

5.4 Dividing the Graph into Bounded-Width Regions

The following definition gives the central notion of this section.

Definition 2. A (λ, β)-*thinning set* for $[0 \ldots \ell]$ is a subset of $A \subseteq [0 \ldots \ell]$ with the following properties:

- The members of A can be partitioned into blocks of β integers each, such that each group contains consecutive integers.
- For each $i \in [0 \ldots \ell - \lambda]$, the set A contains an element of $[i \ldots i + \lambda]$.

Thinning sets will be used to formalize the idea of reducing the treewidth of a planar graph by reducing the number of vertex-disjoint paths between distant parts of it. Before going into technical details, let us describe how thinning sets will be used. For a plane graph G, recall that $\mathrm{rad}(G)$ denotes the largest index of a non-empty outerplanarity layer of G. Suppose that $A \subseteq [0 \ldots \mathrm{rad}(G)]$ is a (λ, β) thinning set for the set of indices of outerplanarity layers of a plane graph G. Consider what happens when removing from G all the vertices in outerplanarity layers whose index is contained in A. Since A contains a layer in $[i \ldots i+\lambda]$ for each $i \in [\ldots \ell-\lambda]$, removing the layers indexed by A splits the graph into connected components of radius at most λ, and therefore reduces the treewidth to $\mathcal{O}(\lambda)$ by Proposition 8. Hence the λ parameter, which will be $\mathcal{O}(k)$ in our applications, controls the treewidth reduction. The β parameter controls how many contiguous layers are contained in each block of the thinning set. The layers in a thinning set will be declared undeletable. By demanding that a block of layers contains a sufficient (but constant) number of contiguous layers, we ensure that the reduction rule described in Section 3 can be applied within such a block to reduce the number of disjoint-paths that can be routed through it, because in the presence of many such paths the block will contain two disjoint nested undeletable planarizing cycles around a vertex, which will be shown to be irrelevant. With this intuition, we continue the technical discussion.

Definition 3. Consider a plane graph G with outerplanarity layers $L_0, \ldots, L_{\mathrm{rad}(G)}$, along with a vertex set $Y \subseteq V(G)$. A (λ, β)-thinning set $A \subseteq [0 \ldots \mathrm{rad}(G)]$ *avoids* Y if, for all layers L_i that contain a vertex of Y, we have $i \notin A$. Conversely, the (λ, β)-thinning set A *covers* Y if, for all layers L_i that contain a vertex of Y, we have $i \in A$.

The following lemma will be applied independently to each torso of the Tutte decomposition to divide it into thinning sets. The disjoint thinning sets $S_1, \ldots, S_{k+1}$ described in the lemma will be used to branch into $k + 1$ ways. In each branch, we will declare the vertices covered by the thinning set to be undeletable and reduce the treewidth of the instance based on that assumption. If a k-apex set exists, then there is at least one thinning set that avoids the apex set, and whose vertices can therefore be assumed to be undeletable while preserving the existence of a solution.

Lemma 9. *There is an $\mathcal{O}(kn)$-time algorithm that, given a plane graph G with outerplanarity layers $L_0, \ldots, L_{\mathrm{rad}(G)}$, a vertex set W of size at most $4k$, and integers k and $\beta \geq 2$, partitions the integers $[0 \ldots \mathrm{rad}(G)]$ into $k + 2$ groups $S_0, S_1, \ldots, S_{k+1}$, such that for $i \in [1 \ldots k + 1]$, the set S_i is a (λ, β)-thinning set that avoids $Y := \mathbf{R}^7_G(W)$ in G, with $\lambda := \beta^2(13|W| + k + 4)$.*

Proof. We find the vertices in Y. Toward this we construct the radial graph R_G in time $\mathcal{O}(n)$ and compute $N^{14}_{R_G}(w)$ for all the vertices in W. We can compute $N^{14}_{R_G}(w)$ for all the vertices in W in time $\mathcal{O}(kn)$. Clearly, $Y := \mathbf{R}^7_G(W) = N^{14}_{R_G}(W) \cap V(G)$. Let S^1_0 denotes the set of indices of all those layers that contain a vertex of Y. Now we group the numbers into blocks of β integers each. Let $\ell = \lfloor \frac{\mathrm{rad}(G)}{k+1} \rfloor$. Define $\Gamma_i = \{\beta(i-1)+1, \ldots, \beta i\}$ where $i \in \{1, \ldots, \ell\}$. Let S^2_0 denotes those integers in $[0 \ldots \mathrm{rad}(G)]$ that do not appear in any Γ_i. We call Γ_i *good* if $\Gamma_i \cap S^1_0 = \emptyset$ and *bad* otherwise. Now, we go over the blocks from left to right, if a block is bad, ignore it; if it is good, give it to S_i and increase i. Of course if $i = k + 1$ then decrease it to 1. Formally, we do as fellows. Let P denote the set of integers x in $[\ell]$ such that Γ_x is good.

Now we order the vertices in P in the increasing order. Let this ordering be $\sigma_1, \dots, \sigma_{|P|}$. For all integers $i \in [k+1]$, define

$$S_i = \bigcup_{\{x \mid (x \bmod k+1)=i \text{ and } x \le |P|\}} \Gamma_{\sigma_x}.$$

Furthermore, define

$$S_0 = \bigcup S_0^2 \bigcup_{x \in [\ell]-P} \Gamma_x.$$

By construction, for $i \in [1 \dots k+1]$, the set S_i avoids Y and its members can be partitioned into blocks of β integers each, such that each group contains consecutive integers. The only thing that remains to show is that for each $i \in [0 \dots \mathrm{rad}(G) - \lambda]$, the set S_i contains an element of $[i \dots i+\lambda]$. Observe that if a vertex $v \in L_x$, then $u \in \mathbf{R}^c_G(v)$ must lie in L_y, where $y \in \{x-(c-1), \dots, x+c-1\}$. This implies that $|S_0| \le 13\beta|W| + \beta - 1 < \beta(13|W|+1)$. To prove our claim we need to show that there are at least $k+1$ good blocks in $[i \dots i+\lambda]$. The number of blocks in in $[i \dots i+\lambda]$ is at least $\lfloor \frac{\lambda}{\beta} \rfloor - 2 \ge \frac{\lambda}{\beta} - 3$. Thus, the number of good blocks is at least $\frac{\lambda}{\beta} - 3 - \beta(13|W|+1) \ge \beta(k+4) - 3 - \beta \ge (k+1)$. Thus, by construction at least one of these blocks must appear in S_i. This completes the proof. □

5.5 Reducing the Flow through Undeletable Parts

5.5.1 Properties of Outerplanarity Layers

We shall need a few simple properties of outerplanarity layers.

Lemma 10. *Let G be a connected plane graph with outerplanarity layers $L_0, L_1, \dots, L_{\mathrm{rad}(G)}$. For $i \in [0, \dots, \mathrm{rad}(G)]$, define $G_i := G[\bigcup_{j=0}^{i} L_j]$ and $G_{\ge i} := G[\bigcup_{j \ge i} L_j]$. The following is true.*

1. *For every $i \in [0, \dots, \mathrm{rad}(G)]$, graph G_i is connected.*
2. *If C is a connected component of $G_{\ge i}$, then:*
   - *(a) for all $i \le j \le \mathrm{rad}(G)$, the graph $G[C \cap (\bigcup_{i \le x \le j} L_x)]$ is also connected, and*
   - *(b) if $i \ge 1$ there is a connected component C' of $G[L_{i-1}]$ such that $N_G(C) \subseteq C'$.*

Proof. (1) The first property can be proved using induction on i. Let $i = 0$. The set L_0 contains the vertices on the outer face of G. Since G is connected, the vertices of L_0 form a closed walk in G, and hence $G[L_0]$ is connected. Suppose we have proved that G_i is connected and now we want to show that G_{i+1} is connected. By definition the vertices of L_{i+1} are on the outer face of $G_{\ge i+1} = G - G_i$. Let $C_1, \dots, C_\ell$ be the connected components of $G_{\ge i+1}$ and let O_p, for $p \in [\ell]$, denote the set of vertices on the outer face of C_p. Thus, $L_{i+1} = \bigcup_{p \in [\ell]} O_p$. Furthermore, we know that O_p is connected. To show that G_{i+1} is connected we only need to argue that for every $p \in [\ell]$, there is a vertex in O_p that has a neighbor in $\bigcup_{j=1}^{i} L_j$. Suppose not. Then since G is connected we have that there is a vertex, say y, in $C_p - O_p$ that has a neighbor, say z, in $\bigcup_{j=1}^{i} L_j$. But then by Observation 2, since adjacent vertices are incident to a common face, the difference between the outerplanarity indices of y and z must be at most one, a contradiction. This concludes the proof of the first part of the lemma.

For the last two properties, consider a connected component C of $G_{\ge i}$ for some i.

(2a) Observe that $L_i \cap C, L_{i+1} \cap C, \dots, L_t \cap C$ are the outerplanarity layers of C. Now applying the first part of the lemma to C and these layers yields the desired claim.

(2b) Finally, we prove that there is a connected component C' of $G[L_{i-1}]$ such that $N_G(C) \subseteq C'$, if $i \ge 1$. Consider the defining process for outerplanarity layers, where we repeatedly delete the vertices on the outer face. Consider the graph $G_{\geq i-1}$, which is obtained after removing the first $i-2$ layers. Since all vertices in C have outerplanarity index at least i, none of them are on the outer face of the obtained drawing of $G_{\geq i-1}$ (else they would have index at most $i-1$). Observe that each connected component of $G_{\geq i}$ is drawn entirely within one face of $G[L_{i-1}]$, since it is a connected part of the drawing of G by (2a). Hence C is drawn within one face of $G[L_{i-1}]$, say f^*. By definition, no vertex of C is adjacent to a vertex with outerplanarity index at least i in G; otherwise that vertex would belong to the same connected component C of $G_{\geq i}$. Hence, using Observation 2, all neighbors of C have outerplanarity index exactly $i-1$, and exist in the graph $G[L_{i-1}]$. By the plane embedding of G, if a vertex of C, that is drawn within the region of the plane formed by f, has a neighbor in $G[L_{i-1}]$, then this neighbor lies on face f. Hence all neighbors of C lie on the boundary of f.

Since f is a face in the outerplanar graph $G[L_{i-1}]$ that is not the outer face, the boundary of f is a connected set of vertices in $G[L_{i-1}]$, and therefore in G: we can traverse the boundary walk of f to connect the vertices. Hence all vertices on the boundary of f belong to the same connected component of $G[L_{i-1}]$. Since all neighbors of C are contained in the boundary of f, this proves the claim. $\square$

Lemma 11. *Let G be a connected plane graph. Let T be the graph obtained from G by contracting all edges whose endpoints have the same outerplanarity index. Then T is a tree.*

Proof. The prove the statement by induction on $\mathrm{rad}(G)$, the index of the largest nonempty outerplanarity layer of G. If $\mathrm{rad}(G) = 0$ then G is outerplanar and all vertices are incident with the outer face. Hence all vertices have the same outerplanarity index. Since G is connected, contracting all edges results in a single vertex, which is a tree.

Now consider a graph G with nonempty outerplanarity layers $L_0, \ldots, L_{\mathrm{rad}(G)}$ for $\mathrm{rad}(G) > 0$. Let $G' := G[L_0, \ldots, L_{\mathrm{rad}(G)-1}]$. By removing the vertices of $L_{\mathrm{rad}(G)}$ from the drawing of G, we can get an embedding of G' whose outerplanarity layers are $L_0, \ldots, L_{\mathrm{rad}(G)-1}$. By (1) of Lemma 10 the graph G' is connected. Let T' be the graph resulting from G' by contracting all edges between vertices whose outerplanarity indices coincide. By induction, T' is a tree.

Now consider the vertices of T that are not in T'. Since each connected component of $G[\mathrm{rad}(G)]$ is contracted into a single vertex, the vertices $V(T) \setminus V(T')$ correspond exactly to the connected components of $G[\mathrm{rad}(G)]$. To prove that T is a tree, we show that each such vertex has degree exactly one in T, and thus that T can be obtained from T' by adding a sequence of leaf vertices to the tree T'. Since adding leaves to a tree preserves the fact that the graph is a tree, this will yield the proof.

So consider a vertex $v \in V(T) \setminus V(T')$, and let C be the corresponding connected component of $G[\mathrm{rad}(G)]$. By (2b) of Lemma 10, the neighbors of C are contained in a single connected component C' of $G[L_{i-1}]$. Since G is connected, there is at least one such neighbor. Consequently, all the vertices of C' are contracted to a single vertex u in T'. Thus the unique neighbor of v in the graph T is the vertex u, proving that v is a leaf in T. $\square$

Lemma 12. *Let G be a connected plane graph with outerplanarity layers $L_0, L_1, \cdots, L_{\mathrm{rad}(G)}$. Let S be a subset of $[0 \ldots \mathrm{rad}(G)]$. Let G^* be the graph obtained from G by contracting all edges whose endpoints both belong to layers of S, or both do not belong to layers of S. Then G^* is a tree.*

Proof. We split the process of contracting G into G^* in two steps. Since the order of edge contractions does not affect the end result, we can obtain G^* by first contracting all edges of G whose endpoints have the same outerplanarity index, creating a graph T, and then creating edges in T between vertices whose outerplanarity indices are either both contained in S, or both not contained in S. Observe that since we only merge vertices with identical outerplanarity indices in the first step, this is well defined. By Lemma 11, the graph T is a tree. Since G^* is a contraction of T, it is a tree as well. $\square$

5.5.2 Finding Irrelevant Vertices

Armed with these structural properties of outerplanarity layers, we prove the following lemma. It is the main algorithmic ingredient of the treewidth reduction scheme.

Lemma 13. *There is an $\mathcal{O}(k^{\mathcal{O}(1)}n)$-time algorithm that, given a triconnected plane graph G with outerplanarity layers $L_0, \ldots, L_{\mathrm{rad}(G)}$, a vertex set $W \subseteq V(G)$ of size at most $4k$, and a (λ, β)-thinning set $A \subseteq [0 \ldots \mathrm{rad}(G)]$ that avoids $\mathbf{R}^7_G(W)$, computes a vertex set $Z \subseteq V(G)$ and a tree decomposition $(T, \mathcal{X})$ of $G - Z$ of width $\mathcal{O}(k)$, such that for each vertex $v \in Z$ there are two vertex-disjoint cycles $C_1, C_2 \subseteq V(G) \setminus (\mathbf{R}^7_G(W) \cup Z)$ whose vertices are covered by A, such that, in graph G, $V(C_1)$ separates v from $\mathbf{R}^7_G(W)$, and $V(C_2)$ separates v from $V(C_1)$. Here, $\beta = 35$ and $\lambda = 70000k$.*

Proof. Our proof is divided into three parts. In the first part we show how to identify the desired set Z. As explained in Section 3, the set Z will consist of two types of irrelevant vertices, Z_1 and Z_2. In the second part we show that for each vertex $v \in Z$ there are two vertex-disjoint cycles $C_1, C_2 \subseteq V(G) \setminus (W \cup Z)$ whose vertices are covered by A, and which satisfy the claimed separation conditions. Finally, we give the desired tree decomposition $(T, \mathcal{X})$ of $G - Z$ of width $\mathcal{O}(k)$. Throughout the proof, we utilize that the outerplanarity layers of G correspond directly to the vertex-layers of a breadth-first search in the radial graph R_G of G, that starts at the vertex that represents the outer face.

Identification of Z. Let $Q_1, \ldots, Q_t$ be the blocks of β consecutive integers in A, in increasing order. For a given block $Q_j = \{s_j, s_j+1, \ldots, s_j+\beta-1\}$, $j \in [t]$, we define $R_j = \{s_j+2, \ldots, s_j+\beta-3\}$ and $U_j = \{s_j+8, \ldots, s_j+\beta-9\}$. So R_j is the subblock of Q_j obtained by omitting its first and last two layers, while U_j is the even thinner block obtained by omitting Q_j's first and last eight layers. For $X \subseteq A$, we define $V(X)$ as $\bigcup_{i \in X} L_i$. We use V_Q as an abbreviation for $\bigcup_{i=1}^t V(Q_i)$. We identify the vertex set Z in two phases. In what follows we give the description of the first phase, leading to the identification of Z_1. For $i \in \{0, \ldots, t\}$, let M_i be defined as follows:

$$M_i = \begin{cases} \bigcup_{x < s_i+2} L_x & \text{if } i = 0\ , \\ \bigcup_{s_i+\beta-2 \le x < s_{i+1}+2} L_x & \text{if } 1 \le i \le t-1, \\ \bigcup_{x \ge s_i+\beta-2} L_x & \text{if } i = t. \end{cases}$$

Furthermore, let $R := \bigcup_{i=1}^t V(R_i)$ and $M := \bigcup_{i=0}^t M_i$. Obtain the graph G^* from G by contracting every edge e of G for which both endpoints belong to R, and those for which both endpoints belong to M. Each vertex v of G^* then corresponds to a connected set S_v in G. Each S_v is either contained in R_i or in M_i. Let $\widehat{R}_i$ contain all vertices v in G^* for which $S_v \subseteq V(R_i)$. Similarly, we define $\widehat{M}_i$. Applying Lemma 12 with $S := R$, we have that G^* is a tree. Observe that we can get G^* in time $\mathcal{O}(kn)$.

Let $\widehat{M} = \bigcup_{j=0}^t \widehat{M}_j$. By Lemma 10 we know that $G[M_0]$ is a connected set and thus there exists a vertex v'_r corresponding to it in G^*. We root the tree G^* in v'_r. Now, we classify every vertex v of $\widehat{M}$ into interesting or boring, by a bottom-up process: a vertex $v \in \widehat{M}$ is *interesting* if (i) one of its descendants is interesting, or (ii) the set S_v has a nonempty intersection with $\mathbf{R}^7_G(W)$. Otherwise it is *boring*. It is easy to implement this procedure in $\mathcal{O}(kn)$ time. In this way we classify each vertex of $\widehat{M}$ as interesting or boring; the vertices of $\widehat{R}$ are not classified. Let Z_1^* consist of all the vertices in S_v for which v has been classified as boring. The first set of irrelevant vertices, Z_1, is defined as $Z_1 = Z - V_Q$. The procedure described above can clearly be implemented in time $\mathcal{O}(kn)$.

Claim 12. Consider the vertex set $\widehat{R}_i$ in G^*. The children of $\widehat{R}_i$ are vertices in $\widehat{M}_i$. Let μ_i denote the number of children of $\widehat{R}_i$ in G^* that have been classified as interesting. For any $i \in [t]$, $\mu_i \leq |W|$.

Proof. Let $v_1, \dots, v_\gamma$ denote children of $\widehat{R}_i$ in G^* that have been classified as interesting. Recall that the thinning set A avoids $\mathbf{R}^7_G(W)$. As G is a connected graph, for each vertex $w \in W$ the set $\mathbf{R}^7_G(w)$ is connected. As none of its vertices are contained in a layer of the thinning set, during the contraction process all vertices of $\mathbf{R}^7_G(w)$ have been contracted into a single point, for each $w \in W$. Therefore a vertex $w \in W$ can only cause one child of $\widehat{R}_i$ to be interesting. This implies that μ_i is upper bounded by $|W|$. ♢

Now we describe the second phase of the identification of irrelevant vertices, leading to Z_2. The set Z_2 will consist of some vertices contained in $\bigcup_{i=1}^t U_i$, that is, it will contain vertices from the thinnest subblocks of the Q_i's, the blocks of the thinning set. The layers of Q_i that are not contained in U_i, will never be marked irrelevant, and will provide the insulation of the irrelevant vertices from the nonplanar parts. So we will perform some computations on the graphs induced by the blocks Q_i to find the set Z_2. To ensure we do not have to inspect the entire graph too often, for each $i \in [t]$ we zoom in on the last two layers of Q_i. Note that the tree G^* was obtained by contractions with respect to the R_j's, the subblocks of the Q_j's obtained by slicing off the first and last two layers. Hence for a vertex $v \in \widehat{M}$ that is marked as interesting, the connected set S_v in G that it corresponds to, can intersect the last two layers of exactly one Q_is. We will therefore keep track of the interesting vertices in $\widehat{M}$ by their intersections with the last two layers of the blocks Q_i. Toward that end, we define for all $i \in [t]$ the graph

$$\widehat{G}_i = G[\bigcup_{x \in \{s_i+\beta-2, s_i+\beta-1\}} L_x],$$

which is the subgraph of G induced by the last two layers in Q_i. We classify the connected components of $\widehat{G}_i$ into boring of interesting, based on the classification of vertices of G^*. We again do a bottom-up traversal in G^*, from the leaves to the root v'_r. For every interesting vertex w in $\widehat{M}$, we look at the intersection of its set S_w with the last two layers of the blocks Q_i, for all choices of i, and mark the intersection as interesting. From Lemma 10 it follows that each non-empty intersection corresponds to exactly one connected component of a graph $\widehat{G}_i$. In this way we can classify, for all $i \in [t]$, whether the connected components of $\widehat{G}_i$ are interesting or boring. This can be done in $\mathcal{O}(kn)$ time, overall.

Claim 13. Let $C^i_1, \dots, C^i_\ell$ be the connected components of $\widehat{G}_i$ that contain an interesting vertex. Then $\ell \leq \mu_i$. Here, μ_i is as defined in Claim 12.

Proof. We start with the observation that for every vertex v in $\widehat{M}_i$ the set S_v is a connected set in $G - \bigcup_{x \leq s_i+\beta-3} L_x$ and thus by Lemma 10, $S_v \cap V(\widehat{G}_i)$ is connected. Thus, in particular the set $S_v \cap V(\widehat{G}_i)$ is connected for the vertices v that have been classified interesting. Hence for each C^i_j there exists a unique vertex $v^i_j \in \widehat{M}_i$ such that v^i_j is classified interesting and $C^i_j = S_{v^i_j} \cap V(\widehat{G}_i)$. This implies that C^i_j is interesting if and only if $S_{v^i_j}$ is interesting. Hence, by Claim 12, $\ell \leq \mu_i$. ♢

For every block Q_i of the thinning set, we will define a vertex set $Z_{i,2}$ corresponding to it. The set Z_2 will simply be the union of the sets $Z_{i,2}$ over all relevant i. Consider the radial graph R_G of the plane graph G, and let $F_0, L_0, F_1, \cdots$ be the layers corresponding to breadth-first search starting at the vertex that represents the outer face in R_G. Here, the F_i's and L_i's denote the layers corresponding to faces and vertices, respectively. Let us consider the graph

$$R^i_G := R_G\left[\left(\bigcup_{y\in\{s_i,s_i+1,\ldots,s_i+\beta-1\}} L_y\right)\cup\left(\bigcup_{y\in\{s_i+1,\ldots,s_i+\beta-1\}} F_y\right)\right].$$

We start with the following claim.

Claim 14. Let $I := \{s_i+5,\ldots,s_i+\beta-6\}$. For every vertex $v\in L_y$, where $y =\in I$, we have that $\mathbf{R}^3_G(v) = \mathbf{R}^3_{G[V(Q_i)]}(v)$.

Proof. The proof of this claim follows from the fact that $\mathbf{R}^3_G(v)\subseteq V(Q_i)$ and thus $\mathbf{R}^3_G(v)\cap (V(G)-V(Q_i)) = \emptyset$. Now by Observation 1 we get that $\mathbf{R}^3_G(v) = \mathbf{R}^3_{G-(V-V(Q_i))}(v) = \mathbf{R}^3_{G[V(Q_i)]}(v)$. This concludes the proof. $\diamondsuit$

Now consider $\widehat{G}_i$. Let $C^i_1,\ldots,C^i_\ell$ be the connected components of $\widehat{G}_i$ that were classified as interesting. For each $x\in\ell$, we want to find shortest path P'^i_x in R^i_G between a vertex in layer L_{s_i} and a vertex in the interesting component C^i_x. We find all such paths simultaneously, by performing BFS in the graph R^i_G, where we initialize the BFS queue to contain every vertex in L_{s_i} with a distance value of zero. For each interesting component, the first of its vertices that is discovered by the BFS gives a shortest path, by tracing back over predecessors. From the paths P'^i_x in the graph R^i_G, we get a *shortest* radial path P^i_x in the graph G between a vertex in layer L_{s_i} and a vertex in C^i_x, by omitting all the vertices on P'^i_x that correspond to faces.

Let $\mathcal{T}_i$ be the union of these paths. That is, $\mathcal{T}_i = \bigcup_{x\in[\ell]} P^i_x$. Now take the radial ball $\mathbf{R}^3_G(\mathcal{T}_i)$ and define $Z_{i,2}$ as $U_i - \mathbf{R}^3_G(\mathcal{T}_i)$, i.e., all vertices in the innermost subblock U_i that have radial distance more than three to all vertices on the found paths. Let $Z_2 := \bigcup_{i=1}^{\ell} Z_{i,2}$. To find Z_2 efficiently, we first compute the union of the trees T_i, and afterward perform a single BFS in the radial graph R_G of G. We mark each vertex in G by its minimum radial distance to a member of $\bigcup_{i\in[t]}\mathcal{T}_i$. The vertices contained in the layers of $\bigcup_{i\in[t]} U_i$, whose minimum radial distance to the union of trees exceeds three, form the desired set Z_2. Hence these steps can be implemented in $\mathcal{O}(k^{\mathcal{O}(1)}n)$ time. Since the subblocks U_i are separated from other subblocks U_j by at least ten layers, this process indeed identifies Z_2. We let $Z := Z_1\cup Z_2$. This completes the identification process of Z.

Finding separating cycles around each vertex of Z. Now we show that we can find the desired separating cycles around every vertex in Z.

Claim 15. For each vertex $v\in Z$ there are two vertex-disjoint cycles $C_1, C_2\subseteq V(G)\setminus(\mathbf{R}^7_G(W)\cup Z)$ whose vertices are covered by A, such that in graph G, (C_1) separates v from $\mathbf{R}^7_G(W)$, and $V(C_2)$ separates v from $V(C_2)$.

Proof. We first show the desired claim for vertices in Z_1. Let z be an arbitrary vertex in Z_1. This implies that there exists a vertex $v_z\in G^*\cap\widehat{M}$ such that $z\in S_{v_z}$, and v_z has been classified as boring. Let v_p denote the parent of v_z in the tree G^*. Let i be the integer such that $v_p\in\widehat{R}_i$. Delete the edge between v_p and v_z from G^* and let $G^*_{v_z}$ and $G^*_{v_p}$ denote the subtrees of G^* that contain v_z and v_p respectively. Let $X_1 = \bigcup_{u\in V(G^*_{v_p})} S_u$ and $X_2 = \bigcup_{u\in V(G^*_{v_z})} S_u$. Observe that X_i, $i\in\{1,2\}$, is a connected set in G. Furthermore, observe that since v_z is boring we have that every vertex in $G^*_{v_z}$ is classified boring and thus $\mathbf{R}^7_G(W)\cap X_2=\emptyset$. Let C be the connected component of $X_2 - V(Q_i)$ that contains the vertex v. Notice that $\mathbf{R}^2_G(X_1)\cap X_2\subseteq V(Q_i)$ and thus $\mathbf{R}^2_G(X_1)\cap C=\emptyset$ and thus $d_G(X_1,C)\geq 3$. Thus, by Proposition 3, there are two cycles C_1 and C_2 such that all vertices on C_x, $x \in \{1,2\}$, have radial distance exactly x from X_1, and such that $V(C_x)$ is a X_1C-separator in G. The cycles C_x, $x \in \{1,2\}$, satisfy $V(C_x) \subseteq X_2 \cap \mathbf{R}^2_G(X_1) \subseteq V(Q_i)$ and hence C_1 and C_2 are covered by A. Both C_1 and C_2 separate z from $\mathbf{R}^7_G(W)$. If C_2 does not separate z from C_1 then the connectivity of C_1 and C_2 implies that C_1 separates z from C_1 and we may swap the names of C_1 and C_2. This completes the proof for the vertices in Z_1.

Now we show the desired claim for the vertices in Z_2. Let $z \in V(U_i) \cap Z_2$. Recall that by $C^i_1, \dots, C^i_\ell$, we denote the interesting components of $\widehat{G}_i = G[\bigcup_{x \in \{s_i+\beta-2, s_i+\beta-1\}} L_x]$ and by $\mathcal{T}_i$ we denote the union of shortest radial paths P^i_x between L_{s_i} and C^i_x. We know that for each C^i_j there exists a unique vertex $v^i_j \in \widehat{M}_i$ such that v^i_j is classified interesting and $C^i_j = S_{v^i_j} \cap V(\widehat{G}_i)$. Let $G^*_{v^i_j}$ denote the subtree of G^* rooted at v^i_j and $S_{\geq v^i_j} = \bigcup_{x \in V(G^*_{v^i_j})} S_x$. Clearly, $S_{\geq v^i_j}$ is connected. Consider the set

$$X_1 = \left(\bigcup_{x < s_i} L_x \right) \cup \mathcal{T}_i \cup \left(\bigcup_{j=1}^{\ell} S_{\geq v^i_j} \right).$$

Let $N = \mathbf{R}^1_G(X_1)$. We claim that the set N is connected. To see this, note that every pair of consecutive vertices on a path P^i_x, lie on a common face. In $\mathbf{R}^1_G(X_1)$, all the vertices on this face are included, and hence N is indeed a connected set in G. Observe that $\mathbf{R}^2_G(N) = \mathbf{R}^3_G(X_1)$. Since $z \notin \mathbf{R}^2_G(N)$ it follows that $d_G(N, \{z\}) \geq 3)$. Hence, by Proposition 3 there are two disjoint cycles C_1 and C_2 that both separate z from N, and such that all vertices on a cycle C_q have radial distance exactly exactly q from N in G, for $q \in \{1,2\}$. If $V(C_2)$ does not separate z from $V(C_1)$, then $V(C_1)$ separates z from $V(C_2)$ and we may swap their names. Since $C_q \subseteq V(G) \setminus N$ for $q \in \{1,2\}$, it follows that C_1 and C_2 are covered by A. This completes the proof. $\diamondsuit$

Tree decomposition of $G - Z$. As the final step we show how to find the desired tree decomposition of $G-Z$ in time $\mathcal{O}(kn)$. Our idea is as follows. We will show that for every $i \in [t]$, there exists a separator $D_i \in V(Q_i)$ of size $\mathcal{O}(k)$ such that there is no edge between vertices in layers L_a and L_b with $a < s_i$ and $b \geq s_i + \beta$. This would imply that the plane graph between "D_i and D_{i+1}" is $\mathcal{O}(k)$-outerplanar and thus has treewidth at most $\mathcal{O}(k)$, by Proposition 8. We find a tree decomposition for each such piece and finally get a tree decomposition for $G - Z$ by adding D_i's appropriately.

Let us fix an index $i \in [t]$ and let us consider the graph $G[V(Q_i)] - Z$. Now we add a vertex s and make it adjacent to every vertex in L_{s_i}. Recall that $C^i_1, \dots, C^i_\ell$ denote the interesting components of $\widehat{G} = G[\bigcup_{x \in \{s_j+\beta-2, s_j+\beta-1\}} L_x]$. Claims 12 and 13 together imply that $\ell \leq |W| \leq k$. Now we contract every interesting component C^i_j to t_j. We call the graph resulting from addition of s and contraction of connected components as H_i. Recall that P^i_x denote a shortest radial path between a vertex in the layer L_{s_i} and C^i_x. Let the path starting at s and then followed by P^i_x be denoted by P^i_{sx}. Observe that every path $P^i_{st_i}$ intersects every layer of $Q_i - \{s_i + \beta - 1\}$. Recall that every vertex in $U_i - \mathbf{R}^3_G(\mathcal{T}_i)$ is in Z_2. Here, $\mathcal{T}_i$ is the union of the paths P^i_x. Observe that the length of every path is equal. Let this length be q.

Claim 16. Let $T = \{t_1, \dots, t_\ell\}$ and for all $t \in T$, let P^i_{st} denote the radial path between s and t described above. Let $X = \bigcup_{t \in T} \mathbf{R}^7_G(w_t)$. Here, w_t is the $\lceil q/2 \rceil^{th}$ vertex of the path P^i_{st}. If $q \geq 34$, then there is no path from s to any vertex in T in $H_i - X$.

Proof. By Claim 14 we know that

$$X = \bigcup_{t \in T} \mathbf{R}^7_G(w_t) = \bigcup_{t \in T} \mathbf{R}^7_{H_i}(w_t).$$

Let $U = \bigcup_{t\in T} P^i_{st}$ and $P^i_{st_i} := s = y_{i1}, \dots, y_{iq} = t_i$. Suppose there is a path $P' := s = x_1 \cdots x_\gamma = t$ in $H_i - X$ from s to some vertex $t \in T$. Observe that P intersects every layer of $Q_i - \{s_i+\beta-1\}$. Let P be the subpath of P' that only contain vertices of $V(U_i)$ and intersects every layer in U_i. Now for every vertex x_i in P, we associate a vertex y^i of U such that it has shortest radial distance to x_i among the vertices of the set U (of course $x_i \in \mathbf{R}^3_G(y^i)$). For all $i \in [\ell]$, define $\text{Left}^i = \{s = y_{i1}, \dots, y_{i\lceil q/2\rceil-8}\}$ and $\text{Right}^i = \{y_{i\lceil q/2\rceil+8}, \dots, y_{iq}\}$. Let $\mathsf{Left} = \bigcup_{i\in[\ell]} \text{Left}^i$ and $\mathsf{Right} = \bigcup_{i\in[\ell]} \text{Right}^i$. Clearly, every vertex y^i associated with a vertex on P either belongs to Left or Right. Observe that for a vertex x_j of P that lies in the layer L_{s_i+8} the y^j lies in Left. This follows from the fact that y_i must belong to L_δ where $\delta \in \{s_i+5, \dots, s_i+11\}$ and every vertex in Right belongs to layer L_y where $y \geq s_i + 12$. Similarly, we can show that the last vertex of P that belongs to the last layer of U_i is associated with a vertex in Right. Let x_i and x_{i+1} be two consecutive vertices on P such that $y^i \in \mathsf{Left}$ and $y^{i+1} \in \mathsf{Right}$. Let $y^i \in \text{Left}^a$ and $y^{i+1} \in \text{Right}^b$. Now consider the radial path P' consisting of subpath $P^i_{st_a}[s, y^i]$ between s and y^i, a shortest radial path between y^i and x_i, the edge x_i and x_{i+1}, a shortest radial path between x_{i+1} and y^{i+1} and the subpath $P^i_{st_b}[y^{i+1}, t_b]$ between y^{i+1} and t_b. The number of vertices in P' is at most

$$|P^i_{st_a}[s,y^i]|+d+1+d+|P^i_{st_b}[y^{i+1},t_b]| \leq |\text{Left}^a|+2d+1+|\text{Right}^b| = |\text{Left}^b|+2d+1+|\text{Right}^b| < |P^i_{st_b}|.$$

This contradicts the fact that $P^i_{st_b}$ is a shortest radial path between s and t_b in R^i_G. This completes the proof. ♢

Now we describe the desired tree decomposition. Toward this we first find the separators D_i as follows. Since $\beta = 35$, we have that $q \geq 34$ and thus by Claim 16, we know that there exists a set of vertices $v_1, \dots, v_\ell$ such that $X = \bigcup_{i=1}^{\ell} \mathbf{R}^3_G(v_i)$ separates s from each of the terminals, that is, in $H_i - X$ there is no path from s to any t_i. We contract the vertices in L_{s_i} and L_{s_i+1} to s in H_i and obtain H_i^*. Clearly, X separates T from s even in H^*. However, by Claim 14 we have that

$$X = \bigcup_{t\in T} \mathbf{R}^7_G(w_t) = \bigcup_{t\in T} \mathbf{R}^7_{H_i}(w_t) = \bigcup_{t\in T} \mathbf{R}^7_{H_i^*}(w_t).$$

Now by applying Lemma 14 on H_i^* we know that there exists a vertex set D_i of size at most $21\ell \leq 21k$ such that in $H_i^* - D_i$there is no path from s to any t_i. To compute the desired D_i we do as follows. Add a vertex t to H_i^* and make it adjacent to $\{t_1, \dots, t_\ell\}$ and then contract $\{t_1, \dots, t_\ell\}$ to t. Now using the Ford-Fulkerson algorithm for finding max-flow min-cut, we can compute the desired D_i in time $\mathcal{O}(k^2|V(Q_i)|)$. The running time follows from the fact that we only need to run the flow augmentation at most $\mathcal{O}(k)$ times. Thus we can find all the desired D_i's in time $\mathcal{O}(k^{\mathcal{O}(1)}n)$. Let Γ_i' denotes the component of H_i^* that contains the vertex s after deletion of D_i and $\Gamma_i = (\Gamma_i' \cup L_{s_i} \cup L_{s_i+1}) - \{s\}$. Let Υ_i' be the union of all the components that do not contain s after deletion of D_i. Let $\Upsilon_i = (\Upsilon_i' \bigcup_{j=1}^{p} C^i_j) - \{t\}$.

Let $G_Z = G - Z$ and define $\mathcal{M}_i$ as follows.

$$\mathcal{M}_i = \begin{cases} \bigcup_{x<s_i} L_x \cup \Gamma_1 & \text{if } i = 0 \text{ ,} \\ \Upsilon_i \bigcup_{s_i+\beta \leq x < s_{i+1}} L_x \bigcup \Gamma_{i+1} & \text{if } 1 \leq i \leq t-1 \text{ ,} \\ \Upsilon_i \bigcup_{x \geq s_i+\beta} L_x & \text{if } i = t. \end{cases}$$

Notice that for every pair of indices i and j, $\mathcal{M}_i$ and $\mathcal{M}_j$ are pairwise vertex disjoint and there is no edge between a vertex in $\mathcal{M}_i$ and a vertex in $\mathcal{M}_j$. It is clear that each $\mathcal{M}_i$ is an $\mathcal{O}(k)$-outerplanar graph and thus by Proposition 8 there exists a tree decomposition $(T_i, \mathcal{X}_i)$. Now we obtain a tree decomposition of G_Z as follows. Let $\xi_0 = D_1$, $\xi_i = D_i \cup D_{i+1}$, $i \in [t-1]$, and $\xi_t = D_t$. Now we add ξ_i to every bag of the tree decomposition $(T_i, \mathcal{X}_i)$. Let $P = w_0, \cdots, w_t$ be a path on $t+1$ vertices and with a vertex w_i associate a bag ξ_i. Finally obtain a tree T by making w_i on the path P adjacent to a vertex in T_i. Thus, we get a tree decomposition $(T, \mathcal{X})$ of G_Z of width $\mathcal{O}(k)$. This completes the proof. $\square$

5.5.3 The Separation Lemma

In this section we prove the separation lemma that was used in the proof of Lemma 13.

Lemma 14. *Let G be a plane graph and $s, t_1, \ldots, t_p$ be a subset of vertices called terminals. Furthermore, let $v_1, \ldots, v_p$ be the set of vertices such that $X = \bigcup_{i=1}^{p} \mathbf{R}_G^d(v_i)$ separates s from each of the terminals, that is, in $G-X$ there is no path from s to any t_i. If $X \cap \{s, t_1, \ldots, t_p\} = \emptyset$ then there exists a vertex set D of size at most $\theta = (2d+1)3p$ such that in $G - D$ there is no path from s to any t_i.*

Proof. We prove the statement using contradiction. Suppose there is no set D of size at most θ such that it intersects all the paths from s to $T = \{t_1, \ldots, t_p\}$. Thus, by Menger's theorem there exists a family of paths $\mathcal{P} = \{P_1, \ldots, P_\gamma\}$ from s to T that are pairwise internally vertex disjoint and intersect T only at one of the endpoint. Here, $\gamma > \theta$. Let G' be the plane subgraph of G by induced by paths in $\mathcal{P}$. That is, G' has only those vertices and edges that appear on some path in $\mathcal{P}$. Now, we obtain another plane graph H from G' by thinking of a path between s and some t_i as a curve between s and t_i and thus an edge between s and t_i. Since the paths in $\mathcal{P}$ are internally vertex disjoint, this results in a plane bipartite multigraph H that inherits its embedding from G'. Here, bipartitions consist of s and T.

From H we make an auxiliary graph P_H as follows. Here, we have vertices for paths and we make two paths adjacent if they share a face of size two in H. Next we bound the number of connected components in P_H.

Claim 17. The number of connected components in P_H is at most $2(p-1)$.

Proof. Make a set J by selecting an arbitrary vertex (a path) from every connected component of P_H. Notice that if a connected component consists of a single vertex then clearly it belongs to J. Now delete all the edges from H that do not correspond to a path in J. Let the set of edges deleted be denoted by $\{e_1, \ldots, e_\ell\}$ and the resulting graph be H^*. Observe that H^* does not have face of length three as H is a bipartite graph. Next we claim that every face in H^* has length at least 4.

Toward this for all $i \in [\ell]$, define $E_i = \{e_1, \ldots, e_i\}$ and $H_i = H - E_i$. Clearly, $H^* = H_\ell$ and let $H_0 = H$. We delete edges from H one at a time in the order $e_1, \ldots, e_\ell$ and at every step maintain an invariant, that for any connected component C in P_{H_i} there exists a connected component C' in $P_{H_{i-1}}$ such that $C \subseteq C'$. Let us see how this invariance implies that H^* has no face of size two. Suppose it does, then it implies that P_{H_ℓ} has a connected component C containing the vertices corresponding to the paths that enclose this face of size two. But then by backward transitivity there is a connected component of P_H containing these vertices. However, in H^* there is exactly one path (vertex in P_H) from each connected component of P_H, a contradiction. So it only remains to prove that the invariance is maintained. To prove this we show that if there is an edge between P_a and P_b in P_{H_i} then they belong to the same connected component in $P_{H_{i-1}}$. If P_a and P_b share a face of size two in H_i that is also a two sized face in H_{i-1} then clearly they belong to the same connected component in $P_{H_{i-1}}$. So assume that P_a and P_b share a face of size two in H_i but they do not share a face of size two in H_{i-1}. This implies that the new face of size two enclosed by P_a and P_b in H_i appears after deleting the edge e_i corresponding to a path P_c. This implies that P_a and P_c share a face of size two in H_{i-1} and P_b and P_c share a face of size two in H_{i-1}. Thus P_a and P_b belong to the same connected component in $P_{H_{i-1}}$.

Let f denotes the number of faces in H^*. Now by applying Euler's formula and the fact that every face has size at least 4, the number of edges in H^* is given by

$$2f \leq |E(H^*)| \leq (p+1) + f - 2.$$

This implies that the number of faces is upper bounded by $p-1$ and thus the number of edges in H^* is upper bounded by $2(p-1)$. Since the number of connected components in P_H and the number of edges in H^* are in bijective correspondence, we have that the number of connected components in P_H is at most $2(p-1)$. This completes the proof. $\diamondsuit$

Now let us look at each connected component C of P_H. The vertices in C corresponds to edges in H between s and a terminal vertex $t \in T$. Each edge in C corresponds to a face of size two in H. Now we make the following simple observation.

Observation 4. There is at most one face of size two corresponding to an edge in C that has a terminal vertex in its interior.

Now for every $t \in T$, we divide the paths between s and t into blocks of size $2d+1$. If a block B consists of paths $P_1^B, P_2^B, \ldots, P_{2d+1}^B$ then it has the property that every pair P_i^B and P_{i+1}^B, $i \in [2d]$, shares a face of size two and their interior does not contain any terminals. Let $B_1, \ldots, B_\ell$ denote these blocks. By Observation 4 we know that at most one face of size two corresponding to an edge in C has a terminal vertex in its interior and hence for any i and j we have that the interiors of blocks B_i and B_j are disjoint. Now we claim that for any B_i its interior must contain a vertex from $\{v_1, \ldots, v_p\}$. Once we show this we will be done. Indeed, since the interiors of the blocks are pairwise disjoint we have that $\ell \leq p$. For each component of P_H at most $2d+1$ of its paths do not appear in any block and thus the number of paths is upper bounded by $\theta = (2d+1)p + (2d+1)2p = (2d+1)3p$. This contradicts that $\gamma > \theta$. So to conclude the proof of the lemma we show that for any B_i its interior must contain a vertex from $\{v_1, \ldots, v_p\}$. Recall that, $X = \bigcup_{i=1}^{p} \mathbf{R}_G^d(v_i)$ separates s from each of the terminals. Thus, for each block B_i, there is a vertex $v \in \{v_1, \ldots, v_p\}$ and a vertex w on the path $P_{d+1}^{B_i}$ such that $w \in \mathbf{R}_G^d(v)$. So if there is no vertex from $\{v_1, \ldots, v_p\}$ in the interior of B_i then the shortest radial path, say L, from v to w must intersect every path either in $\mathcal{P}_1 = \{P_1^{B_i}, \ldots, P_d^{B_i}\}$ or in $\mathcal{P}_2 = \{P_{d+2}^{B_i}, \ldots, P_{2d+1}^{B_i}\}$. However, by the assertion of the lemma $\mathbf{R}_G^d(v) \cap \{s, t_1, \ldots, t_p\} = \emptyset$ and thus L must intersect either $\mathcal{P}_1$ or $\mathcal{P}_2$ on at least d vertices (as paths in $\mathcal{P}_1$ or $\mathcal{P}_2$ are internally vertex disjoint). This contradicts that $w \in \mathbf{R}_G^d(v)$ and concludes the proof of the lemma. $\square$

5.5.4 Proof of Irrelevance

We now prove that vertices that are surrounded by two vertex-disjoint, nested, planarizing cycles, are irrelevant to the planarization question for apex sets that leave those two cycles in tact. We first prove the result for the irrelevance of a single vertex (Lemma 15), and then use this in an induction step to prove the simultaneous irrelevance of a set of vertices (Lemma 16).

Lemma 15. *Let G be a graph and v be a vertex in G. If G contains two vertex-disjoint, nested, v-planarizing cycles C_1, C_2, then a set $S \subseteq V(G) \setminus (V(C_1) \cup V(C_2) \cup \{v\})$ is an apex set in G if and only if S is an apex set in $G - \{v\}$.*

Proof. The "only if" direction is trivial, so assume S is an apex set in $G - v$ of size at most k. We prove that S is an apex set in G.

Consider the graph $G - S$ and the cycle C_1 in it. By Lemma 1 the graph $G - S$ is planar if and only if the overlap graph $O(G - S, C_1)$ is bipartite and all C_1-bridges B in $G - S$ yield planar graphs $C_1 \cup B$. Let us first establish the latter requirement. For any C_1-bridge B that does not contain v, the graph $C_1 \cup B$ is a subgraph of $G - S - v$ and therefore planar. For the bridge B_v containing v, we find that $C_1 \cup B_v$ is a subgraph of $G^{C_1}(v)$, the connected component of C_1 that contains v together with C_1. Since C_1 is a v-planarizing cycle it follows that $G^{C_1}(v)$ is planar, and hence $C_1 \cup B_v$ is planar.

Now focus on the overlap graph. As the cycles C_1 and C_2 are nested, vertex v has no neighbors on C_1 and therefore adding v to a C_1-bridge cannot change the attachment points of that bridge. Therefore, there are two ways in which the overlap graph $O(G - S - v, C_1)$ can change by adding v. The vertex v might become a new C_1-bridge, or might cause multiple existing bridges to be merged into one. If v becomes a new C_1-bridge by itself, then $N_G(v) \subseteq S$ which shows that v is an isolated vertex in $G - S$. As adding an isolated vertex to the planar graph $G - S - v$ preserves planarity, $G - S$ is planar in this case. In the remainder we may assume there is at least one neighbor of v that survives in $G - S - v$. Adding v might connect multiple distinct C_1-bridges together, effectively merging them in the overlap graph. We prove that the overlap graph remains bipartite by showing that whenever two bridges are merged, at least one bridge has no attachments on C_1; hence whenever a pair of vertices of the overlap graph is merged, at least one of them has no incident edges. It is easy to see that such merge operations preserve bipartiteness.

So assume for a contradiction that there are distinct C_1-bridges B, B' in $G - S - v$, which are merged together by the addition of v because $N_G(v) \setminus S$ contains at least one vertex in each bridge, and such that both bridges have at least one attachment on C_1. Let $b \in V(B) \cap N_G(v) \setminus S$, and $b' \in V(B') \cap N_G(v) \setminus S$. As the separators are nested, the vertex set of C_2 separates v from C_1 in G, and hence in $G - S$. Any bridge that has an attachment on C_1 contains a vertex adjacent to C_1. Therefore, any bridge that contains a neighbor of v and that also has attachments on C_1, intersects C_2. But as C_2 is not intersected by S and forms a cycle, all vertices which have a path to a vertex on C_2 in $G - S - v$ are actually part of the same connected component of $G - S - V(C_1)$. So the bridges B and B' cannot be distinct; contradiction. Hence building $O(G - S, C_1)$ from $O(G - S - v, C_1)$ corresponds to repeatedly merging isolating vertices into other vertices, which does not create odd cycles in the graph. Therefore $O(G - S, C_1)$ is bipartite which proves by Lemma 1 that $G - S$ is planar. □

Lemma 16. *Let G be a graph, and let D and Z be disjoint vertex subsets such that for every $z \in Z$, graph G contains two vertex-disjoint, nested, z-planarizing cycles $C_1, C_2 \subseteq D$. Then a set $S \subseteq V(G) \setminus (D \cup Z)$ is an apex set in G if and only if S is an apex set in $G - Z$.*

Proof. Let G, D, Z and S be as in the lemma statement. We use induction on $|Z|$. The base case $Z = \emptyset$ is trivial. For the induction step, let $z \in Z$ and let $Z' := Z \setminus \{z\}$. Clearly, if $(G - Z) - S$ is nonplanar, then its supergraph $G - S$ is nonplanar. For the other direction we will use Lemma 15. We may apply induction to the graph $G - z$ with the sets D and Z', to conclude that S is an apex set in $G - z$. Observe that as $z \notin D$ since $Z \cap D = \emptyset$, the nested z'-planarizing cycles for $z' \in Z'$ through D that exist in G by assumption, still exist in $G - z$. Hence we are justified in invoking the induction hypothesis to this setting. Given that S is an apex set in $G - z$, we prove that S is an apex set in G. By the preconditions to the lemma, there are two vertex-disjoint, nested, z-planarizing cycles $C_1, C_2 \subseteq V(D)$ in G. Since $Z \cap D = \emptyset$, these cycles do not contain z. We may therefore apply Lemma 15 to the graph G, using z as the vertex v in the lemma, and S as the apex set, to conclude that $G - S$ is planar. This concludes the proof. □

5.6 Finding a Tree Decomposition

In this section we combine the tools developed so far, to reduce an instance of Disjoint Vertex Planarization to a series of $k + 1$ low-treewidth instances of the following problem.

	RESTRICTED DISJOINT VERTEX PLANARIZATION
Input:	A graph G, an integer k, a set of undeletable vertices $D \subseteq V(G)$, and an apex set $S \subseteq D$ of G of size at most $6k$.
Parameter:	k.
Output:	An apex set for G of size at most k that is disjoint from D, or NO if such a set does not exist.

Lemma 17. *There is an $\mathcal{O}(k^{\mathcal{O}(1)}n)$-time algorithm that, given a graph G, an integer k, and a $6k$-apex set S of G, either decides that G has no k-apex set disjoint from S, or produces $k+1$ instances (G'_j, k, D'_j, S) ($j \in [k+1]$) of* RESTRICTED DISJOINT VERTEX PLANARIZATION, *along with tree decompositions $(T_j, \mathcal{X}_j)$ of G_j of width $\mathcal{O}(k)$, such that:*

1. *Each graph G'_j is a minor of G.*

2. *If G has a k-apex set disjoint from S, then at least one graph G'_j has a k-apex set disjoint from $D'_j \cup S$.*

3. *For each $j \in [k+1]$, any apex set in G'_j that does not intersect D'_j, is also an apex set in G.*

Proof. The algorithm combines the pieces developed so far. Given the input (G, k, S), we proceed as follows.

Preparing for the branching stage. We first apply Lemma 7 to (G, k, S). In $\mathcal{O}(kn)$ time it produces a minor G' of G without contracting edges incident on S, and a Tutte decomposition $(F', \mathcal{X}')$ of G' whose leaves intersect $N_{G'}(S)$, such that any apex set in G' is also an apex set in G. As we do not contract edges incident on S, Proposition 1 shows that if G has a k-apex set disjoint from S, then G' does as well.

We then apply Lemma 8 to the graph G' with the set S and the Tutte decomposition $(F', \mathcal{X}' = \{X'_i \mid i \in V(F)\})$. If it concludes that G' has no k-apex set disjoint from S, then clearly G does not have one either, and we output NO. Otherwise we obtain for each $i \in V(F)$ a set W_i of size at most $4k$, such that the radial 7-balls around W_i in $\mathrm{Torso}(G' - S, X'_i)$ contain all neighbors of S in X'_i, and all adhesion vertices of X'_i. We then process each node $i \in V(F)$ of the Tutte decomposition individually, performing the following steps.

By Definition 1 the graph $\mathrm{Torso}(G' - S, X'_i)$ is triconnected. As the torsos of a Tutte decomposition are minors of the original graph by Proposition 13, the torso is in fact a planar triconnected graph. We compute a plane embedding in $\mathcal{O}(|X'_i|)$ time using one of the known linear-time algorithms. Then we compute the outerplanarity layers of the embedding by BFS in the radial graph, to which we apply Lemma 9 with $\beta := 35$. It partitions the indices of the outerplanarity layers of $\mathrm{Torso}(G' - S, X'_i)$ into sets $S_{i,0}, \ldots, S_{i,k+1}$ such that the layers corresponding to a group $S_{i,j}$ with $j \in [k+1]$ form a (β, λ) thinning set for $\mathrm{Torso}(G' - S, X'_i)$, with $\beta = 35$ and $\lambda := \beta^2(13|W_i| + k + 4) \leq 70000k$. The lemma guarantees that all adhesion vertices in X'_i, as well as all vertices in $\mathbf{R}^7_{\mathrm{Torso}(G'-S,X'_i)}(W_i)$ are contained in layers whose index is in $S_{i,0}$.

The branching stage. For each $j \in \{1, \ldots, k+1\}$ we create a small-treewidth instance of RESTRICTED DISJOINT VERTEX PLANARIZATION, as follows. For each choice of j, we apply Lemma 13 to each node $i \in V(F)$ of the Tutte decomposition of $G - S$. We use $\mathrm{Torso}(G' - S, X'_i)$ as the planar triconnected graph, the set W'_i as the set from which the irrelevant vertices have to be separated, and $S_{i,j}$ as the thinning set. The outerplanarity layers are computed by radial BFS, and also given to the algorithm. The output is a set $Z_{i,j}$ of irrelevant vertices for each node i, along with a tree decomposition $(T_{i,j}, \mathcal{X}_{i,j})$ of the graph $\text{Torso}((G' - S) - Z_{i,j}, \mathcal{X}'_i)$ of width $\mathcal{O}(k)$. The lemma guarantees that for each vertex $z \in Z_{i,j}$, there are two cycles C_1, C_2 in $\text{Torso}((G' - S), \mathcal{X}'_i) - Z_{i,j}$, whose vertices are covered by the thinning set $S_{i,j}$, and which separate z from $R^7_{\text{Torso}(G'-S,X'_i)}(W_i)$. This fact will be used later when proving correctness of the reduction.

After applying Lemma 13 to all nodes of F, for a particular choice of j, we combine the tree decompositions of the torsos together into a tree decomposition of $(G' - S) - \bigcup_{i \in V(F)} Z_{i,j}$. We derive the structure of the decomposition tree for the graph $(G' - S) - \bigcup_{i \in V(F)} Z_{i,j}$, from the structure of the Tutte decomposition. Basically, we want to replace every node i in the Tutte decomposition, by a tree decomposition of $\text{Torso}(G' - S, \mathcal{X}'_i) - Z_{i,j}$. As the node i might have neighbors in the Tutte decomposition forest F, we need to make sure that when replacing i by a decomposition of its torso, a connection is made to a tree decomposition of the neighboring torso, such that the connectivity requirement of tree decompositions is satisfied. Effectively, this comes down to the following. When replacing a node i in the Tutte decomposition by the low-width tree decomposition $(T_{i,j}, \mathcal{X}_{i,j})$ of its reduced graph, then for each neighbor $j \in N_F(i)$ we need to reconnect j to a bag of $T_{i,j}$ that contains $X'_i \cap X'_j$, i.e., the adhesion between the two bags. By the definition of Tutte decomposition, each pair of bags has an intersection of size at most two, and if the intersection indeed has size two then the edge between the two adhesion vertices is part of the torso, by Proposition 12. Hence the tree decomposition $(T_{i,j}, \mathcal{X}_{i,j})$ is guaranteed to have a bag containing the adhesion. All have to do, to merge the decompositions of the torsos into a decomposition for the entire reduced graph, is to find such attachment points efficiently. This can be done as follows. For each graph $\text{Torso}(G' - S, X'_i) - Z_{i,j}$, we will add a pointer from each edge, to a bag in $(T_{i,j}, \mathcal{X}_{i,j})$ containing both endpoints of the bag. First we compute a constant out-degree orientation of the torso in linear time. Afterward we can scan through each bag $x \in V(T_{i,j})$, and for each pair of vertices in the bag we test in constant time whether there is an edge between them. If an edge exists, we add a pointer in the edge record to bag x. As the bag size is $\mathcal{O}(k)$, this takes $\mathcal{O}(k^2|X'_i|)$ time per bag, for a total of $\mathcal{O}(k^2 n)$ by Proposition 10. Since each edge is realized in a bag we hereby identify, for each edge, a bag where both its endpoints appear.

Using this information, and the fact that edges of the decomposition tree are labeled by the corresponding adhesion points, it is straight-forward to connect the tree decompositions of the torsos into one big tree decomposition for $(G' - S) - \bigcup_{i \in V(F)} Z_{i,j}$. As all edges of that graph are contained in one of the torsos, it is obvious that all edges are represented afterward. The width does not increase. By our connection scheme, we maintain the connectivity property of tree decompositions. Afterward we may add S to all bags, increasing the width by at most $6k$. Let $G'_j := G' - \bigcup_{i \in V(F)} Z_{i,j}$. The process we just described results in a tree decomposition of G'_j of width $\mathcal{O}(k)$, for each $j \in [k+1]$, in time $\mathcal{O}(k^{\mathcal{O}(1)} n)$ in total. Let $V(S_{i,j})$ be the vertices in $\text{Torso}(G' - S, X'_i)$ that are covered by the thinning set $S_{i,j}$, and let $D'_j := \bigcup_{i \in V(F')} (V(S_{i,j}) \setminus Z_{i,j})$, i.e., the vertices covered by the thinning set that are not marked as irrelevant.

Claim 18. For each $j \in [k+1]$, for each $i \in V(F')$, for each vertex $z \in Z_{i,j}$, the graph G' contains two vertex-disjoint, nested, z-planarizing cycles $C_1, C_2 \subseteq D'_j$.

Proof. Consider a vertex $z \in Z_{i,j}$. As $Z_{i,j}$ was computed by Lemma 13, applied to $\text{Torso}(G' - S, X'_i)$ with thinning set $S_{i,j}$ and W'_i as the vertices to be avoided, the lemma guarantees the existence of two vertex-disjoint cycles $C_1, C_2 \subseteq X'_i \setminus (\mathbf{R}_{\text{Torso}}(G' - S, X'_i)^3(W_i) \cup Z_{i,j})$, whose vertices are covered by the thinning set $S_{i,j}$, such that $V(C_1)$ separates z from $\mathbf{R}^3_{\text{Torso}(G'-S,X'_i)}(W_i)$, and $V(C_2)$ separates z from C_1. Now observe that by construction of the set W_i, all neighbors of S in $\text{Torso}(G' - S)$ are contained in $\mathbf{R}^3_{\text{Torso}(G'-S,X'_i)}(W_i)$. Similarly, all adhesion points in the bag X'_i are contained in this set. So in graph G', the set $V(C_1)$ actually separates z from all neighbors in X'_1 of the apices S, and from all adhesion points of the decomposition. So the connected component of $G' - V(C_1)$ that contains z, does not contain $N_G(S) \cap X'_i$, nor any adhesion vertices. To prove that C_1 is z-planarizing, we need to show that $G'^{C_1}(z)$ is planar. Recall that $G'^{C_1}(z)$ is the subgraph of G' induced by the following set of vertices: $V(C_1)$ together with the vertices in the connected component of $G' - V(C_1)$ that contains z. But as $V(C_1)$ does not contain any adhesion points of the decomposition, nor any neighbors of S, and separates z from all those vertices, we find that $G'^{C_1}(z)$ is a subgraph of $\text{Torso}(G' - S, X'_i)$, and therefore planar. Hence C_1 is a v-planarizing cycle in G'. As C_2 separates v from C_1 in $\text{Torso}(G' - S, X'_i)$, and the torso contains all the edges of $G'[X'_i]$, the component of $G' - V(C_2)$ that contains z, is a subgraph of the component of $G' - V(C_1)$ that contains v; it is therefore also planar. So C_2 is also a v-planarizing cycle. As the vertex sets of C_1 and C_2 are covered by the thinning set $S_{i,j}$, and are disjoint from $Z_{i,j}$, the nested cycles are contained in D'_j, by definition of the latter set. ♢

Claim 19. For any $j \in [k+1]$, any k-apex set in G'_j that is disjoint from $D'_j \cup S$, is also an apex set in G.

Proof. Let S' be a k-apex set in G'_j that is disjoint from $D'_j \cup S$. Let $Z_j := \bigcup_{i \in V(F)} Z_{i,j}$. Recall that $G'_j = G' - \bigcup_{i \in V(F)} Z_{i,j} = G' - Z_j$. As each $z \in Z_j$ is contained in some set $Z_{i,j}$, by Claim 18 the graph G' has two vertex-disjoint, nested, z-planarizing cycles $C_1, C_2 \subseteq D'_j$. Hence we may apply Lemma 16 to the graph G', the set D'_j, the irrelevant vertices Z_j, and the apex set S', to conclude that $G' - S'$ is planar. As the trimming step of the Tutte decomposition ensured that any apex set in G', is also an apex set in G, we find that S' is an apex set in G. ♢

Claim 20. If G' has a k-apex set S' disjoint from S, then there is an index $j^* \in [k+1]$ such that S' is an apex set in G'_{j^*} that is disjoint from D'_{j^*}.

Proof. Assume that G' has a k-apex set S' disjoint from S. For $j \in \{1, \ldots, k+1\}$, define the vertex set U_j as follows: $U_j := \bigcup_{i \in V(F)} \bigcup_{\sigma \in S_{i,j}} L_{i,\sigma}$. In other words, U_j contains, from each triconnected component, the vertices in its outerplanarity layers whose index is contained in $S_{i,j}$. Now observe that the sets U_j and $U_{j'}$ are disjoint for $j \neq j'$. This follows from the fact that the only vertices that occur in more than one bag of the Tutte decomposition are adhesion vertices, while all adhesion vertices in X'_i belong to layers whose indices are in $S_{i,0}$. Since the sets U_j are pairwise vertex-disjoint, and S' contains at most k vertices, there is at least one index $j^* \in [k+1]$ such that $U_{j^*} \cap S' = \emptyset$. As D'_{j^*} contains the vertices of U_{j^*} that were not removed due to irrelevance, the set D'_{j^*} is a subset of U_{j^*} and therefore disjoint from S'. Since G_{j^*} is an induced subgraph of G', it follows that $G_{j^*} - S'$ is a subgraph of $G' - S'$ and therefore planar. Hence S' is an apex set for j^* disjoint from D'_{j^*}. ♢

Armed with these claims, we complete the proof of Lemma 17. The output consists of the series of instances (G'_j, k, D'_j, S) of Restricted Disjoint Vertex Planarization, along with the width-$\mathcal{O}(k)$ tree decompositions of the graphs. Since the reduction steps we have applied to obtain the graphs G'_j have been minor operations (to trim the Tutte decomposition) and vertex deletions (to reduce the treewidth), each output graph is a minor of the input, showing that condition (1) of the lemma statement is satisfied. By Claim 20, if G' has a k-apex set S' disjoint from S, then it is a k-apex set in some graph G'_{j^*}, and S' is disjoint from D'_{j^*}. As $S' \cap S = \emptyset$ by assumption, such a set S' is disjoint from $D'_{j^*} \cup S$, showing that condition (2) of the lemma statement is satisfied. Finally, Claim 19 shows that (3) is satisfied. As these steps can be implemented in $\mathcal{O}(k^{\mathcal{O}(1)} n)$ time, this concludes the proof of Lemma 17. □

6 Planarization on Graphs of Bounded Treewidth

In this section we show how to solve the following problem by dynamic programming.

	Weighted Vertex Planarization on Bounded Treewidth Graphs
Input:	A graph G with a tree decomposition of width w. A non-negative integer cost c_v for each vertex v of G
Parameter:	w.
Output:	A minimum-cost apex set in G.

The section will be devoted to the proof of the following theorem.

Theorem 3. *There is a $2^{\mathcal{O}(w \log w)} \cdot n$ algorithm for* Weighted Vertex Planarization on Bounded Treewidth Graphs.

Toward the proof of Theorem 3 we will assume that the given tree decomposition is a nice tree decomposition, otherwise we make it nice in linear time. For a given nice tree decomposition $(T, \mathcal{X})$ of G we define the following objects. For a node $v \in V(T)$, T_v denotes the subtree of T rooted at v. We define V_v to be $V_v = \bigcup_{u \in V(T_v)} B_u$ and $G_v = G[V_v]$. For every node v of T we will now define a signature function f_v. The family $\{f_v : v \in V(T)\}$ of functions will form the backbone of the dynamic programming algorithm for Weighted Vertex Planarization on Bounded Treewidth Graphs.

The function f_v takes as input a triple (X, P, F) where $X \subseteq B_v$, P is a plane multigraph with vertex set $B_v \setminus X$ such that $G[B_v] \subseteq P$, and F is a subset of the faces of P. The function outputs a non-negative integer. Specifically, $f_v(X, P, F)$ returns the cost of the minimum cost set $S \subseteq V_v$ such that $S \cap B_v = X$ and there is a plane embedding of $G_v \setminus S$ such that

1. every vertex of B_v and edge of $G[B_v]$ is embedded in the same place as in the embedding of P,
2. every other vertex and edge of $G_v \setminus S$ is embedded in $\bigcup_{f \in F} f$.

We will call an embedding of $G_v \setminus S$ that satisfies the above properties a *good embedding for* (X, P, F). We are interested in computing the value of $f_r(\emptyset, P, \{p\})$ where r is the root node of t, P is the empty graph and p is the face covering the entire plane, since this returns the size of the smallest set S such that $G_r \setminus S = G \setminus S$ can be embedded in the plane. The idea is to compute the function $f_v(X, P, F)$ for every node v in T and every triple (X, P, F) in a bottom up fashion, starting with the leaves. However there are infinitely many plane multigraphs P with vertex set $B_v \setminus X$, and so we can not really compute and store the value of $f_v(X, P, F)$ for every choice of (X, P, F). To overcome this issue we will only consider a carefully chosen subset of the triples (X, P, F). To that end we will devise some "reduction rules" that will allow us to focus only on triples for which these rules may not be applied. In the following lemmata, let v be a node of the tree decomposition, (X, P, F) be a triple where $X \subseteq B_v$, P be a plane multigraph with vertex set $B_v \setminus X$ such that $G[B_v] \subseteq P$, and F be a subset of the faces of P.

Lemma 18. *Let $e \notin E(G)$ be an edge of P incident to faces f_1 and f_2 of P such that $f_1 \notin F$ and $f_2 \notin F$. Then $f_v(X, P, F) = f_v(X, P \setminus e, F)$.*

Proof. Since $e \notin E(G)$ and $f_1 \notin F$, $f_2 \notin F$ we have that any set S and plane embedding of $G_v \setminus S$ good for (X, P, F) is good for $(X, P \setminus \{e\}, F)$ and vice versa. □

For the next lemma we will need the following simple fact on plane embeddings.

Fact 1. *Let G be a planar graph, $\hat{G}$ be a plane embedding of G, ζ be a noose of $\hat{G}$ with interior I and touching the vertices S of $\hat{G}$. Let ζ' be a closed curve in I touching all vertices in S. Then there exists a plane embedding $\hat{G}'$ of G coinciding with $\hat{G}$ on S and outside I and embedding all vertices that $\hat{G}$ embeds in I in the interior of ζ'.*

Here a noose of a plane graph is a simple closed curve that intersects the embedding of the graph only in its vertices. Fact 1 follows from the fact that the interiors of ζ and ζ' are homotopic.

Lemma 19. *Let $e \notin E(G)$ be an edge of P incident to faces $f_1 \notin F$ and $f_2 \in F$ of P such that e is incident to all vertices f_1 is incident to. Then $f_v(X, P, F) = f_v(X, P \setminus e, (F \setminus \{f_1, f_2\}) \cup \{f_1 \cup f_2)\})$.*

Proof. For any set S a plane embedding of $G_v \setminus S$ which is good for (X, P, F) is also good for $(X, P \setminus e, (F \setminus \{f_1, f_2\}) \cup \{f_1 \cup f_2)\})$. For the $\leq$ part of the equality let Z be the set of vertices of P incident to f_1. Consider a set S and plane embedding of $G_v \setminus S$ good for $(X, P \setminus e, (F \setminus \{f_1, f_2\}) \cup \{f_1 \cup f_2)\})$. Applying Fact 1 with the noose ζ being the boundary of $f_1 \cup f_2$ and ζ' being the boundary of f_2 yields an embedding where all vertices and edges that were embedded in $f_1 \cup f_2$ are embedded in f_2 instead. This new embedding of $G_v \setminus S$ is good for (X, P, F). □

Lemma 20. *Let $f_1 \in F$ be a face of P incident to at most two vertices, $f_2 \in F$ be incident to the same vertices as f_1. Then $f_v(X, P, F) = f_v(X, P, F \setminus f_1)$.*

Proof. For any set S a plane embedding of $G_v \setminus S$ which is good for $(X, P, F \setminus f_1)$ is also good for (X, P, F). For the $\geq$ part of the equality consider a set S and plane embedding of $G_v \setminus S$ good for (X, P, F). Let Z be the vertices embedded inside f_1 in this embedding. Since f_2 is incident to at most two vertices there is a face $f_2' \subseteq f_2$ in this embedding (of $G_v \setminus S$) that is incident to the same vertices of P as f_2 is. Since f_2 is incident to the same vertices of P as f_1 is, f_2' is incident to the same vertices of P as f_1 and hence we may change the embedding to embed Z in f_2' rather than in f_1. This yields a plane embedding of $G_v \setminus S$ good for $(X, P, F \setminus f_1)$. □

We will say that a triple (X, P, F) is *reduced* if none of Lemmata 18, 19 and 20 can be applied. Given a triple (X, P, F) it is easy to check in $(|X| + |E(P)|)^{O(1)}$ time whether the conditions of Lemmata 18, 19 or 20 hold. If they do one may change the triple (X, P, F) to a triple (X', P', F') such that $f_v(X, P, F) = f_v(X, P', F')$ and $|E(P')| < |E(P)|$ or $|E(P')| = |E(P)|$ and $|F'| < |F|$. This yields the following lemma.

Lemma 21. *There is an algorithm that given a node v of the tree decomposition and triple (X, P, F) with $X \subseteq B_v$, P a plane multigraph with vertex set $B_v \setminus X$ such that $G[B_v] \subseteq P$, and F a subset of the faces of P, outputs in $(|X| + |E(P)|)^{O(1)}$ time a reduced triple (X, P', F') with $f_v(X, P, F) = f_v(X, P', F')$.*

Finally we need to bound the number of edges and faces of P in reduced triples.

Lemma 22. *In a reduced triple (X, P, F). $|E(P)| \leq 19w + 6$ and the number of faces of P is at most $18w + 8$.*

Proof. Consider the face-vertex incidence graph, and apply Lemma 2 with the vertex cover being $V(P)$. By Lemma 2 there exists a collection $\mathcal{C}$ of subsets of $V(P)$ such that $|\mathcal{C}| \leq 6|V(P)| + 1$ and for every face f of P the set Z_f of vertices incident to f is in $\mathcal{C}$. Let $\mathcal{C} = \mathcal{C}_{\geq 3} \cup \mathcal{C}_{\leq 2}$ where $\mathcal{C}_{\geq 3}$ contains all sets in $\mathcal{C}$ of size at least 3 and $\mathcal{C}_{\leq 2}$ contains all the sets in $\mathcal{C}$ of size at most 2. There are at most $2|\mathcal{C}_{\geq 3}|$ faces of P that are incident to at least 3 vertices, since otherwise there are at least three faces incident to the same three vertices, contradicting planarity of the face-vertex incidence graph of P.

We now prove that the number of faces of P incident to at most two vertices is at most $|\mathcal{C}_{\leq 2}|+6|V(P)|-12$. Suppose not. Since G is a simple planar graph, $|E(G)\cap E(P)| \leq 3|V(P)|-6$ and thus at most $6|V(P)|-12$ faces of P are incident to an edge of $E(G)$. Hence there are two faces f and f' that are incident to at most two vertices, are incident to exactly the same vertices, and are not incident to any edges of G. If $f \notin F$ let e be any edge incident to f and all vertices that f is incident to. Such an e exists since f is incident to at most two vertices. Let f_2 be the face other than f incident to e. If $f_2 \notin F$ then Lemma 18 applies, contradicting that (X, P, F) is reduced. If $f_2 \in F$ then Lemma 19 applies, contradicting that (X, P, F) is reduced. Hence $f \in P$. An identical argument shows that $f' \in P$. But then Lemma 20 applies to f and f' contradicting that (X, P, F) is reduced.

Thus the total number of faces in P is at most

$$2|\mathcal{C}_{\geq 3}| + |\mathcal{C}_{\leq 2}| + 6|V(P)| - 12 \leq 2|\mathcal{C}| + 6|V(P)| - 12 \leq 18|V(P)| - 10 \leq 18w + 8.$$

For each connected component of P one may now use Euler's formula to upper bound the number of edges. This yields the desired upper bound $|E(P)| \leq 19w + 6$. □

Proposition 15. *For any constant c there are $w^{O(w)}$ distinct plane multigraphs on at most w vertices and at most cw edges. Furthermore there is an algorithm that lists all plane multigraphs on at most w vertices and at most cw edges in time $w^{O(w)}$.*

Proof. Let P be a plane multigraph on at most w vertices and at most cw edges. Subdividing every edge of P twice yields a plane simple graph P' on $(2c+1)w$ vertices and $3cw$ edges, and P can be recovered from P' in *poly*(w) time. We can list all plane graphs on at most $3cw$ edges spending at most $O(w^{O(1)})$ time per graph using an algorithm of Yamanaka and Nakano [36]. For each listed plane graph P' we check in *poly*(w) time whether it is the result of taking a plane multigraph P and subdividing every edge twice. If it is we output P. All that remains to show is that the number of simple plane graphs on at most $(2c+1)w$ vertices and $3cw$ edges is bounded by $w^{O(w)}$.

Consider a plane graph P at most $(2c+1)w$ vertices and $3cw$ edges. By adding edges inside the faces of P we can obtain a plane triangulation P' of P. Since P' is a simple plane triangulation it has a unique embedding [32] and exactly $(6c+3)w-6$ edges. Thus P is uniquely determined by the *graph* P' (without the embedding) and the subset $E(P') \setminus E(P)$ of edges. There are $w^{O(w)}$ graphs on $(2c+1)w$ vertices and $(6c+3)w-6$ edges and $2^{O(w)}$ choices for $E(P') \setminus E(P)$ completing the proof. □

Proposition together with, Lemma 22 immediately implies the following corollary.

Corollary 1. *For any node v in the decomposition there are at most $w^{O(w)}$ reduced triples (X, P, F).*

We are now ready to give the proof of the main theorem of this section.

Proof of Theorem 3. Our algorithm for Weighted Vertex Planarization on Bounded Treewidth Graphs will compute $f_v(X, P, F)$ for every node v of the decomposition and every reduced triple (X, P, F), starting from the leaves. The algorithm will employ some recurrences on the value of $f_v(X, P, F)$. In these recurrences it will some times be necessary to compute the value of $f_u(X', P', F')$ where u is a child of v and (X', P', F') is *not reduced.* In this case we apply Lemma 21 to compute a reduced triple (X'', P'', F'') such that $f_u(X', P', F') = f_u(X'', P'', F'')$ and compute $f_u(X'', P'', F'')$ instead of computing $f_u(X', P', F')$.

For every leaf node i of T we have that $V_i = B_i = \emptyset$ and so there are only two possible inputs for f_i, namely $(\emptyset, P, \emptyset)$ and $(\emptyset, P, \{p\})$ where P is the empty graph and p is the face covering the entire plane. In both cases f_i returns 0.

We now describe recurrences that show how to compute the function f_u at a node u from the function(s) for the children of u. Suppose u is a join node and let v and w be the children of u. We will show that

$$f_u(X, P, F) = \min_{\substack{P' \supseteq P, \\ F_1, F_2}} f_v(X, P', F_1) + f_w(X, P', F_2) - \sum_{x \in X} c_x \tag{1}$$

In Equation 1 the minimum is taken over all plane multigraphs P' on the same vertex set as P such that $P \subseteq P'$, $|E(P')| \leq |E(P)| + 16w + 4$, all the edges of $E(P') \setminus E(P)$ are embedded in $\bigcup_{f \in F} f$, and over all subsets F_1 and F_2 of the faces of P' such that $F_1 \cap F_2 = \emptyset$ and $(\bigcup_{f \in F_1} f) \cup (\bigcup_{f \in F_2} f) \subseteq \bigcup_{f \in F} f$.

We will now describe a recurrence for the introduce nodes. To that end we need to first describe an operation on pairs (P, F) where P is a plane graph and F is a subset of the faces of P. Let x be a vertex in P. Then $(P, F)/x$ is a pair $(P_{/x}, F_{/x})$ where $P_{/x}$ is a plane graph and $F_{/x}$ is a subset of the faces of $P_{/x}$ obtained from (P, F) as follows. Let $y_1, \ldots y_d$ be the neighbors of x listed in clockwise order in the embedding of P, and set $y_{d+1} = y_1$. For every i from 1 to d add an edge from y_i to y_{i+1}, drawing the edges along the embedding of edges $y_i x$ and $x y_{i+1}$. Now, remove the vertex x from the graph P to obtain the plane graph $P_{/x}$. Let f be the face in $P_{/x}$ which is enclosed by the cycle $y_1, \ldots, y_d, y_1$. For each face $g \in F$ not incident to x, g is also a face of $P_{/x}$, add this face to $F_{/x}$. For each face $g \in F$ incident to x, let $g' = g \setminus f$. Now g' is a face of $P_{/x}$, add g' to $F_{/x}$. This pair $(P_{/x}, F_{/x})$ is $(P, F)/x$.

For an introduce node u with child v, we will show that the following recurrence holds. Let x be the vertex in $B_u \setminus B_v$. Then,

$$f_u(X, P, F) = \begin{cases} c_x + f_v(X \setminus x, P, F) & \text{if } x \in X, \\ f(X, P_{/x}, F_{/x}) & \text{if } x \notin X. \end{cases} \tag{2}$$

In Equation 2 the pair $P_{/x}, F_{/x}$ is equal to $(P, F)/x$.

Consider now a forget node u with child v, such that $B_v \setminus B_u = \{x\}$. We will show that the following recurrence holds.

$$f_u(X, P, F) = \min \left[f_v(X \cup \{x\}, P, F), \min_{P' \supseteq P} f_v(X, P', F') \right] \tag{3}$$

In Equation 3 the minimum is taken over all plane multigraphs P' such that $V(P') = V(P) \cup \{x\}$, $E(P') = E(P) \cup \{xy \ : \ y \in V(P) \textit{ and } xy \in E(G)\}$, the embedding of P and P' coincide on all vertices and edges they share, x is embedded in a face $f \in F$ and F' contains the set of faces of P' such that $\bigcup_{f \in F} f = \bigcup_{f \in F'} f$.

Assuming Equations 1, 2 and 3 we can now give an algorithm for Weighted Vertex Planarization on Bounded Treewidth Graphs. The algorithm computes the function f_u for all possible arguments for leaves u, and marks the leaves as processed. Then the algorithm selects a lowermost unprocessed node u in T and computes the value of the function f_u for all reduced triples (X, P, F) using recurrences 1, 2 or 3, according to whether u is a join, introduce or forget node respectively. To compute $f_u(X, P, F)$ using these recurrences the algorithm will need to compute the value of $f_v(X', P', F')$ for a child v of u and a triple (X', P', F') which is not necessarily reduced. Each time this happens the algorithm computes using Lemma 21 a reduced triple (X'', P'', F'') such that $f_v(X', P', F') = f_v(X'', P'', F'')$ and looks up the value of $f_v(X'', P'', F'')$ in the table computed at v. Since (X'', P'', F'') is reduced and v already has been processed when u is considered this entry of the dynamic programming table has already been filled at this time. Now the algorithm marks u as processed and repeats this step for the next lowermost unprocessed node. When the root r has been processed in this way the algorithm outputs $f_r(\emptyset, P, \{p\})$ where r is the root node of t, P is the empty graph and p is the face covering the entire plane. Correctness follows from the correctness of Equations 1, 2 and 3. We now proceed to the running time analysis.

For each node u in the tree we have to compute f_u for all reduced triples (X, P, F). By Corollary 1 the number of reduced triples is at most $w^{O(w)}$. For each choice of the arguments we compute f_u using Equation 1, 2 or 3 depending on what kind of node u is. In Equation 1 we go over all choices of P', F_1 and F_2, where (X, P, F) is a reduced triple and $|E(P')| \leq |E(P)| + 16w + 4 \leq O(w)$. Thus P' is a plane multigraph on at most $w + 1$ vertices and $O(w)$ edges, and hence there are $2^{O(w \log w)}$ choices for P' and $2^{O(w)}$ choices for F_1 and F_2. For each choice of P', F_1 and F_2 the algorithm needs to compute $f_v(X, P', F_1)$ and $f_w(X, P', F_2)$ where (X, P', F_1) are not necessarily reduced triples. This is done by applying Lemma 21 to each of these triples, which takes time $O(w^{O(1)})$, and then performing constant time table lookups. Thus at each join node u, f_u is computed for all choices of arguments in time $2^{O(w \log w)}$.

Equation 2 can be applied in constant time if the function f_v for the child v has been precomputed, possibly incurring an additional $O(w^{O(1)})$ time overhead for applying Lemma 21. In Equation 3 the minimum is taken over plane multigraphs P' on at most $|E(P)| + w + 1$ edges. Since (X, P, F) is a reduced triple Lemma 22 implies that $|E(P')| = O(w)$ and so there are at most $2^{O(w \log w)}$ choices for P'. For each choice of P' one may have to apply Lemma 21 to compute the value of $f_v(X, P', F')$, incurring a $O(w^{O(1)})$ overhead. Thus at each node u, f_u is computed for all choices of arguments in time $2^{O(w \log w)}$. Finally there are $O(n)$ nodes in the decomposition tree T, and so the total running time becomes $2^{O(w \log w)} \cdot n$. The proof of Equations 1, 2 and 3 is postponed until the end of this section.

The algorithm described here was for the version of Weighted Vertex Planarization on Bounded Treewidth Graphs where we only need to output the size of the smallest set S such that $G \setminus S$ is planar, rather than outputting such a set. To make the algorithm above output the solution instead we can use standard backtracking following the recurrences. □

To conclude the proof of Theorem 3, we need to show correctness of Equations 1, 2 and 3. We start with Equation 1.

Lemma 23. *For a join node u with children v and w it holds that:*

$$f_u(X, P, F) = \min_{\substack{P' \supseteq P, \\ F_1, F_2}} f_v(X, P', F_1) + f_w(X, P', F_2) - \sum_{x \in X} c_x$$

Here the minimum is taken over all plane multigraphs P' on the same vertex set as P such that $P \subseteq P'$, $|E(P')| \leq |E(P)| + 16w + 4$, all the edges of $E(P') \setminus E(P)$ are embedded in $\bigcup_{f \in F} f$, and over all subsets F_1 and F_2 of the faces of P' such that $F_1 \cap F_2 = \emptyset$ and $(\bigcup_{f \in F_1} f) \cup (\bigcup_{f \in F_2} f) \subseteq \bigcup_{f \in F} f$.

Proof. We first show the $\leq$ part of the equation. For any $P' \supseteq P$, F_1, F_2 as described in the equation, suppose there are a sets $S_v \subseteq V_v$, $S_w \subseteq V_w$ such that $S_v \cap B_v = X$, $S_w \cap B_w = X$ such that the following holds. (a) There is an embedding of $G_v \setminus S_v$ embedding the vertices and edges of P' identically as P' and all remaining vertices and edges of $G_v \setminus S_v$ in $\bigcup_{f \in F_1} f$. (b) There is an embedding of $G_w \setminus S_w$ embedding the vertices and edges of P' identically as P' and all remaining vertices and edges of $G_w \setminus S_w$ in $\bigcup_{f \in F_2} f$. The two embeddings of G_v and G_w do not overlap (except in the edges and vertices of P') and so can be drawn in the plane simultaneously. This yields an embedding of $G_u \setminus (S_v \cup S_w)$ embedding the vertices and edges of P identically as P and all other vertices and edges of G_u are embedded in $\bigcup_{f \in F} f$. Since

$S_v \cap S_w = X$ it follows that

$$\sum_{x \in S_v \cup S_w} c_x = \sum_{x \in S_v} c_x + \sum_{x \in S_w} c_x - \sum_{x \in X} c_x.$$

This concludes the proof for the $\leq$ part of the equation.

We now show the $\geq$ part of the equation. Suppose there is a set S such that $G_u \setminus S$ has a plane embedding which coincides with P on the vertices and edges of P, and the remaining vertices and edges are embedded into $\bigcup_{f \in F} f$. Set $S_v = S \cap V_v$ and $S_u = S \cap V_u$, and observe that $\sum_{x \in S} c_x = \sum_{x \in S_v} c_x + \sum_{x \in S_w} c_x - \sum_{x \in X} c_x$.

We make a plane graph P' and face set F' from P and F as follows. Initially, set $P' = P$ and $F' = F$. We will add some edges to P' in such a way that the new edges do not intersect the embedding of $G_u \setminus S$, yielding an embedding of the graph $G'_u \setminus S$, where $G'_u = (V(G_u), E(G_u) \cup E(P'))$, that coincides with P' on the vertices and edges of P' and embeds all other vertices and edges of $G'_u \setminus S$ in $\bigcup_{f \in F'} f$. As long as there is some face f in P' that contains at least one component C_v of $G_v \setminus (P \cup S)$ and at least one component C_w of $G_w \setminus (P \cup S)$, we can add an edge between two vertices in f (or a self-loop inside f), drawing the edge inside f such that this edge does not intersect the embedding of $G'_u \setminus S$ and splits f into two faces f_v and f_w with C_v embedded in f_v and C_w embedded in f_w. Remove f from F' as f is no longer a face of P', add f_v and f_w to F'. If f_v is incident to at most two vertices of P we also modify the embedding as follows. Let Z be the set of vertices of P incident to f_v. Simultaneously re-embed inside of f_v all the connected components of $G_v \setminus (P \cup S)$ whose neighborhood is a subset of $Z \cup S$. Since $|Z| \leq 2$ it is possible to do this without introducing any edge crossings. Similarly, if f_w is incident to at most two vertices of P we also modify the embedding as follows. Let Z be the set of vertices of P incident to f_w. Simultaneously re-embed inside of f_w all the connected components of $G_w \setminus (P \cup S)$ whose neighborhood is a subset of $Z \cup S$. Observe that the modified embedding of $G'_u \setminus S$ coincides with P' on the vertices and edges of P' and embeds all other vertices and edges of $G'_u \setminus S$ in $\bigcup_{f \in F'} f$.

The above process must terminate since the number of components of $G_u \setminus (P \cup S)$ is finite. We now argue that the number of edges added before the process terminates is at most $16|V(P')| - 12 \leq 16w + 4$. Each time the process adds an edge e to P' and does not create any new faces incident to at most two vertices, it creates at least two faces incident to at least three vertices, and destroys one such face. Since P' can have at most $4|V(P')|$ faces incident to at least 3 vertices each (see for example proof of Lemma 2), this can happen at most $4|V(P')|$ times. Each time the process adds an edge e to P' and creates a face f_v which is incident to a set Z on at most two vertices, it moves all components of $G_v \setminus (P \cup S)$ whose neighborhood is a subset of $Z \cup S$ into f_v. Similarly, each time the process adds an edge e to P' and creates a face f_w which is incident to a set Z on at most two vertices, it moves all components of $G_w \setminus (P \cup S)$ whose neighborhood is a subset of $Z \cup S$ into f_w. Hence, for any set Z on at most two vertices, the process creates at most two faces incident to exactly Z. By Lemma 2 applied to the face-vertex incidence graph of P' there are at most $6|V(P)| - 6$ distinct sets $Z \subseteq P$ such that there is a face of P' incident to exactly Z. Hence the process will create at most $12|V(P')| - 12$ new faces incident to at most two vertices. Thus the process must terminate after at most $16|V(P')| - 12 \leq 16w + 4$ steps.

Consider the graph P' and set F' when the process terminates. Each face $f \in F'$ either only contains vertices and edges from $G_v \setminus P$ or vertices and edges from $G_w \setminus P$ (or neither). Furthermore, each face $f \in F'$ is a subset of some face in F. Let F_1 contain all the faces of F' containing vertices of $G_v \setminus P$ and F_2 contain all other faces of F'. We now have an embedding of $G_v \setminus S_v$ coinciding with P' on all vertices of P' with all other vertices of $G_v \setminus S_v$ embedded in F_1. Similarly for we have an embedding of $G_w \setminus S_w$ coinciding with P' on all vertices of P' with all other vertices of $G_w \setminus S_w$ embedded in F_2. Since F_1 and F_2 are sets of disjoint faces of P' and $(\bigcup_{f \in F_1} f) \cup (\bigcup_{f \in F_2} f) \subseteq \bigcup_{f \in F} f$, this concludes the proof. □

Lemma 24. *For an introduce node u with child v, the following recurrence holds. Let x be the vertex in $B_u \setminus B_v$. Then,*

$$f_u(X,P,F) = \begin{cases} c_x + f_v(X \setminus x, P, F) & \text{if } x \in X, \\ f(X, P_{/x}, F_{/x}) & \text{if } x \notin X. \end{cases}$$

Here the pair $P_{/x}, F_{/x}$ is equal to $(P,F)/x$ as described in the proof of Theorem 3

Proof. We first handle the case that $x \in X$. Any embedding of $G_u \setminus S$ coinciding with P and embedding the remainder of $G_u \setminus S$ in $\bigcup_{f \in F} f$ is also such an embedding of $G_v \setminus (S \setminus \{x\})$. Similarly such an embedding of $G_v \setminus S$ is an embedding of $G_u \setminus (S \cup \{x\})$ with the same properties. Hence for the case that $x \in X$ the equation holds.

We now consider the case that $x \notin X$. For the $\leq$ direction of the equation consider an embedding of $G_v \setminus S$ coinciding with $P_{/x}$ on the vertices of $P_{/x}$ and embedding the remaining vertices in $\bigcup_{f \in F_{/x}} f$. We can now embed x in the face f of $P_{/x}$ which contains x in the embedding of P. Since this face is not in $F_{/x}$ we can embed x and all the edges to its neighbors without intersecting the embedding of $G_v \setminus S$. This yields an embedding of $G_u \setminus S$ that coincides with P on the vertices and edges of P and embeds the remaining vertices and edges of $G_u \setminus S$ in $\bigcup_{f \in F} f$.

For the $\geq$ direction of the equation consider an embedding of $G_u \setminus S$ that coincides with P on the vertices and edges of P and embeds the remaining vertices and edges of $G_u \setminus S$ in $\bigcup_{f \in F} f$. Removing x from this embedding and drawing the edges of $E(P_{/x}) \setminus E(P)$ along the edges of x yields an embedding of $G_u \setminus S$ coinciding with $P_{/x}$ on the vertices of $P_{/x}$ and embedding the remaining vertices in $\bigcup_{f \in F_{/x}} f$. ◻

Lemma 25. *Let u forget node with child v, such that $B_v \setminus B_u = \{x\}$. The following recurrence holds.*

$$f_u(X,P,F) = \min \left[f_v(X \cup \{x\}, P, F), \min_{P' \supset P} f_v(X, P', F') \right]$$

Here the minimum is taken over all plane multigraphs P' such that $V(P') = V(P) \cup \{x\}$, $E(P') = E(P) \cup \{xy \ : \ y \in V(P) \text{ and } xy \in E(G)\}$, the embedding of P and P' coincide on all vertices and edges they share, x is embedded in a face $f \in F$ and F' contains the set of faces of P' such that $\bigcup_{f \in F} f = \bigcup_{f \in F'} f$.

Proof. We start with the $\leq$ part of the equation. If there is a set S such that $S \cap B_v = X \cup \{x\}$ such that there is an embedding of $G_v \setminus S$ coinciding with P on all vertices and edges of P and mapping remaining vertices and edges to $\bigcup_{f \in F} f$, then this is also such an embedding of $G_u \setminus S$. Now, suppose there is an embedding of $G_v \setminus S$ coinciding with P' on all vertices and edges of P' and mapping remaining vertices and edges to $\bigcup_{f \in F'} f$ for some choice of P' and F' that the minimum goes over. This is an embedding of $G_u \setminus S$ coinciding with P on all vertices and edges of P and mapping remaining vertices and edges to $\bigcup_{f \in F} f$.

For the $\geq$ part, suppose there is a set S such that $S \cap B_u = X$ such that there is an embedding of $G_u \setminus S$ good for (X,P,F). If $x \in S$ then this embedding of $G_u \setminus S$ is also an embedding of $G_v \setminus S$ good for $(X \cup \{x\}, P, F)$. On the other hand, if $x \notin S$ then let

$$P' = (V(P) \cup \{x\}, E(P) \cup \{xy \ : \ y \in V(P) \text{ and } xy \in E(G)\}).$$

The embedding of $G_u \setminus S$ contains in it an embedding of P' mapping x to a face in F. Let F' be the set of faces in this embedding of P' such that $\bigcup_{f \in F} f = \bigcup_{f \in F'} f$. The embedding of $G_u \setminus S$ maps all remaining vertices and edges to $\bigcup_{f \in F'} f$ and hence it is also an embedding of $G_v \setminus S$ which is good for (X, P', F'). Furthermore $V(P') = V(P) \cup \{x\}$, $E(P') = E(P) \cup \{xy \ : \ y \in$

$V(P)$ *and* $xy \in E(G)\}$, the embedding of P and P' coincide on all vertices and edges they share, x is embedded in a face $f \in F$ and F' contains the set of faces of P' such that $\bigcup_{f \in F} f = \bigcup_{f \in F'} f$. Hence (X, P', F') is considered in the minimization, concluding the proof. $\square$

7 Complete Algorithm

Putting everything together, we establish the following theorem.

Theorem 4. k-Vertex Planarization *can be solved in* $2^{\mathcal{O}(k \log k)} \cdot n$ *time.*

Proof. Given as input a graph G and integer k our algorithm proceeds as follows. We first apply Lemma 4 to either answer the problem, or to find an induced subgraph G' of G with $\mathcal{O}(kn)$ edges, such that (i) any apex set S in G' of size at most k is also an apex set of G, and (ii) if G' is k-apex then any maximal matching in G' has size at least $\frac{|V(G')|}{30 \cdot 2^k (k+13)}$. We then greedily compute a maximal matching M in G', which takes $\mathcal{O}(kn)$ time by the bound on the edge count of G'. If $|M| < \frac{|V(G')|}{30 \cdot 2^k (k+13)}$ then, by the lemma, G' is not k-apex, so G is not k-apex either; we output no. Otherwise we contract the edges in M to obtain a minor G'/M of G'. Writing n' for the number of vertices in G', and n'_M for the number of vertices in G'/M, we find that $n'_M \leq (1 - \frac{1}{30 \cdot 2^k (k+13)})n'$. We then recurse on the instance $(G'/M, k)$ to find a minimum-size apex set S_M in G'/M. Since k-apex graphs are closed under taking minors, if G' has a k-apex set, then G'/M has one as well. If the recursive call results in the answer no, we may therefore safely output no. Otherwise, we uncontract the matching M, as follows. For each vertex v in S_M, if v is the result of a contraction then we put both endpoints of the contracted edge into S_1. Otherwise we just put v into S_1. We now have a set S_1 of size at most $2k$ such that $G - S_1$ is matching-contractible, since contracting the edges in $M - S_1$ results in a minor of the planar graph $(G/M - S_M)$.

Using Theorem 2 we in time $2^{\mathcal{O}(k \log k)} \cdot n$ either conclude that $G - S_1$ has no apex set of size at most k (in which case neither does G), or we find a set S_2 of size at most $4k$ in $G - S_1$ such that $G \setminus (S_1 \cup S_2)$ is planar. Let $S := S_1 \cup S_2$ be the resulting apex set of size at most $6k$. We then use Observation 3 to reduce to the Disjoint Vertex Planarization problem. For each subset $A \subseteq S$, we create an instance $(G - A, k - |A|, S \setminus A)$ of Disjoint Vertex Planarization. By the size bound on S, we produce at most 2^{6k} instances. Any solution to one of these instances of the disjoint problem, gives a k-apex set in G, when combined with the appropriate set A. Conversely, if G has a k-apex set, then at least one of the produced instances has a solution. Hence in the remainder it suffices to solve these $2^{\mathcal{O}(k)}$ instances of Disjoint Vertex Planarization, that each have a parameter value of at most k. Therefore, consider (G_A, k_A, S_A) to be an instance of the disjoint problem that was obtained for some subset $A \subseteq S$.

We apply Lemma 17 to each produced instance (G_A, k_A, S_A) in $\mathcal{O}(k^{\mathcal{O}(1)} n)$ time. We either find out that the instance has no solution, or produce $k+1$ instances $(G_{A,j}, k_A, D_{A,j}, S_A)$ of Restricted Disjoint Vertex Planarization, for $j \in [k+1]$, along with tree decompositions of width $\mathcal{O}(k)$ of the associated graphs. As the graphs are minors of G_A, their vertex and edge counts do not exceed those of the original input G. The lemma guarantees that if (G_A, k_A, S_A) has a solution, then at least one of the $k+1$ produced instances has a solution. Conversely, any solution to one of the $k+1$ produced instances, is a solution to (G_A, k_A, S_A).

To solve the $k+1$ instances of Restricted Disjoint Vertex Planarization, for each subset $A \subseteq S$, we assign to each vertex in $D_{A,j}$ a weight of $k_A + 1$, and to the other vertices a weight of one. We then apply the dynamic programming algorithm of Theorem 3 to the tree decomposition of width $\mathcal{O}(k)$, solving each instance in $2^{\mathcal{O}(k \log k)} \cdot n$ time. If we find a solution of weight at most k_A, then we trace back the series of equivalences and output the corresponding solution to G. If none of the $k+1$ instances for any of the $2^{|S|}$ guesses for A, result in a solution of weight at most k_A, then we output that G has no k-apex set. By tracing the series of equivalences throughout the proof, it is easy to verify that this approach is correct.

Finally, let us bound the time that is needed to execute this approach for an input (G, k) to k-Vertex Planarization. The running time of all steps of the algorithm except for the recursive step is $2^{\mathcal{O}(k \log k)} \cdot n$. Thus the total running time is governed by the recurrence $T(n,k) \leq T(n(1-\frac{1}{30 \cdot 2^k (k+13)}, k) + 2^{6k} \cdot (k+1) \cdot 2^{\mathcal{O}(k \log k)} \cdot n$ which solves to $T(n,k) \leq 2^{\mathcal{O}(k \log k)} \cdot n$. This concludes the proof. □

8 Conclusion

We have shown how the use of elementary techniques and structural insights yields a faster algorithm for k-Vertex Planarization. We expect that by changing the trimming step for the Tutte decomposition, and the dynamic programming scheme (or resorting to Courcelle's Theorem), the techniques introduced here can also be brought to bear on the related problems k-Edge Planarization and k-Planarization by Edge Contraction. While the running time of our algorithm is near-optimal, it remains an interesting challenge to determine whether k-Vertex Planarization can be solved in $2^{\mathcal{O}(k)}n$ time. We conclude with an open problem. Recently there has been much interest in the general k-$\mathcal{F}$-Deletion problem, which asks for a finite, fixed set of graphs whether an input graph can be made $\mathcal{F}$-minor-free using at most k vertex deletions. For every set $\mathcal{F}$ containing only connected graphs and at least one planar graph, k-$\mathcal{F}$-Deletion can be solved in $2^{\mathcal{O}(k)}n$ time by a randomized algorithm [16]. Kim et al. [26] gave an algorithm with running time $2^{\mathcal{O}(k)}n^2$ that works whenever $\mathcal{F}$ has a planar graph, without any connectivity requirements. The k-Vertex Planarization problem is a special case of k-$\mathcal{F}$-Deletion when $\mathcal{F} = \{K_{3,3}, K_5\}$. Observe that the cited algorithms do not apply to planarization, as the obstructions are nonplanar. Can the techniques presented here be used to solve more k-$\mathcal{F}$-Deletion problems in $2^{\text{poly}(k)}n$ time, for $\mathcal{F}$ without any planar graphs? The problem of finding a set of k vertices whose removal results in a graph that is embeddable on the torus, is likely to be the most accessible open case of k-$\mathcal{F}$-Deletion.

References

[1] J. Abello, P. Klavík, J. Kratochvíl, and T. Vyskocil. Matching and ℓ-subgraph contractibility to planar graphs. *CoRR*, abs/1204.6070, 2012.

[2] S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991.

[3] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41:153–180, 1994.

[4] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[5] H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998.

[6] R. B. Borie, R. G. Parker, and C. A. Tovey. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica*, 7(5&6):555–581, 1992.

[7] C. Chekuri and A. Sidiropoulos. Approximation algorithms for Euler genus and related problems. *CoRR*, abs/1304.2416, 2013. To appear in FOCS 2013.

[8] B. Courcelle. The monadic second-order logic of graphs I: Recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.

[9] B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theor. Comput. Sci.*, 109(1&2):49–82, 1993.

[10] G. Di Battista, P. Eades, R. Tamassia, and I. Tollis. *Graph drawing: algorithms for the visualization of graphs.* Prentice Hall, 1999.

[11] R. Diestel. *Graph Theory.* Springer-Verlag, Heidelberg, 4th edition, 2010.

[12] R. Diestel, K. Kawarabayashi, and P. Wollan. The Erdös-Pósa property for clique minors in highly connected graphs. *J. Comb. Theory, Ser. B*, 102(2):454–469, 2012.

[13] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000.

[14] M. R. Fellows and M. A. Langston. Nonconstructive tools for proving polynomial-time decidability. *J. ACM*, 35(3):727–739, 1988.

[15] M. R. Fellows and M. A. Langston. On search, decision, and the efficiency of polynomial-time algorithms. *Journal of Computer and System Sciences*, 49(3):769 – 779, 1994.

[16] F. V. Fomin, D. Lokshtanov, N. Misra, and S. Saurabh. Planar $\mathcal{F}$-Deletion: Approximation, kernelization and optimal FPT algorithms. In *Proc. 53rd FOCS*, pages 470–479, 2012.

[17] C. Gutwenger and P. Mutzel. A linear time implementation of SPQR-trees. In *Proc. 8th GD*, pages 77–90, 2000.

[18] J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2(3):135–158, 1973.

[19] J. E. Hopcroft and R. E. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.

[20] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[21] B. Jansen. Polynomial kernels for hard problems on disk graphs. In *Proc. 12th SWAT*, pages 310–321, 2010.

[22] F. Kammer and T. Tholey. Approximate tree decompositions of planar graphs in linear time. In *Proc. 23rd SODA*, pages 683–698, 2012.

[23] I. Katsikarelis. Computing bounded-width tree and branch decompositions of k-outerplanar graphs. *CoRR*, abs/1301.5896, 2013.

[24] K. Kawarabayashi. Planarity allowing few error vertices in linear time. In *Proc. 50th FOCS*, pages 639–648, 2009.

[25] K. Kawarabayashi, B. Mohar, and B. A. Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In *Proc. 49th FOCS*, pages 771–780. IEEE Computer Society, 2008.

[26] E. Kim, A. Langer, C. Paul, F. Reidl, P. Rossmanith, I. Sau, and S. Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. In *Proc. 40th ICALP*, pages 613–624, 2013.

[27] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980.

[28] D. Marx. What's next? Future directions in parameterized complexity. In H. L. Bodlaender, R. Downey, F. V. Fomin, and D. Marx, editors, *The Multivariate Algorithmic Revolution and Beyond*, volume 7370 of *Lecture Notes in Computer Science*, pages 469–496. Springer, 2012.

[29] D. Marx and I. Schlotter. Obtaining a planar graph by vertex deletion. In *Proc. 33rd WG*, pages 292–303, 2007.

[30] D. Marx and I. Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3-4):807–822, 2012.

[31] B. Mohar. Face covers and the genus problem for apex graphs. *J. Comb. Theory, Ser. B*, 82(1):102–117, 2001.

[32] B. Mohar and C. Thomassen. *Graphs on Surfaces.* Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2001.

[33] B. A. Reed. Rooted routing in the plane. *Discrete Appl. Math.*, 57(2-3):213–227, 1995.

[34] B. A. Reed, N. Robertson, A. Schrijver, and P. D. Seymour. Finding disjoint trees in planar graphs in linear time. In *Proc. Graph Structure Theory*, pages 295–302, 1991.

[35] M. Tedder, D. G. Corneil, M. Habib, and C. Paul. Simpler linear-time modular decomposition via recursive factorizing permutations. In *Proc. 35th ICALP*, pages 634–645, 2008.

[36] K. Yamanaka and S.-I. Nakano. Listing all plane graphs. *J. Graph Algorithms Appl.*, 13(1):5–18, 2009.