Minor-Closed Graph Classes with Bounded Layered Pathwidth

Vida Dujmović [‡] David Eppstein [†] Gwenaël Joret [§]
Pat Morin [*] David R. Wood [¶]

October 18, 2018

Abstract

We prove that a minor-closed class of graphs has bounded layered pathwidth if and only if some apex-forest is not in the class. This generalises a theorem of Robertson and Seymour, which says that a minor-closed class of graphs has bounded pathwidth if and only if some forest is not in the class.

1 Introduction

Pathwidth and treewidth are graph parameters that respectively measure how similar a given graph is to a path or a tree. These parameters are of fundamental importance in structural graph theory, especially in Roberston and Seymour's graph minors series. They also have numerous applications in algorithmic graph theory. Indeed, many NP-complete problems are solvable in polynomial time on graphs of bounded treewidth [20].

Recently, Dujmović, Morin, and Wood [16] introduced the notion of layered treewidth. Loosely speaking, a graph has bounded layered treewidth if it has a tree decomposition and a layering such that each bag of the tree decomposition contains a bounded number of vertices in each layer (defined formally below). This definition is interesting since several natural graph classes, such as planar graphs, that have unbounded treewidth have bounded layered treewidth. Bannister, Devanny, Dujmović, Eppstein, and Wood [1] introduced layered pathwidth, which is analogous to layered treewidth where the tree decomposition is required to be a path decomposition.

[‡]School of Computer Science and Electrical Engineering, University of Ottawa, Ottawa, Canada (vida.dujmovic@uottawa.ca). Research supported by NSERC and the Ontario Ministry of Research and Innovation.

[§]Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium (gjoret@ulb.ac.be). Supported by an ARC grant from the Wallonia-Brussels Federation of Belgium.

[†]Department of Computer Science, University of California, Irvine, California, USA (eppstein@uci.edu). Supported in part by NSF grants CCF-1618301 and CCF-1616248.

[*]School of Computer Science, Carleton University, Ottawa, Canada (morin@scs.carleton.ca). Research supported by NSERC.

[¶]School of Mathematical Sciences, Monash University, Melbourne, Australia (david.wood@monash.edu). Research supported by the Australian Research Council.

The purpose of this paper is to characterise the minor-closed graph classes with bounded layered pathwidth.

1.1 Definitions

Before continuing, we define the above notions. A *tree decomposition* of a graph G is a collection $(B_x \subseteq V(G) : x \in V(T))$ of subsets of $V(G)$ (called *bags*) indexed by the nodes of a tree T, such that:

(i) for every edge vw of G, some bag B_x contains both v and w, and

(ii) for every vertex v of G, the set $\{x \in V(T) : v \in B_x\}$ induces a non-empty connected subtree of T.

The *width* of a tree decomposition is the size of the largest bag minus 1. The *treewidth* of a graph G, denoted by $\mathrm{tw}(G)$, is the minimum width of a tree decomposition of G.

A *path decomposition* is a tree decomposition in which the underlying tree is a path. We denote a path decomposition by the corresponding sequence of bags $(B_1, \dots, B_n)$. The *pathwidth* of G, denoted by $\mathrm{pw}(G)$, is the minimum width of a path decomposition of G.

A graph H is a *minor* of a graph G if a graph isomorphic to H can be obtained from a subgraph of G by contracting edges. A class of graphs $\mathcal{G}$ is *minor-closed* if for every $G \in \mathcal{G}$, every minor of G is in $\mathcal{G}$.

A *layering* of a graph G is a partition $(V_0, V_1, \dots, V_t)$ of $V(G)$ such that for every edge $vw \in E(G)$, if $v \in V_i$ and $w \in V_j$ then $|i - j| \leqslant 1$. Each set V_i is called a *layer*. For example, for a vertex r of a connected graph G, if V_i is the set of vertices at distance i from r, then $(V_0, V_1, \dots)$ is a layering of G, called the *bfs layering* of G starting from r.

Dujmović et al. [16] introduced the following definition. The *layered width* of a tree decomposition $(B_x : x \in V(T))$ of a graph G is the minimum integer ℓ such that, for some layering $(V_0, V_1, \dots, V_t)$ of G, each bag B_x contains at most ℓ vertices in each layer V_i. The *layered treewidth* of a graph G, denoted by $\mathrm{ltw}(G)$, is the minimum layered width of a tree decomposition of G. Bannister et al. [1] defined the *layered pathwidth* of a graph G, denoted by $\mathrm{lpw}(G)$, to be the minimum layered width of a path decomposition of G.

1.2 Examples and Applications

Several interesting graph classes have bounded layered treewidth (despite having unbounded treewidth). For example, Dujmović et al. [16] proved that every planar graph has layered treewidth at most 3, and more generally that every graph with Euler genus g has layered treewidth at most $2g + 3$. Note that layered treewidth and layered pathwidth are not minor-closed parameters (unlike treewidth and pathwidth). In fact, several graph classes that contain arbitrarily large clique minors have bounded layered treewidth or bounded layered pathwidth. For example, every graph that can be drawn on a surface of Euler genus g with at most k crossings per edge has layered treewidth at most $2(2g + 3)(k + 1)$. Even with $g = 0$ and

$k = 1$, this family includes graphs with arbitrarily large clique minors. Map graphs have similar behaviour [14].

Bannister et al. [1] identified the following natural graph classes that have bounded layered pathwidth (despite having unbounded pathwidth): every squaregraph has layered pathwidth 1; every bipartite outerplanar graph has layered pathwidth 1; every outerplanar graph has layered pathwidth at most 2; every Halin graph has layered pathwidth at most 2; and every unit disc graph with clique number k has layered pathwidth at most $4k$.

Part of the motivation for studying graphs with bounded layered treewidth or pathwidth is that such graphs have several desirable properties. For example, Norin proved that every n-vertex graph with layered treewidth k has treewidth less than $2\sqrt{kn}$ (see [16]). This leads to a very simple proof of the Lipton-Tarjan separator theorem. A standard trick leads to an upper bound of $11\sqrt{kn}$ on the pathwidth (see [14]).

Another application is to stack layouts (or book embeddings), queue layouts and track layouts. Dujmović et al. [16] proved that every n-vertex graph with layered treewidth k has track- and queue-number $O(k \log n)$. This leads to the best known bounds on the track- and queue-number of several natural graph classes. For graphs with bounded layered pathwidth, the dependence on n can be eliminated: Bannister et al. [1] proved that every graph with layered pathwidth k has track- and queue-number at most $3k$. Similarly, Dujmović, Morin, and Yelle [17] proved that every graph with layered pathwidth k has stack-number at most $4k$.

Graph colouring is another application area for layered treewidth. Esperet and Joret [19] proved that every graph with maximum degree Δ and Euler genus g is (improperly) 3-colourable with bounded clustering, which means that each monochromatic component has size bounded by some function of Δ and g. This resolved an old open problem even in the planar case ($g = 0$). The clustering function proved by Esperet and Joret [19] is roughly $O(\Delta^{32\Delta\, 2^g})$. While Esperet and Joret [19] made no effort to reduce this function, their method will not lead to a sub-exponential clustering bound. On the other hand, Liu and Wood [27] recently proved that every graph with layered treewidth k and maximum degree Δ is 3-colourable with clustering $O(k^{19}\Delta^{37})$. In particular, every graph with Euler genus g and maximum degree Δ is 3-colorable with clustering $O(g^{19}\Delta^{37})$. This greatly improves upon the clustering bound of Esperet and Joret [19]. Moreover, the proof by Liu and Wood [27] is relatively simple, avoiding many technicalities that arise when dealing with graph embeddings. This result highlights the utility of layered treewidth as a general tool.

1.3 Characterisations

We now turn to the question of characterising those minor-closed classes that have bounded treewidth. The key example is the $n \times n$ grid graph, which has treewidth n. Indeed, Robertson and Seymour [31] proved that every graph with sufficiently large treewidth contains the $n \times n$ grid as a minor. The next theorem follows since every planar graph is a minor of some grid graph. Several subsequent works have improved the bounds [6, 7, 13, 26, 32].

Theorem 1 (Robertson and Seymour [31]). *A minor-closed class has bounded treewidth if and only if some planar graph is not in the class.*

An analogous result for pathwidth holds, where the complete binary tree is the key example (the analogue of grid graphs for treewidth). Let T_h be the complete binary tree of height h. It is well known and easily proved that $\mathrm{pw}(T_h) = \lceil \frac{h}{2} \rceil$, and every forest is a minor of some complete binary tree. Robertson and Seymour [30] proved the following characterisation.

Theorem 2 (Robertson and Seymour [30]). *A minor-closed class has bounded pathwidth if and only if some forest is not in the class.*

Note that Bienstock, Robertson, Seymour, and Thomas [2] proved the following quantitatively stronger result: for every forest T with $|V(T)| \geqslant 2$ every graph containing no T minor has pathwidth at most $|V(T)| - 2$.

Now consider layered analogues of Theorems 1 and 2. A graph G is *apex* if $G - v$ is planar for some vertex v. Define the $n \times n$ *pyramid* to be the apex graph obtained from the $n \times n$ grid by adding one dominant vertex v. (Here a vertex is *dominant* if it is adjacent to every other vertex in the graph.) The $n \times n$ pyramid has treewidth $n + 1$ and layered treewidth at least $\frac{n+2}{3}$, since every layering uses at most three layers. Pyramids are 'universal' apex graphs, in the sense that every apex graph is a minor of some pyramid graph (since every planar graph is a minor of some grid graph). Dujmović et al. [16] proved the following characterisation.

Theorem 3 (Dujmović et al. [16]). *A minor-closed class has bounded layered treewidth if and only if some apex graph is not in the class.*

Theorem 3 generalises the above-mentioned result that graphs of bounded Euler genus have bounded layered treewidth. Note that the proof of Theorem 3 uses the graph minor structure theorem and thus relies on Theorem 1.

A graph G is an *apex-forest* if $G - v$ is a forest for some vertex v. The following theorem is the main result of this paper.

Theorem 4. *A minor-closed class has bounded layered pathwidth if and only if some apex-forest is not in the class.*

Theorem 4 is analogous to Theorem 3 for layered treewidth. However, unlike the proof of Theorem 3 which depends on Theorem 1, our proof of Theorem 4 does not depend on Theorem 2. In fact, Theorem 4 implies Theorem 2, as we now explain. Let T be a forest, and let G be a graph with no T minor. Let T^+ be the apex-forest obtained from T by adding a dominant vertex v. Let G^+ be the graph obtained from G by adding a dominant vertex x. Suppose for the sake of contradiction that G^+ contains a T^+-minor. A T^+-minor in G^+ can be described by a mapping from the vertices of T^+ to vertex-disjoint trees in G^+ such that whenever two vertices in T^+ are adjacent, the corresponding two trees induce a connected subgraph of G. From this mapping, remove two (not necessarily distinct) trees, the image of v and the tree (if it exists) that contains x. If the tree that contains x was the image of a vertex w in T, then instead map w to the tree that was the image of v. The resulting mapping describes a T-minor in G, as claimed. This contradiction shows that G^+ is T^+-minor-free. By Theorem 4, G^+ has layered pathwidth at most $c = c(T^+)$. Since G^+ has radius 1, at most three layers are used. Thus G^+ and G have pathwidth less than $3c$.

Layered treewidth is closely related to the notion of 'local treewidth', which was first introduced by Eppstein [18] under the guise of the 'treewidth-diameter' property. A graph class $\mathcal{G}$ has *bounded local treewidth* if there is a function f such that for every graph G in $\mathcal{G}$, for every vertex v of G and for every integer $r \geqslant 0$, the subgraph of G induced by the vertices at distance at most r from v has treewidth at most $f(r)$. If $f(r)$ is a linear function, then $\mathcal{G}$ has *linear local treewidth*. See [9, 10, 18, 20, 21] for results and algorithmic applications of local treewidth. Dujmović et al. [16] observed that if some class $\mathcal{G}$ has bounded layered treewidth, then $\mathcal{G}$ has linear local treewidth. On the other hand, bounded layered treewidth is a stronger property that bounded or linear local treewidth.

Local pathwidth is defined similarly to local treewidth. A graph class $\mathcal{G}$ has *bounded local pathwidth* if there is a function f such that for every graph G in $\mathcal{G}$, for every vertex v of G and for every integer $r \geqslant 0$, the subgraph of G induced by the vertices at distance at most r from v has pathwidth at most $f(r)$. The observation of Dujmović et al. [16] extends to the setting of local pathwidth; see Lemma 9 below.

Theorem 4 is extended to capture local pathwidth by the following theorem, which also provides a structural description in terms of a tree decomposition with certain properties that we now introduce. If T is a tree indexing a tree decomposition of a graph G, then for each vertex v of G, let $T[v]$ denote the subtree of T induced by those nodes corresponding to bags that contain v. Thus $T[v]$ is non-empty and connected. Say that a tree decomposition of a graph G is (w,p)-*good* if its width is at most w and, for every $v \in V(G)$, the subtree $T[v]$ has pathwidth at most p. We illustrate this definition with two examples. Let T be a tree, rooted at some vertex. For each node x of T, introduce a bag B_x consisting of x and its parent node (or just x if x is the root). Then $(B_x : x \in V(T))$ is a tree decomposition of T with width 1. Moreover, for each vertex v, the subtree $T[v]$ is a star, which has pathwidth 1. Thus every tree has a $(1,1)$-good tree decomposition. Now, consider an outerplanar triangulation G. Let T be the weak dual tree (ignoring the outerface). For each node x of T, let B_x be the set of three vertices on the face corresponding to x. Then $(B_x : x \in V(G))$ is a tree decomposition of G with width 2. Moreover, for each vertex v of G, the subtree $T[v]$ is a path, which has pathwidth 1. Thus every outerplanar graph has a $(2,1)$-good tree decomposition (since every outerplanar graph is a subgraph of an outerplanar triangulation). These constructions are generalised via the following theorem, which immediately implies Theorem 4.

Theorem 5. *The following are equivalent for a minor-closed class $\mathcal{G}$:*

(1) some apex-forest graph is not in $\mathcal{G}$,

(2) $\mathcal{G}$ has bounded local pathwidth,

(3) $\mathcal{G}$ has linear local pathwidth.

(4) $\mathcal{G}$ has bounded layered pathwidth,

(5) there exist integers w and p, such that every graph in $\mathcal{G}$ has a (w,p)-good tree decomposition.

Here is some intuition about property (5). Suppose that $\mathcal{G}$ excludes some apex-forest graph as a minor. Since every apex-forest graph is planar, by Theorem 1, the graphs in $\mathcal{G}$ have bounded treewidth. Thus we should expect that the tree decompositions in (5) have bounded width. Moreover, if $\mathcal{G}$ has bounded layered pathwidth, then $G[N(v)]$ has bounded pathwidth for each vertex v in each graph $G \in \mathcal{G}$. Property (5) takes this idea further, and says that each subtree $T[v]$ has bounded pathwidth, which implies that $G[N(v)]$ has bounded pathwidth (since the width of the tree decomposition is bounded).

In Section 2 we prove (5) $\implies$ (4) $\implies$ (3) $\implies$ (2) $\implies$ (1). In Section 3 we close the loop by proving (1) $\implies$ (5). This proof uses a recent characterisation by Dang [8] of the unavoidable minors in 3-connected graphs of large pathwidth.

Throughout the proof we use the following 'universal' apex-forest graph. Let Q_k be the graph obtained from the complete binary tree T_k by adding one dominant vertex. Note that $\mathrm{pw}(Q_k) = \lceil \frac{k}{2} \rceil + 1$ and the layered pathwidth of Q_k is at least $\frac{k+4}{6}$, since every layering of Q_k uses at most three layers. Since every forest is a minor of some complete binary tree, every apex-forest graph is a minor of some Q_k.

2 Downward Implications

We start with a few simple but useful lemmas.

Lemma 6. *If a graph G has a tree decomposition of width k indexed by a tree of pathwidth p, then G has pathwidth at most $(p+1)(k+1)-1$.*

Proof. Let $(B_x : x \in V(T))$ be a tree decomposition of G of width k. Let $(C_1, \dots, C_n)$ be a path decomposition of T of width p. For $i \in \{1, \dots, n\}$, let $D_i := \bigcup_{x \in C_i} B_x$. Then $(D_1, \dots, D_n)$ is a path decomposition of G of width $(p+1)(k+1)-1$ (since $|C_i| \leqslant p+1$ and $|B_x| \leqslant k+1$). □

Lemma 7. *Let T_1 and T_2 be subtrees of a tree T, such that $T = T_1 \cup T_2$. Then*

$$\mathrm{pw}(T) + 1 \leqslant (\mathrm{pw}(T_1) + 1) + (\mathrm{pw}(T_2) + 1).$$

Proof. Let $(B_1, \dots, B_s)$ be a path decomposition of T_1 with bag size at most $\mathrm{pw}(T_1) + 1$. Each component of $T - V(T_1)$ is contained in T_2 and therefore has a path decomposition with bag size at most $\mathrm{pw}(T_2) + 1$. For each such component J of $T - V(T_1)$, there is exactly one vertex v in T_1 adjacent to some vertex in J (otherwise T would contain a cycle consisting of two edges between a path in T_1 and a path in J). Say v is in bag B_i. We say J *attaches* at v and at B_i. By doubling bags in the path decomposition of T_1, we may assume that distinct components of $T - V(T_1)$ attach at distinct B_i. For each component J of $T - V(T_1)$, if $(D_1, \dots, D_t)$ is a path decomposition of J with bag size at most $\mathrm{pw}(T_2) + 1$, then replace B_i by $(B_i \cup D_1, \dots, B_i \cup D_t)$. We obtain a path decomposition of T with bag size at most $(\mathrm{pw}(T_1) + 1) + (\mathrm{pw}(T_2) + 1)$. The result follows. □

Corollary 8. *Let $T_1, \dots, T_k$ be subtrees of a tree T, such that $T = T_1 \cup \dots \cup T_k$. Then*

$$\mathrm{pw}(T) + 1 \leqslant \sum_{i=1}^{k} (\mathrm{pw}(T_i) + 1).$$

We now prove the downward implications in Theorem 5. First note that (2) implies (1), since if every graph in $\mathcal{G}$ has local pathwidth at most k, then the apex-forest graph Q_{6k} is not in $\mathcal{G}$. It is immediate that (3) implies (2). That (4) implies (3) is the above-mentioned observation of Dujmović et al. [16] specialised for pathwidth. We include the proof for completeness.

Lemma 9. *Let $\mathcal{G}$ be a class of graphs such that every graph in $\mathcal{G}$ has layered pathwidth at most k. Then $\mathcal{G}$ has linear local pathwidth with binding function $f(r) = (2r+1)k - 1$.*

Proof. For a graph $G \in \mathcal{G}$, let $(B_1, \dots, B_s)$ be a path decomposition of G with layered width k, with respect to some layering $V_0, V_1, \dots, V_t$. Let v be a vertex in V_i. Let r be a positive integer. Let H be the subgraph of G induced by the vertices at distance at most r from v. Thus $V(H) \subseteq V_{i-r} \cup V_{i-r+1} \cup \cdots \cup V_{i+r}$. Each bag B_j contains at most k vertices in each layer. Hence $(B_1 \cap V(H), \dots, B_s \cap V(H))$ is a path decomposition of H with at most $(2r+1)k$ vertices in each bag. Therefore $\mathcal{G}$ has linear local pathwidth with binding function $f(r) = (2r+1)k - 1$. □

The next lemma shows that (5) implies (4).

Lemma 10. *If a graph G has a (w,p)-good tree decomposition, then $\operatorname{lpw}(G) \leqslant w(p+1)(w+1)$.*

Proof. Let $\mathcal{T} = (B_x : x \in V(T))$ be a tree decomposition of G with width w, such that $\operatorname{pw}(T[v]) \leqslant p$ for each vertex v of G. Since adding edges does not decrease the layered pathwidth, we may add edges to G between two non-adjacent vertices in the same bag of $\mathcal{T}$. Now each bag is a clique, and G is chordal with maximum clique size $w+1$. Let $V_0, V_1, \dots, V_t$ be a bfs layering in G. That is, V_i is the set of vertices in G at distance i from some fixed vertex r of G. In particular, $V_0 = \{r\}$.

Consider a component H of $G[V_i]$ for some $i \geqslant 1$. Let C_H be the set of vertices in V_{i-1} adjacent to at least one vertex in H. Since G is chordal, C_H is a clique of size at most w (see [15, 25]), called the *parent clique* of H. Define $T_H := \bigcup_{w \in C_H} T[w]$. Since C_H is a clique, which is contained in a single bag of $\mathcal{T}$, there is a node x of T such that $x \in T[w]$ for each $w \in C_H$. Thus T_H is a (connected) subtree of T. Moreover, T_H is the union of at most w subtrees, each with pathwidth at most p. Thus $\operatorname{pw}(T_H) + 1 \leqslant w(p+1)$ by Corollary 8. Let $\hat{H} := G[V(H) \cup C_H]$.

We now prove that $\mathcal{T}_H := (B_x \cap V(\hat{H}) : x \in V(T_H))$ is a tree decomposition of $\hat{H}$. We first prove condition (ii). For a vertex v of C_H, the set of bags of $\mathcal{T}_H$ that contain v is precisely those indexed by nodes in $T[v]$, which is non-empty and connected, by assumption. Now, consider a vertex v in H. Let w be the neighbour of v on a shortest vr-path in G. Thus w is in C_H. Since vw is an edge, v and w appear in a common bag of $\mathcal{T}$, which corresponds to a node in T_H (since that bag contains w). Hence $T_H[v]$ is non-empty. We now prove that $T_H[v]$ is connected. Let B_1 and B_2 be distinct bags of $\mathcal{T}_H$ containing v. Let P be the B_1B_2-path in T. Since $T[v]$ is connected, v is in the bag associated with each node in P. To conclude that $T_H[v]$ is connected, it remains to prove that $P \subseteq T_H$. By construction, some vertex w_1 is in $B_1 \cap C_H$ and some vertex w_2 is in $B_2 \cap C_H$. Since w_1 and w_2 are adjacent, the bag associated with each node in P contains w_1 or w_2. Hence $P \subseteq T_H$ and $T_H[v]$ is connected. This proves condition (ii). Now we prove condition (i). Since C_H is contained in some bag of

$\mathcal{T}_H$, condition (i) holds for each edge with endpoints in C_H. For each edge vw with $v \in V(H)$ and $w \in C_H$, v and w are in a common bag B_x of $\mathcal{T}$, implying x is in T_H (since B_x contains w), as desired. Finally, consider an edge uv with $u, v \in V(H)$. Suppose on the contrary that u and v have no common neighbour in C_H. By construction, u has a neighbour w_1 in C_H, and v has a neighbour w_2 in C_H. Thus $w_1 \neq w_2$. Since C_H is a clique, w_1 and w_2 are adjacent. Since $uw_2 \notin E(G)$ and $vw_1 \notin E(G)$, the 4-cycle (u, w_1, w_2, v) is chordless, and G is not chordal, which is a contradiction. Hence u and v have a common neighbour w in C_H. Thus $\{u, v, w\}$ is a triangle in G, which is in a common bag of T, and therefore in a common bag of $\mathcal{T}_H$, implying that u and v are in a common bag of $\mathcal{T}_H$. This proves condition (i) in the definition of tree decomposition. Therefore $\mathcal{T}_H$ is a tree decomposition of $\hat{H}$. By construction, it has width at most w.

Since $\mathrm{pw}(T_H) + 1 \leqslant w(p+1)$ and T_H indexes a tree decomposition of $\hat{H}$ with width at most w, by Lemma 6, $\mathrm{pw}(\hat{H}) \leqslant (w(p+1)+1)(w+1) - 1$.

We now construct a path decomposition of G with layered width at most $w(p+1)(w+1)$ with respect to layering $V_0, V_1, \ldots, V_t$. Let $G_i := G[V_0 \cup V_1 \cup \cdots \cup V_i]$. We now prove, by induction on i, that G_i has a path decomposition with layered width at most $w(p+1)(w+1)$ with respect to layering $V_0, V_1, \ldots, V_i$. This claim is trivial for $i = 0$. Now assume that $(B_1, \ldots, B_q)$ is a path decomposition of G_{i-1} with layered width at most $w(p+1)(w+1)$ with respect to layering $(V_0, V_1, \ldots, V_{i-1})$. For each component H of $G[V_i]$, there is a bag B_j that contains C_H; pick one such bag and call it the *parent bag* of H. By doubling the bags, we may assume that distinct components of $G[V_i]$ have distinct parent bags. Now, for each component H of $G[V_i]$ with parent bag B_j, if $(D_1, \ldots, D_s)$ is a path decomposition of $\hat{H}$ with width $w(p+1)(w+1) - 1$, then replace B_j by $B_j \cup D_1, \ldots, B_j \cup D_s$. Doing this for each component of $G[V_i]$ produces a path decomposition of G_i with layered width at most $w(p+1)(w+1)$ with respect to layering $(V_0, V_1, \ldots, V_i)$. In particular, we obtain a path decomposition of G with layered width at most $w(p+1)(w+1)$ with respect to layering $(V_0, V_1, \ldots, V_t)$. □

3 Proof that (1) implies (5)

The goal of this section is to show that if a graph G excludes some apex-forest graph H as a minor, then G has (w, p)-good tree decomposition for some $w = w(H)$ and $p = p(H)$. Since every apex-forest graph is a minor of some Q_k, it suffices to prove this result for $H = Q_k$, in which case we denote $w = w(k)$ and $p = p(k)$.

We will be working with two related trees S and T and one graph G. To help the reader keep track of things we use variables a, b, and c as names for *nodes* of S and T and variables v, x, y, and z to refer to *vertices* of G.

We now give an outline of the proof. First, we show that a recent result by Dang [8] implies that every 3-connected graph G with no Q_k minor has pathwidth at most $w = w(k)$. Thus, in this case, G has a $(w, 1)$-good tree decomposition. Next we deal with cut vertices by showing that if each block of a graph G has a (w, p)-good tree decomposition, then G has a $(w, p+1)$-good tree decomposition.

Therefore, the main difficulty is to show that every 2-connected graph G with no Q_k minor has a (w,p)-good tree decomposition $(B_a : a \in V(T))$. By the result of Bienstock et al. [2] described in the introduction, if $\mathrm{pw}(T[v]) > 2^{h+1} - 3$ for some $v \in V(G)$ then $T[v]$ contains a T_h minor. For sufficiently large h, we then construct a Q_k minor (from the T_h minor in $T[v]$) in which v plays the role of the apex vertex.

To construct the tree decomposition $(B_a : a \in V(T))$ we use two tools: An SPQR-tree, S, represents a graph G as a collection of subgraphs (S- and R-nodes) that are joined at 2-vertex cutsets (P-nodes). These subgraphs consist of cycles (S-nodes) and 3-connected graphs (R-nodes). Cycles have pathwidth 2 and, by the result of Dang discussed above, the 3-connected graphs have pathwidth at most $w = w(k)$. Replacing the S- and R-nodes of the SPQR-tree with these path decompositions produces the tree T in our tree decomposition.

To show that this tree decomposition is (w,p)-good, we first show that if $T[v]$ contains a subdivision of a sufficiently large complete binary tree, then the SPQR-tree S also contains a subdivision of a large complete binary tree all of whose nodes have subgraphs that contain v. Using this large binary tree in S we then piece together a subgraph of G that has a Q_k minor in which v is the apex vertex.

3.1 Dang's Result

First we show how the following result of Dang [8] implies that every 3-connected graph with no Q_k minor has pathwidth at most $w = w(k)$.

Theorem 11 (Dang [8, Theorem 1.1.5]). *Let P be a graph with two distinct vertices u_1 and u_2 such that $P - \{u_1, u_2\}$ is a forest, Q be a graph with a vertex v such that $Q - v$ is outerplanar, and R be a tree with a cycle going through its leaves in order from the leftmost leaf to the rightmost leaf so that R is planar. Then there exists a number $w = w(P,Q,R)$ such that every 3-connected graph of pathwidth at least w has a P, Q, or R minor.*

Note that R is a Halin graph, except that degree-2 vertices are allowed in the tree.

To use Theorem 11 we need a small helper lemma. For every $k \geqslant 0$, let T_k^+ be the graph obtained from the complete binary tree T_k of height k by adding a new vertex adjacent to the leaves. The next lemma is well known.

Lemma 12. *For every integer $k \geqslant 0$, T_{2k}^+ contains Q_k as a minor.*

Proof. The statement is immediate for $k = 0$. For $k \geqslant 1$, partition the edges of T_{2k}^+ into the tree T_{2k} and the remaining edges, which form a star centered at some vertex v. Let a_1, a_2, a_3, a_4 be the grandchildren of the root of T_{2k} ordered from left to right. Contract the entire subtree comprised of the subtree rooted at a_2, the subtree rooted at a_3, and the path from a_2 to a_3. Applying the same procedure recursively on the copy of $T_{2(k-1)}^+$ rooted at a_1 and the copy of $T_{2(k-1)}^+$ rooted at a_4 produces Q_k, as can be easily verified by induction. □

Corollary 13. *There exists a number $w = w(k)$ such that every 3-connected graph of pathwidth at least w has a Q_k minor.*

Proof. Let P be obtained from the complete binary tree T_k by adding two dominant vertices u_1 and u_2. Let Q be the graph obtained from the outerplanar graph ∇_k, whose weak dual is a complete binary tree of height k, by adding a dominant vertex v. Let R be the graph obtained from T_{2k+1} by adding a cycle on its leaves, so that R is planar.

Then P contains Q_k as a minor since $P - \{u_2\}$ is isomorphic to Q_k. Q also contains Q_k as a minor because ∇_k contains a complete binary tree of height k as a subgraph. Finally, R also contains a Q_k minor: Contract the cycle, then we have a complete binary tree of height $2k$ plus an apex vertex linked to its leaves, which contains Q_k as a minor by Lemma 12. Theorem 11 implies that there exists $w = w(k)$ such that every 3-connected graph with pathwidth at least w contains at least one of P, Q, or R as a minor and therefore contains a Q_k minor. □

3.2 Dealing with Cut Vertices

A *block* in a graph is either a maximal 2-connected subgraph, the subgraph induced by the endpoints of a bridge edge, or the subgraph induced by an isolated vertex.

Lemma 14. *Let G be a graph, such that each block of G has a (w,p)-good tree decomposition. Then G has a $(w,p+1)$-good tree decomposition.*

Proof. Let $C_1,\dots,C_r$ be the blocks of G. For each $i \in \{1,\dots,r\}$, let T_i be the underlying tree in a (w,p)-good tree decompositions of C_i.

We create a tree decomposition of G as follows: For each cut vertex or isolated vertex v in G, introduce a new tree node a_v with $B_{a_v} = \{v\}$. In each block C_i that contains v, the tree decomposition $(B_a : a \in V(T_i))$ of C_i has at least one node a such that $v \in B_a$; make a_v adjacent to exactly one such node for each C_i.

It is straightforward to verify that this defines a tree decomposition of G and we now argue this decomposition is $(w,p+1)$-good. The resulting tree decomposition of G has width at most w. For each isolated vertex $v \in V(G)$, the subtree $T[v]$ consists of one node. For each cut vertex $v \in V(G)$, the subtree $T[v]$ is composed of some number of subtrees, each adjacent to a_v and each having a path decomposition of width at most p. We obtain a path decomposition of $T[v]$ by concatenating the path decompositions of each subtree and adding v to every bag of the resulting path decomposition. The resulting path decomposition of $T[v]$ has width at most $p+1$. □

3.3 SPQR-Trees

In this section, we quickly review SPQR-trees, a structural decomposition of 2-connected graphs used previously to characterize planar embeddings [28], to design efficient algorithms for triconnected components [23], and in efficient data structures for incremental planarity testing [11, 12].

Let G be a 2-connected graph. An *SPQR-tree* S of G is a tree in which each node $a \in V(S)$ is associated with a minor H_a of G. For any S- or R-node a of S, H_a is a simple graph. If a is a P-node, on the other hand, then H_a is a *dipole graph* having two vertices and at least

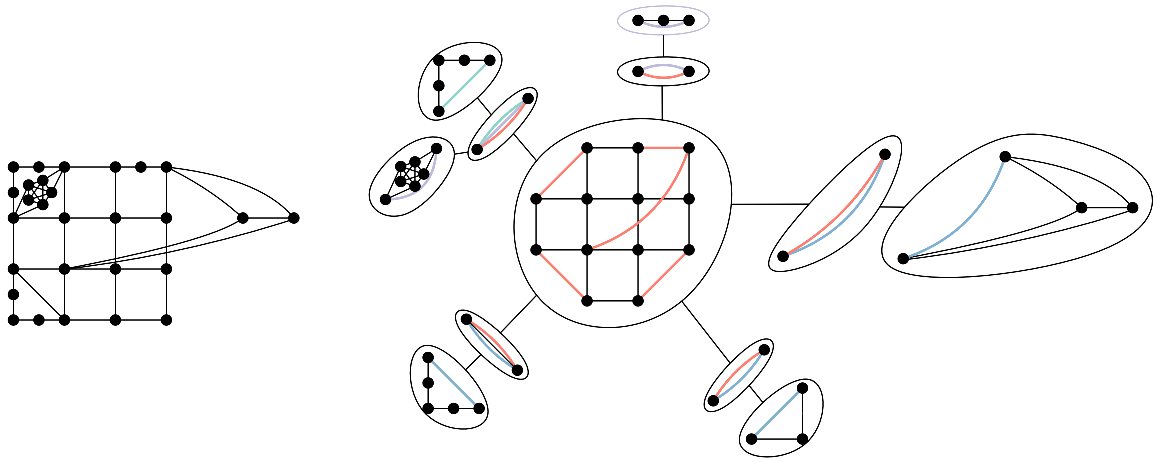

Figure 1: A graph and its SPQR-tree.

two parallel edges. In all cases, H_a is a minor of G. For a P-node a in which H_a contains vertices x and y and t parallel edges, this means that G contains t internally disjoint paths from x to y. For each node a of S each edge $xy \in E(H_a)$ is classified either as a *virtual edge* or a *real edge*. An SPQR-tree S is defined recursively as follows (see Figure 1):[1]

1. If G is a cycle, then S consists of a single node a (an S-node) in which $H_a = G$ and all edges of H_a are real.

2. If G is 3-connected, then S consists of a single node a (an R-node) in which $H_a = G$ and all edges of H_a are real.

3. Otherwise G has a cutset $\{x, y\}$ such that x and y each have degree at least 3. Then let $C_1, \dots, C_r$, $r \geqslant 2$, be the connected components of $G - \{x, y\}$. For each $i \in \{1, \dots, r\}$, let $\tilde{G}_i$ be $G[V(C_i) \cup \{x, y\}]$ along with the additional edge xy, if not already present. Note that (because of the inclusion of xy) each $\tilde{G}_i$ is 2-connected, so each has an SPQR-tree S_i. Then an SPQR-tree for G is obtained by creating a node a (a P-node) with H_a being a dipole graph with vertices x and y and having r virtual edges joining x and y. In addition to these virtual edges, H_a contains the real edge xy if $xy \in E(G)$. The construction and the fact that xy is an edge in each $\tilde{G}_i$ imply that, for each $i \in \{1, \dots, r\}$, there exists exactly one node a_i in S_i such that xy is a real edge in H_{a_i}. To complete S, make a adjacent to each of $a_1, \dots, a_r$, and make xy a virtual edge in each of $H_{a_1}, \dots, H_{a_r}$.

Let S be an SPQR-tree of a 2-connected graph G. For each node a of S, we let $E_r(H_a)$ denote the set of real edges in H_a and $E_v(H_a)$ denote the multiset of virtual edges in H_a. For a connected subtree S' of S we define $G[S']$ as the subgraph of G whose vertex set is $V(G[S']) = \bigcup_{a \in V(S')} V(H_a)$ and whose edge set is $E(G[S']) = \bigcup_{a \in V(S')} E_r(H_a)$. For a vertex $v \in V(G)$, let $S[v] := S[\{a \in V(S) : v \in V(H_a)\}]$, which is called the subtree of S *induced by* v. We make use of the following properties of S:

[1] This definition includes P-nodes consisting of only two virtual edges, which some works exclude because they are unnecessary. However, their inclusion simplifies some of our analysis.

1. Every R-node and S-node is adjacent only to P-nodes and no two P-nodes are adjacent.
2. The degree of every node a is equal to the number of virtual edges in H_a.
3. For every vertex $v \in V(G)$, $S[v]$ is connected.
4. If a is an R-node or S-node, then H_a is a simple graph; that is, H_a contains no parallel edges.
5. If a P-node a has degree 2 and both its neighbors are S-nodes then H_a has a real edge.
6. For each node a of S and component S' of $S - \{a\}$, $G[S']$ is connected.
7. For each $xy \in E(G)$ there is exactly one node a of S for which xy is a real edge in H_a.

3.4 The Good Tree Decomposition

To obtain our good tree decomposition $(B_a : a \in V(T))$ of a 2-connected graph G we start with an SPQR-tree S for G. For each R-node or S-node a of S, let $(B_c : c \in V(P_a))$ be minimum-width path decomposition of H_a. We say that the node a *generates* the nodes in the path P_a and that each node in P_a is *generated by* a.

Each S- or R-node a is adjacent to some set of P-nodes in S. For each such P-node b whose dipole graph H_b has vertices x and y, the edge xy is a (virtual) edge in H_a and therefore x and y appear in some common bag B_c with $c \in V(P_a)$. We make c and b adjacent in T. This defines the tree T in the tree decomposition.

We now describe the contents of T's bags. Each P-node a of S becomes a node in T whose bag contains only the two vertices of H_a. Every node a in T that is generated by an S- or R-node a' of S is a node in some path decomposition of $H_{a'}$ and already has an associated bag B_a that it inherits from this path decomposition.

It is straightforward to verify that $(B_a : a \in V(T))$ is indeed a tree decomposition of G: For each vertex $v \in V(G)$, the connectivity of the subtree $T[v]$ follows from Property 3 of SPQR-trees and the equivalent property for the path decompositions that include v. Each edge xy of G appears as an edge in H_a for at least one node a of S and therefore x and y appear in a common bag in the path decomposition of H_a.

Each bag B_a of $(B_a : a \in V(T))$ either has size in $\{2, 3\}$ (when a is generated by a P-node or an S-node) or it has size at most $w(k)+1$ where $w(k)$ is the function in Corollary 13 (when a is generated by an R-node). Thus, $(B_a : a \in V(T)\}$ is a tree decomposition of G whose width is upper bounded by a function of k. It remains to show that, for every $v \in V(G)$, $T[v]$ has pathwidth that is upper bounded by a function of k.

In the remainder of this section, we fix G to be a 2-connected graph, S to be an SPQR-tree of G, and $(B_a : a \in V(T))$ to be a tree decomposition of G obtained using the procedure described above.

Lemma 15. *For every integer $h \geqslant 1$, if $T[v]$ has pathwidth greater than $2^{2h+1} - 3$, then $S[v]$ contains a subdivision of T_h.*

Proof. In the following, a *binary tree* is a tree rooted at a degree-2 node, such that every other node has degree in $\{1, 2, 3\}$. In a binary tree, the root and every degree 3 node is called a *branching node*. Every branching node and every leaf is a *distinctive node*. We use the convention that all binary trees are ordered, possibly arbitrarily, so that we can distinguish between the left and right child of a branching node. For a node a in a binary tree T, we denote by $\hat{a}$ the subtree rooted at a; that is, the subtree of T induced by the set of nodes that have a as an ancestor, including a itself.

Recall, from the result of Bienstock et al. [2] discussed in the introduction, that if $T[v]$ has pathwidth greater than $2^{2h+1} - 3$ then T contains a subdivision T' of T_{2h}. Note that T' does not immediately imply the existence of T_h in $S[v]$ since two or more distinctive nodes of T' may have been generated by the same node of S. Label each node of T' with the node of S that generated it. Recall that each node a in S generates a path in T. So a maximal subset of nodes of T' with a common label induces a path in T'.

We claim that T' contains a subdivision T'' of T_h such that each of the distinctive nodes of T'' has a unique label. We establish this claim by induction on h: If $h = 0$ then the claim is trivial. Otherwise, let a be the root of T' and let a' and a'' be the highest branching nodes in the left and right subtrees of $\hat{a}$, respectively. Let a_1 and a_2 be the highest distinctive nodes in the left and right subtrees of $\hat{a}'$, respectively, and let a_3 and a_4 be the highest distinctive nodes in the left and right subtrees of $\hat{a}''$, respectively. Since each label induces a path in T', at least one of $\{a_1, a_2\}$, say a_1, and at least one of $\{a_3, a_4\}$, say a_4, does not have the same label as a. Furthermore, since a_1 and a_4 are separated by a, the set of labels of nodes in $\hat{a}_1$ is disjoint from the set of labels of nodes in $\hat{a}_4$. Applying induction on $\hat{a}_1$ and $\hat{a}_4$ yields two subdivisions of T_{h-1} in which each distinctive node has a unique label. Connecting these two subdivisions with the unique path from a_1 to a_4 yields the desired subdivision of T_h in which each distinctive node has a unique label.

Now, each distinctive node in T'' has a unique label and therefore corresponds to a unique node of S. Thus, if we contract all nodes of T'' sharing a common label, then we obtain a subtree T''' of $S[v]$ that is a subdivision of T_h. □

Thus far we have established that if $T[v]$ has sufficiently high pathwidth, then $S[v]$ contains a subdivision of a large complete binary tree.

Lemma 16. *If $S[v]$ contains a subdivision of $T_{7(k+1)}$ then G contains a Q_k minor.*

Proof. First we note that if $S[v]$ contains a subdivision of $T_{7(k+1)}$ then $S[v]$ contains a subdivision T' of T_{k+1} such that the path between each pair of distinctive nodes in T' has length at least 7.

It is convenient to work with a simplified SPQR-tree S' and graph G' obtained by repeating the following operation exhaustively: Consider some edge ab of S with $a \in V(T')$ and $b \notin V(T')$. The edge ab is associated with some virtual edge xy in H_a. In S', replace the virtual edge xy in H_a with a real edge. At the same time, remove the maximal subtree $\hat{b}$ of S that contains b and not a. By Property 6 of SPQR trees, in G' this operation is equivalent to contracting all the real edges in $\bigcup_{c \in V(\hat{b})} E_r(H_c)$ and removing any resulting parallel edges. Since the

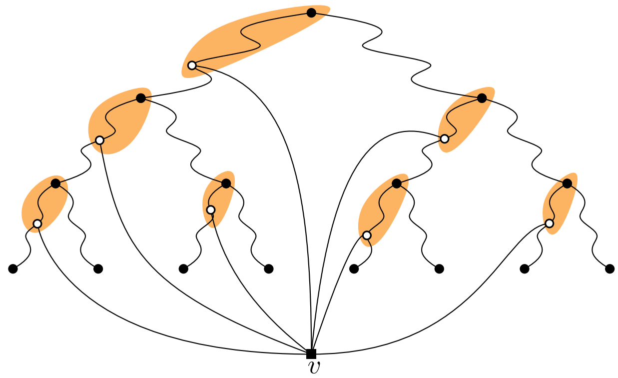

Figure 2: Finding a Q_k minor in the proof of Lemma 16. Distinctive nodes are indicated by black disks and anchor nodes by white circles.

resulting graph G' is a minor of G, this operation is safe in the sense that the existence of a Q_k minor in G' implies the existence of a Q_k minor in G.

With this simplification, the tree T' is an SPQR-tree for the graph G' and every virtual edge is incident to v. We now turn our efforts to finding the Q_k minor in G'. Recall that Q_k is obtained from a complete binary tree T_k by adding a dominant vertex. We begin by finding a subdivision T'' of T_{k+1} in G'. In this subdivision, each edge of T_{k+1} that joins a node to its left child is represented by a path $P_{\mu\nu}$ joining a branching node μ to a distinctive node ν. We show that G' contains a path from v to some *anchor node* η of $P_{\mu\nu}$ with $\eta \neq \nu$, which is vertex disjoint from T'' except for η. Furthermore, except for their common endpoint v, all of these paths are disjoint. The union of T'' and these paths contains a Q_k minor since contracting the path from each anchor node to its closest ancestor branching node produces Q_k. See Figure 2.

Let a be a branching node of T' and let b be the nearest distinctive node in one of a's two subtrees. Consider the path $a = c_1, c_2, \dots, c_r = b$ in T'. For each $i \in \{1, \dots, r-1\}$, the edge $c_i c_{i+1}$ is associated with a cutset $\{v, x_i\}$ in G' and vx_i is a virtual edge in H_{c_i} and $H_{c_{i+1}}$. Note that this implies that, for each $i \in \{2, \dots, r\}$, H_{c_i} contains both vertices x_i and x_{i-1}.

We claim that, for each $i \in \{2, \dots, r-1\}$, H_{c_i} contains a path P_i from x_{i-1} to x_i that does not contain v; refer to Figure 3. When c_i is a P-node, this claim is trivial since, in this case, $x_{i-1} = x_i$. The case in which c_i is an S-node or R-node is also easy: In these cases H_{c_i} is 2-connected, therefore there is a path from x_{i-1} to x_i that avoids v. Now note that the paths $P_1, \dots, P_{r-1}$ are disjoint, except for each of the common endpoints x_i where P_i ends and P_{i+1} begins. This is because each $\{v, x_i\}$ is a cutset of G' that separates $\bigcup_{j=1}^{i} V(H_{c_j}) \setminus \{v, x_i\}$ from $\bigcup_{j=i+1}^{r} V(H_{c_j}) \setminus \{v, x_i\}$. By concatenating $P_2, \dots, P_{r-1}$ we obtain a path P_{ab} from $x_1 \in V(H_a)$ to $x_{r-1} \in V(H_b)$ that we call the *subdivision path* for nodes a and b.

By Properties 2 and 4 of SPQR-trees and the fact that all virtual edges are incident to v, every branching node a of T' is either an R-node or a P-node. In the case where a is a P-node, all the subdivision paths that begin or end at a vertex of H_a include the same vertex of H_a. In the case where a is an R-node, each subdivision path that begins or ends at a vertex of H_a includes a different vertex (for up to 3 different vertices x, y, and z). Now, since H_a is 3-connected, these three vertices are in the same component of $H_a - \{v\}$. In particular,

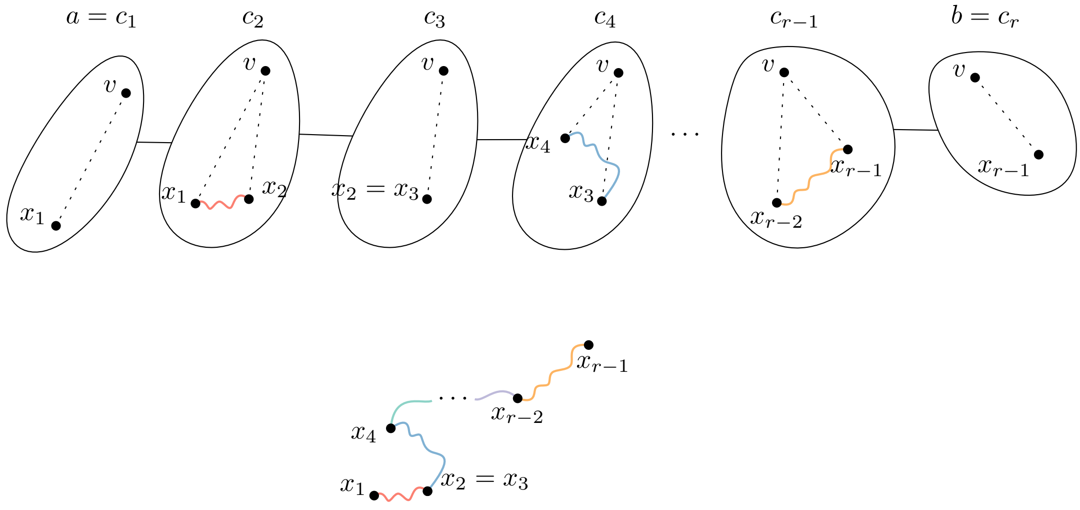

Figure 3: Finding a path connecting a vertex of H_a to a vertex of H_b.

$H_a - \{v\}$ contains an edge-minimal tree that includes x, y, and z. Adding each of these trees to the union of all subdivision paths produces the subdivision T'' of T_{k+1}.

Next we show how to construct paths from v to anchor nodes. Let a be a branching node of T', let b be the highest distinctive node in a's left subtree and let $c_1 = a, \ldots, c_r = b$ be the path in T' having endpoints a and b. Thus far we have established that $G'[\{c_2, \ldots, c_{r-1}\}]$ contains a simple path P_{ab} from x_1 to x_{r-1} that does not include v. We now show that $G'[\{c_2, c_3, c_4, c_5\}]$ contains an *apex path* P' from v to some *anchor node* of P_{ab} such that the internal vertices of P' are disjoint from $V(P_{ab}) \cup V(H_b)$. We first describe the path P' in $G'[\{c_2, c_3, c_4, c_5\}]$ and then show that P' contains no vertex of $V(H_b) \setminus \{v\}$. There are two cases to consider:

1. c_i is an R-node, for some $i \in \{2, \ldots, 5\}$: Since H_{c_i} is 3-connected, there are three paths in H_{c_i} with endpoints v and x_i and no other vertices in common. Since H_{c_i} has only two virtual edges, at least one of these paths uses only real edges in H_{c_i}. This path therefore contains a subpath P' joining v to some vertex of P_{ab} (the anchor vertex) that is otherwise disjoint from P_{ab}.

2. Otherwise, none of $c_2, \ldots, c_5$ is an R-node. Property 1 of SPQR-trees implies that, for at least one $i \in \{2, 3\}$, c_i is an S-node, c_{i+1} is a P-node and c_{i+2} is an S-node. But in this case, Property 5 of SPQR-trees ensures that $H_{c_{i+1}}$ contains the real edge vx_{i+1}. Therefore, in this case there is a single edge path P' from v to P_{ab}.

It remains to show that P' does not contain any vertices of $V(H_b) \setminus \{v\}$. By Properties 1 and 6 of SPQR-trees, for each $x \in V(G') \setminus \{v\}$, the subtree $T'[x]$ of T' consisting only of nodes a such that $x \in V(H_a)$ is a star; that is, the distance between any two nodes in $T'[x]$ is at most 2. Now, since the distance between any two distinctive nodes of T' is at least 7, we have $r \geqslant 8$ and therefore $H_b = H_{c_r}$ has no vertex, except v, in common with any of $H_{c_2}, \ldots, H_{c_5}$. Therefore, P' joins v to a vertex in P_{ab} that is not in H_b, as required.

Adding the set of all apex paths to T'' then produces a subgraph of G' that contains a Q_k-minor. □

Finally, we have all the pieces in place to complete the proof.

Proof that (1) implies (5). Let G be a graph excluding some apex-forest graph H as a minor. As explained earlier, G contains no Q_k minor for some $k = k(H)$. We wish to show that there are w and p that depend only on k such that G has a (w, p)-good tree decomposition. By Lemma 14 we may assume that G is 2-connected.

Consider the good tree decomposition $(B_a : a \in V(T))$ of G described in Section 3.4. This decomposition has width at most w where $w = w(k)$ is the function that appears (implicitly) in Theorem 11 and Corollary 13. We claim that, for each $v \in V(G)$, $\mathrm{pw}(T[v]) \leqslant 2^{14(k+1)} - 3$, so that this tree decomposition is $(w, 2^{14(k+1)} - 3)$-good. Otherwise, by Lemma 15, there is a vertex $v \in V(G)$ such that an SPQR-tree S has a subtree $S[v]$ that contains a subdivision of $T_{7(k+1)}$. Therefore, by Lemma 16, G contains a Q_k minor, contradicting the supposition that G has no Q_k minor. □

Note that we have not tried to optimise constants in the above proof. For example, with more work the constant 14 can be reduced to less than 3.

Finally, we address the computational complexity of our main theorem.

Theorem 17. *There exists a function $f : \mathbb{N} \to \mathbb{N}$ and an algorithm that takes as input an n-vertex graph and outputs, in $O(f(k)n)$ time, an $(f(k), f(k))$-good tree decomposition of G and a layered path decomposition of G of layered width at most $f(k)$, where k is largest integer such that G contains a Q_k minor.*

Proof Sketch. In the following, for each $i \in \{0, \dots, 5\}$, $f_i : \mathbb{N} \to \mathbb{N}$ is an unspecified function that is known to exist. Since G has no Q_k-minor, $|E(G)| \leqslant f_0(k)n$ (see [29]). An SPQR-tree S of G can be computed in $O(f_0(k)n)$ time, and the total size of the graphs $\{H_a : a \in V(S)\}$ is at most $f_0(k)n$ (see [22]). A path decomposition of H_a with width at most 2 for each S-node or P-node a is easily computed in time linear in the size of H_a. For each R-node a, the pathwidth of H_a is at most $p = p(k) = O(1)$. Therefore, a minimum-width path decomposition of H_a for an R-node a can be computed in $O(f_1(k)|V(H_a)|)$ time [3–5, 24]. These path decompositions are all that is needed to construct the tree T and an $(f_2(k), f_3(k))$-good tree decomposition $\{B_a : a \in V(T)\}$ in $O(f_1(k)n)$ time.

The proof that (5) implies (4) in Section 2 is constructive and immediately gives an $O(f_4(k)n)$ time algorithm to convert the $(f_2(k), f_3(k))$-good tree decomposition into a layered path decomposition of layered width at most $f_5(k)$. This establishes the theorem for $f(k) = \max\{f_i(k) : i \in \{0, \dots, 5\}\}$. □

References

[1] Michael J. Bannister, William E. Devanny, Vida Dujmović, David Eppstein, and David R. Wood. Track layouts, layered path decompositions, and leveled planarity. *Algorithmica*, 2018. doi: 10.1007/s00453-018-0487-5.

[2] Dan Bienstock, Neil Robertson, Paul Seymour, and Robin Thomas. Quickly excluding a forest. *J. Combin. Theory Ser. B*, 52(2):274–283, 1991. doi: 10.1016/0095-8956(91)90068-U. MR: 1110475.

[3] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996. doi: 10.1137/S0097539793251219.

[4] Hans L. Bodlaender and Ton Kloks. Better algorithms for the pathwidth and treewidth of graphs. In Javier Leach Albert, Burkhard Monien, and Mario Rodríguez-Artalejo, eds., *Proc. 18th International Colloquium on Automata, Languages and Programming (ICALP '91)*, vol. 510 of *Lecture Notes in Computer Science*, pp. 544–555. Springer, 1991. doi: 10.1007/3-540-54233-7_162.

[5] Kevin Cattell, Michael J. Dinneen, and Michael R. Fellows. A simple linear-time algorithm for finding path-decompositions of small width. *Inf. Process. Lett.*, 57(4):197–203, 1996. doi: 10.1016/0020-0190(95)00190-5.

[6] Chandra Chekuri and Julia Chuzhoy. Polynomial bounds for the grid-minor theorem. *J. ACM*, 63(5):40, 2016. doi: 10.1145/2820609. MR: 3593966.

[7] Julia Chuzhoy. Excluded grid theorem: Improved and simplified. In *Proc 48th Annual ACM on Symposium on Theory of Computing*, STOC '15, pp. 645–654. ACM, 2015. doi: 10.1145/2746539.2746551.

[8] Thanh N. Dang. *Minors of graphs of large path-width*. Ph.D. thesis, Georgia Institute of Technology, 2018. `http://aco.gatech.edu/sites/default/files/documents/dang-thesis.pdf`.

[9] Erik D. Demaine, Fedor V. Fomin, MohammadTaghi Hajiaghayi, and Dimitrios M. Thilikos. Bidimensional parameters and local treewidth. *SIAM J. Discrete Math.*, 18(3):501–511, 2004/05. doi: 10.1137/S0895480103433410. MR: 2134412.

[10] Erik D. Demaine and MohammadTaghi Hajiaghayi. Equivalence of local treewidth and linear local treewidth and its algorithmic applications. In *Proc. 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '04)*, pp. 840–849. SIAM, 2004. `http://dl.acm.org/citation.cfm?id=982792.982919`.

[11] Giuseppe Di Battista and Roberto Tamassia. On-line maintenance of triconnected components with SPQR-trees. *Algorithmica*, 15(4):302–318, 1996. doi: 10.1007/BF01961541.

[12] Giuseppe Di Battista and Roberto Tamassia. On-line planarity testing. *SIAM J. Comput.*, 25(5):956–997, 1996. doi: 10.1137/S0097539794280736.

[13] Reinhard Diestel, Tommy R. Jensen, Konstantin Yu. Gorbunov, and Carsten Thomassen. Highly connected sets and the excluded grid theorem. *J. Combin. Theory Ser. B*, 75(1):61–73, 1999. doi: 10.1006/jctb.1998.1862.

[14] Vida Dujmović, David Eppstein, and David R. Wood. Structure of graphs with locally restricted crossings. *SIAM J. Disc. Math.*, 31(2):805–824, 2017. doi: 10.1137/16M1062879. MR: 3639571.

[15] Vida Dujmović, Pat Morin, and David R. Wood. Layout of graphs with bounded tree-width. *SIAM J. Comput.*, 34(3):553–579, 2005. doi: 10.1137/S0097539702416141. MR: 2137079.

[16] Vida Dujmović, Pat Morin, and David R. Wood. Layered separators in minor-

closed graph classes with applications. *J. Combin. Theory Ser. B*, 127:111–147, 2017. doi: 10.1016/j.jctb.2017.05.006. MR: 3704658.

[17] Vida Dujmović, Pat Morin, and Céline Yelle. Book embeddings of graphs with bounded layered pathwidth, 2018. In preparation.

[18] David Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3-4):275–291, 2000. doi: 10.1007/s004530010020. MR: 1759751.

[19] Louis Esperet and Gwenaël Joret. Colouring planar graphs with three colours and no large monochromatic components. *Combinatorics, Probability & Computing*, 23(4):551–570, 2014. doi: 10.1017/S0963548314000170. MR: 3217360.

[20] Jörg Flum and Martin Grohe. *Parameterized complexity theory*. Springer-Verlag, 2006. doi: 10.1007/3-540-29953-X. MR: 2238686.

[21] Martin Grohe. Local tree-width, excluded minors, and approximation algorithms. *Combinatorica*, 23(4):613–632, 2003. doi: 10.1007/s00493-003-0037-9. MR: 2046826.

[22] Carsten Gutwenger and Petra Mutzel. A linear time implementation of SPQR-trees. In Joe Marks, ed., *Proc. 8th International Symposium on Graph Drawing (GD 2000)*, vol. 1984 of *Lecture Notes in Computer Science*, pp. 77–90. Springer, 2000. doi: 10.1007/3-540-44541-2_8.

[23] John E. Hopcroft and Robert E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2:135–158, 1973. doi: 10.1137/0202012.

[24] Ton Kloks. *Treewidth*. Ph.D. thesis, Utrecht University, Utrecht, the Netherlands, 1993.

[25] Andre Kündgen and Michael J. Pelsmajer. Nonrepetitive colorings of graphs of bounded tree-width. *Discrete Math.*, 308(19):4473–4478, 2008. doi: 10.1016/j.disc.2007.08.043. MR: 2433774.

[26] Alexander Leaf and Paul Seymour. Tree-width and planar minors. *J. Comb. Theory, Ser. B*, 111:38–53, 2015. doi: 10.1016/j.jctb.2014.09.003. MR: 3315599.

[27] Chun-Hung Liu and David R. Wood. Clustered coloring of graphs with bounded layered treewidth. 2018. In preparation.

[28] Saunders Mac Lane. A structural characterization of planar combinatorial graphs. *Duke Math. J.*, 3(3):460–472, 1937. doi: 10.1215/S0012-7094-37-00336-3.

[29] Bruce Reed and David R. Wood. Forcing a sparse minor. *Combin. Probab. Comput.*, 25:300–322, 2016. doi: 10.1017/S0963548315000073.

[30] Neil Robertson and Paul Seymour. Graph minors. I. Excluding a forest. *J. Combin. Theory Ser. B*, 35(1):39–61, 1983. doi: 10.1016/0095-8956(83)90079-5. MR: 723569.

[31] Neil Robertson and Paul Seymour. Graph minors. V. Excluding a planar graph. *J. Combin. Theory Ser. B*, 41(1):92–114, 1986. doi: 10.1016/0095-8956(86)90030-4. MR: 0854606.

[32] Neil Robertson, Paul Seymour, and Robin Thomas. Quickly excluding a planar graph. *J. Combin. Theory Ser. B*, 62(2):323–348, 1994. doi: 10.1006/jctb.1994.1073. MR: 1305057.