Theoretical Computer Science 412 (2011) 4100–4109

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Constant-factor approximations of branch-decomposition and largest grid minor of planar graphs in $O(n^{1+\epsilon})$ time ☆

Qian-Ping Gu [a,*], Hisao Tamaki [b]

[a] School of Computing Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada

[b] Department of Computer Science, Meiji University, 1-1-1 Higashi-mita, Tamaku, Kawasaki-shi, 214-8571, Japan

ARTICLE INFO

Keywords:
Graph algorithms
Branch-decompositions
Graph minors

ABSTRACT

We give constant-factor approximation algorithms for computing the optimal branch-decompositions and largest grid minors of planar graphs. For a planar graph G with n vertices, let $\mathrm{bw}(G)$ be the branchwidth of G and $\mathrm{gm}(G)$ the largest integer g such that G has a $g \times g$ grid as a minor. Let $c \geq 1$ be a fixed integer and α, β arbitrary constants satisfying $\alpha > c+1$ and $\beta > 2c+1$. We give an algorithm which constructs in $O(n^{1+\frac{1}{c}} \log n)$ time a branch-decomposition of G with width at most $\alpha\, \mathrm{bw}(G)$. We also give an algorithm which constructs a $g \times g$ grid minor of G with $g \geq \frac{\mathrm{gm}(G)}{\beta}$ in $O(n^{1+\frac{1}{c}} \log n)$ time. The constants hidden in the Big-O notations are proportional to $\frac{c}{\alpha-(c+1)}$ and $\frac{c}{\beta-(2c+1)}$, respectively.

1. Introduction

The notions of branchwidth and branch-decompositions were introduced by Robertson and Seymour [23] in relation to the more celebrated notions of treewidth and tree-decompositions [21,22] in graph minor theory. Grid minors also play an important role in graph minor theory. All these notions have important algorithmic applications. A graph of small treewidth/branchwidth admits efficient dynamic programming algorithms for a vast class of problems on the graph [2,5]. A tree-/branch-decomposition-based dynamic programming algorithm usually runs in exponential time in the width of the tree-/branch-decomposition. Grid minors are fundamental in many algorithms studied in algorithmic graph minor theory and bidimensionality theory [9–12]. The ratio of the treewidth or branchwidth of a graph over the largest size of the grid minor of the graph typically appears in the exponent of the running time of those algorithms.

For an arbitrary graph G, the treewidth $\mathrm{tw}(G)$ of G and the branchwidth $\mathrm{bw}(G)$ of G are linearly related by inequalities $\mathrm{bw}(G) \leq \mathrm{tw}(G) + 1 \leq \lfloor \frac{3\mathrm{bw}(G)}{2} \rfloor$, and there are simple translations between tree- and branch-decompositions that prove these inequalities [23]. The problems of deciding the treewidth/branchwidth of a given graph and constructing a tree-/branch-decomposition of minimum width have a long history of research. For general graphs, the problem of deciding whether a given graph has treewidth smaller than k is NP-complete, if k is part of the input [1]. If k is upper-bounded by a constant, then both the decision problem and the optimal decomposition problem can be solved in linear time [6], although the dependency of the time on k is huge. There are exact parallels for branchwidth and branch-decompositions to these results: NP-completeness [26] and a linear-time algorithm for fixed k [7].

For some classes of graphs, however, these two types of width/decomposition problem dramatically differ in terms of known results. For example, it is easy to construct an optimal tree-decomposition of chordal graphs in polynomial time,

☆ A preliminary version of this paper appeared in (Q.P. Gu, H. Tamaki, Constant-factor approximations of branch-decomposition and largest grid minor of planar graphs in $O(n^{1+\epsilon})$ time, in: Proc. of the 2009 International Symposium on Algorithms and Computation, ISAAC 2009, 2009, pp. 984–993) [19].

* Corresponding author. Tel.: +1 778 782 6705; fax: +1 778 782 3045.
E-mail addresses: qgu@cs.sfu.ca (Q.-P. Gu), tamaki@cs.meiji.ac.jp (H. Tamaki).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2010.07.017

while it is NP-complete to decide, given a chordal graph G and a positive integer k, if the branchwidth of G is smaller than k [20]. This situation is partially reversed on planar graphs, where the decision problem for branchwidth can be solved in $O(n^2)$ time by the well-known rat-catching algorithm of Seymour and Thomas [26] and optimal branch-decompositions can be constructed in $O(n^3)$ time [15,26], while it is not known whether the problem of deciding the treewidth of a planar graph is polynomially solvable or NP-complete (the problem is widely believed NP-hard though).

Approximation algorithms for computing the width and minimum-width decompositions have also been extensively studied (see recent work [3,4] for literature). Because of the relationship between treewidth/tree-decomposition and branchwidth/branch-decomposition stated above, the approximation problems for these two types of width/decomposition are almost equivalent. For general graphs, the best known approximation factor achievable in polynomial time is $O(\sqrt{\log k})$ [13], where k is the optimal width, and constant-factor approximation algorithms take time exponential in the optimal width [3,24]. For planar graphs, the best-known approximation result for treewidth is the obvious $O(n^2 \log n)$ time 1.5-approximation algorithm, which uses the rat-catching algorithm of Seymour and Thomas [26] and a binary search. Tree-decompositions take $O(n^3)$ time for 1.5-approximation. Bodlaender, Grigoriev, and Koster give another constant-factor approximation algorithm for the treewidth of planar graphs that runs in $O(n^2 \log n)$ time but uses less memory [4].

Computing large grid minors of planar graphs is a key ingredient in algorithmic graph minor theory and bidimensionality theory [9–12]. It is not known whether a largest grid minor can be computed in polynomial time for planar graphs. For a graph G, let $\mathrm{gm}(G)$ denote the largest size of a grid minor of G, that is, the largest integer g such that G has a $g \times g$ grid minor. From the definition of branchwidth (see Section 2), $\mathrm{gm}(G) \le \mathrm{bw}(G)$. The branchwidth $\mathrm{bw}(G)$ is also upper-bounded by some function of $\mathrm{gm}(G)$. For general graphs, the known upper bound is $\mathrm{bw}(G) \le 20^{2(\mathrm{gm}(G))^5}$, while for planar graphs a linear bound $\mathrm{bw}(G) \le 4\mathrm{gm}(G)$ is known [25]. This linear bound gives an algorithm which finds a $g \times g$ grid minor with $g \ge \frac{\mathrm{gm}(G)}{4}$ for planar graphs.[1] An $O(n^2 \log n)$-time algorithm which gives the same bound of $g \ge \frac{\mathrm{gm}(G)}{4}$ is also known [4]. The inequalities $\mathrm{tw}(G) \le \lfloor \frac{3\mathrm{bw}(G)}{2} \rfloor$ and $\mathrm{bw}(G) \le 4\mathrm{gm}(G)$ give a linear bound $\mathrm{tw}(G) \le 6\mathrm{gm}(G)$ for planar graphs. This bound has been improved to $\mathrm{tw}(G) \le 5\mathrm{gm}(G)$ [14,28]. The bounds $\mathrm{bw}(G) \le 4\mathrm{gm}(G)$ and $\mathrm{tw}(G) \le 5\mathrm{gm}(G)$ for planar graphs have been exploited in many algorithms developed under bidimensionality theory, which work on a large grid minor if they find one and otherwise use a tree-/branch-decomposition of small width. In those applications, improving the coefficient in the bound is important, as it appears in the exponent of the running time of the algorithms. Recently, the present authors have improved the bound, showing that $\mathrm{bw}(G) \le 3\mathrm{gm}(G) + 1$ for planar graph G [17].[2] From the relation $\mathrm{tw}(G) \le \lfloor \frac{3\mathrm{bw}(G)}{2} \rfloor$, this gives $\mathrm{tw}(G) \le 4.5\mathrm{gm}(G) + 2$ for planar graphs, improving the previous $\mathrm{tw}(G) \le 5\mathrm{gm}(G)$ results of [14,28]. The algorithm implied by their proof, given a planar graph G on n vertices and a positive integer k, runs in $O(n^2)$ time and either finds a $k \times k$ grid minor or certifies that $\mathrm{bw}(G) \le 3k - 2$. This algorithm yields an asymptotic 3-approximation algorithm for finding the largest grid minor of planar graphs which is the best known in terms of the approximation ratio but not faster than other known constant-factor approximation algorithms. For branchwidth, the result may not have significant algorithmic consequences, because of the rat-catching algorithm of Seymour and Thomas mentioned earlier.

The purpose of this paper is to build on the ideas in [17] to develop faster constant-factor algorithms for branch-decompositions and largest grid minors of planar graphs. Our results are the fastest known constant-factor approximation algorithms for both problems. To gain in speed, we sacrifice the approximation ratio. In fact, our algorithms are parameterized and provide a trade-off between the running time and the approximation ratio. Our algorithm for grid minors actually finds a more general cylinder minor from which a grid minor can be straightforwardly derived. A $k \times k'$ cylinder is a Cartesian product of a cycle on k vertices and a path on k' vertices. Our main results are expressed in the following theorem.

Theorem 1.1. *Let $c \ge 1$ be a fixed integer, $\delta > 0$ be a constant, and $\lambda = \frac{1}{2}$ or 1. Given a planar graph G with n vertices and an integer k, there is an algorithm which in $O(n^{1+\frac{1}{c}})$ time constructs either a branch-decomposition of G with width at most $(2\lambda(c+1)+\delta)k$ or a $k \times \lceil \lambda k \rceil$ cylinder minor of G, where the constant hidden in the Big-O notation is proportional to $\frac{c}{\delta}$.*

Because a $k \times \lceil \lambda k \rceil$ cylinder has branchwidth $\min\{2\lceil \lambda k \rceil, k\}$ [17], Theorem 1.1 with $\lambda = \frac{1}{2}$, together with a binary search, implies the following result.

Theorem 1.2. *Let $c \ge 1$ be a fixed integer, $\delta > 0$ be an arbitrary constant, and $\alpha = \delta + c + 1$. Given a planar graph G with n vertices, we can in $O(n^{1+\frac{1}{c}} \log n)$ time construct a branch-decomposition of G with width at most $\alpha\,\mathrm{bw}(G)$.*

Theorem 1.2 can be readily extended to planar hypergraphs. This theorem naturally implies an $O(n^{1+\epsilon})$-time constant-factor approximation algorithm for tree-decompositions of planar graphs, with an additional multiplicative factor of 1.5.

Since a $k \times \lceil \lambda k \rceil$ cylinder has a $k \times \lceil \lambda k \rceil$ grid minor, taking $\lambda = 1$, the following result can be obtained from Theorem 1.1 and a binary search.

Theorem 1.3. *Let $c \ge 1$ be a fixed integer, $\delta > 0$ be an arbitrary constant, and $\beta = \delta + 2c + 1$. Given a planar graph G with n vertices, we can in $O(n^{1+\frac{1}{c}} \log n)$ time construct a $g \times g$ grid minor of G with $g \ge \frac{\mathrm{bw}(G)}{\beta}$.*

---

1 An $O(n^2 \log n)$ time implementation can be realized by using the rat-catching algorithm of Seymour and Thomas [26] to construct an oracle for the tangles which are required in constructing the grid minor.

2 The bound stated in [17] is $\mathrm{bw}(G) \le 3\mathrm{gm}(G) + 2$ and is improved to $\mathrm{bw}(G) \le 3\mathrm{gm}(G) + 1$ by Remark 3.1 of [18].

The next section gives preliminaries of the paper. We describe the basic approach for our algorithm in Section 3. We prove Theorem 1.1 in Section 4, assuming a main lemma, which asserts the existence of an efficient algorithm for performing a certain subtask. This main lemma is proved in Section 5. The final section concludes the paper.

2. Preliminaries

A *hypergraph* G consists of a set $V(G)$ of vertices and a set $E(G)$ of edges, where each edge e of $E(G)$ is a subset of $V(G)$ with at least two elements. For a set $E \subseteq E(G)$ of edges, let $V(E)$ denote $\cup_{e \in E} e$. A hypergraph G is a graph if $|e| = 2$ for every edge $e \in E(G)$. We say that a vertex v and an edge e are incident to each other if $v \in e$. We say that two edges e_1 and e_2 are incident to each other if $e_1 \cap e_2 \neq \emptyset$. A hypergraph H is a subgraph of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A subgraph H of G is *induced* by $E \subseteq E(G)$ if $E(H) = E$ and $V(H) = V(E)$. We denote by $G[E]$ the subgraph induced by E.

For a hypergraph G and a subset $A \subseteq E(G)$ of edges, we denote $E(G) \setminus A$ by $\overline{A}$ when G is clear from the context. A *separation* of hypergraph G is an ordered pair $(A, \overline{A})$ of subsets of $E(G)$. For each $A \subseteq E(G)$, we denote by $\partial(A)$ the vertex set $V(A) \cap V(\overline{A})$. The *order* of separation $(A, \overline{A})$ is $|\partial(A)| = |\partial(\overline{A})|$.

The notions of branchwidth and branch-decomposition were introduced by Robertson and Seymour [23]. A *branch-decomposition* of hypergraph G is a pair (ϕ, T), where T is a tree each internal node of which has degree 3 and ϕ is a bijection from the set of leaves of T to $E(G)$. Consider an edge e' of T and let L_1 and L_2 denote the sets of leaves of T in the two respective subtrees of T obtained by removing e'. We say that the separation $(\phi(L_1), \phi(L_2))$ is induced by this edge e' of T. We define the width of the branch-decomposition (ϕ, T) to be the largest order of the separations induced by edges of T. The *branchwidth* of G, denoted by $\mathrm{bw}(G)$, is the minimum width of all branch-decompositions of G. In the rest of this paper, we identify a branch-decomposition (ϕ, T) with the tree T, leaving the bijection implicit and regarding each leaf of T as an edge of G.

A walk in hypergraph G is a vertex-edge alternating sequence $v_0, e_1, v_1, \ldots, e_k, v_k$ such that $v_i \neq v_{i+1}$ for $0 \leq i < k$, $v_i \in e_i$ for $1 \leq i \leq k$ and $v_i \in e_{i+1}$ for $0 \leq i < k$. The length of a walk is the number of edges in the walk. A walk is a path if vertices $v_0, \ldots, v_k$ are distinct. We call v_0 and v_k the end vertices of the path. A cycle is a walk such that $k \geq 3$, $v_0 = v_k$ and vertices $v_1, \ldots, v_k$ are distinct. We denote by P_k the graph which is a path on k vertices (of length $k - 1$). We denote by C_k the graph which is a cycle on k vertices.

For two graphs G and H, the Cartesian product $G \times H$ is a graph with $V(G \times H) = V(G) \times V(H)$ and $E(G \times H) = \{\{(u, v), (w, v)\} | v \in V(H), \{u, w\} \in E(G)\} \cup \{\{(v, u), (v, w)\} | v \in V(G), \{u, w\} \in E(H)\}$. A $k \times k'$ grid is the Cartesian product $P_k \times P_{k'}$. A $k \times k'$ cylinder is the Cartesian product $C_k \times P_{k'}$.

The contraction of an edge e in a hypergraph G is to remove e from G, identify all vertices of e by a new vertex, and make all edges of G incident to e incident to the new vertex. A hypergraph H is a minor of hypergraph G if H is isomorphic to a hypergraph obtained from G through a (possibly empty) sequence of edge contractions and edge/vertex deletions (which take the subgraphs induced by the remaining sets of edges/vertices).

Let Σ be a fixed sphere. A set D of points in Σ is a *topological disk* of Σ if it is homeomorphic to an open disk $\{(x, y) \mid x^2 + y^2 < 1\}$ in the plane. For an open disk D, we denote by $\overline{D}$ the closure of D and by $\mathrm{bd}(D) = \overline{D} \setminus D$ the *boundary* of D.

A planar embedding of a hypergraph G is a mapping $\rho : V(G) \cup E(G) \to \Sigma \cup 2^{\Sigma}$ satisfying the following properties.

- For $u \in V(G)$, $\rho(u)$ is a point of Σ, and for distinct $u, v \in V(G)$, $\rho(u) \neq \rho(v)$.
- For each edge $e \in E(G)$, $\rho(e)$ is a topological disk of Σ, and for each vertex $u \in e$, $\rho(u)$ is on the boundary of $\rho(e)$.
- For distinct $e_1, e_2 \in E(G)$, $\overline{\rho(e_1)} \cap \overline{\rho(e_2)} = \{\rho(u) | u \in e_1 \cap e_2\}$.

A hypergraph is planar if it has a planar embedding. A plane hypergraph is a pair (G, ρ), where ρ is a planar embedding of G. We may simply use G to denote the plane hypergraph (G, ρ), leaving the embedding ρ implicit. For a plane hypergraph G, each connected component of $\Sigma \setminus (\cup_{e \in E(G)} \overline{\rho(e)})$ is a face of G. We denote by $F(G)$ the set of faces of G. We say that a vertex v (respectively, edge e) is incident to a face r if $\rho(v) \in \overline{r}$ (respectively, $\overline{\rho(e)} \cap \overline{r} \neq \emptyset$). We denote by $V(r)$ and $E(r)$ the sets of vertices and edges incident to face r, respectively. We denote by $F(e)$ and $F(v)$ the sets of faces an edge e and a vertex v incident to, respectively. We do not distinguish a vertex v (respectively, an edge e) from its embedding $\rho(v)$ (respectively, $\rho(e)$) when there is no confusion.

A plane hypergraph G is *biconnected* if, for any vertex $x \in V(G)$ and each pair of vertices $u, v \in V(G) \setminus \{x\}$, there is a path of G between u and v that does not pass through x. It suffices to prove Theorem 1.1 for biconnected plane graphs since, if a plane graph G is not biconnected, the problems of finding branch-decompositions and grid minors of G can be solved individually for each biconnected component.

Let G be a plane hypergraph. We say that a curve μ on the sphere Σ is *G-normal* if μ does not intersect with itself or any edge of G. We may use *normal* for G-normal, leaving G implicit. The length of a normal curve μ is the number of connected components of $\mu \setminus \bigcup_{v \in V(G)} \rho(v)$. A *noose* of G is a closed normal curve on Σ. A *minimum noose* satisfying certain properties is a noose with the minimum length satisfying the properties. A segment of a noose is *open* if it is homeomorphic to $\{(x, 0) | 0 < x < 1\}$ in the plane. For an open segment P, we denote by $\overline{P}$ the closure of P and the two points of $\overline{P} \setminus P$ the *end points* of P.

For vertices $u, v \in V(G)$, we define the *normal distance* $\mathrm{nd}_G(u, v)$ to be the length of the shortest normal curve between $\rho(u)$ and $\rho(v)$. We define the *normal distance* between two vertex-subsets $X, Y \subseteq V(G)$ to be $\mathrm{nd}_G(X, Y) = \min_{u\in X, v\in Y} \mathrm{nd}_G(u, v)$. We also write $\mathrm{nd}_G(X, v)$ for $\mathrm{nd}_G(X, \{v\})$ and $\mathrm{nd}_G(u, Y)$ for $\mathrm{nd}_G(\{u\}, Y)$.

Let ν be a noose of G and let R_1 and R_2 be the two open regions of the sphere separated by ν. Then, ν induces a separation $(A, \overline{A})$ of G, with $A = \{e \in E(G) \mid \rho(e) \subseteq R_1\}$ and $\overline{A} = \{e \in E(G) \mid \rho(e) \subseteq R_2\}$. We also say that noose ν induces edge subset A of G if ν induces a separation $(A, \overline{A})$ having A on one side. We call a separation or an edge subset *noose-induced* if it is induced by some noose. We say that a noose *separates* edge sets X and Y if the noose induces a separation $(A, \overline{A})$ with $X \subseteq A$ and $Y \subseteq \overline{A}$.

Let G be a plane hypergraph and let A be a noose-induced edge subset of G. We denote by $G|A$ a plane hypergraph defined by $V(G|A) = (V(G) \setminus V(A)) \cup \partial(A)$ and $E(G|A) = (E(G) \setminus E(A)) \cup \{\partial(A)\}$. Note that, in $G|A$, we replace all the hyperedges of A by a single hyperedge $\partial(A)$. Let R_1 be the region separated by the noose which induces A and containing A. We assume that the embedding of $G|A$ is naturally derived from that of G by "painting out" R_1 by a topologic disk representing the hyperedge $\partial(A)$. For a collection $\mathcal{A} = \{A_1, \ldots, A_r\}$ of mutually disjoint edge subsets of G, we denote $(\ldots(G|A_1)|\ldots)|A_r$ by $G|\mathcal{A}$.

3. Basic approach

We first give some known results on which our algorithm relies and the basic approach of our algorithm. Let G be a plane hypergraph, $(A, \overline{A})$ a noose-induced separation of G, and T_A and $T_{\overline{A}}$ branch-decompositions of $G|\overline{A}$ and $G|A$, respectively. We define $T_A + T_{\overline{A}}$ to be the tree obtained from T_A and $T_{\overline{A}}$ by first identifying the leaf of T_A and the leaf of $T_{\overline{A}}$ both corresponding to $\partial(A)$ and joining the two edges incident to these leaves into one edge and removing the identified leaves. The following lemma is straightforward from the definition of branch-decompositions.

Lemma 3.1. *Let G be a plane hypergraph, $(A, \overline{A})$ a noose-induced separation of G, and T_A and $T_{\overline{A}}$ branch-decompositions of $G|\overline{A}$ and $G|A$, respectively. Then $T_A + T_{\overline{A}}$ is a branch-decomposition of G with width $\max\{|\partial(A)|, k_A, k_{\overline{A}}\}$, where k_A is the width of T_A and $k_{\overline{A}}$ is the width of $T_{\overline{A}}$.*

We use this lemma to recursively construct branch-decompositions of a given plane hypergraph. For each recursive step, we need some known results on the branchwidth of planar hypergraphs.

Let G be a plane hypergraph with each edge of G incident to at most $k - 1$ vertices and $|V(G)| + |E(G)| = n$. Let $d > 0$ be an integer. Assume that there is an edge e_0 of G such that, for any vertex v of G, $\mathrm{nd}_G(e_0, v) \leq d$. It is proved in [17] that G has branchwidth at most $k - 1 + 2d$. It is further shown in [18] that, given such an edge e_0, a branch-decomposition of G with width at most $k - 1 + 2d$ can be constructed in $O(n)$ time.

Lemma 3.2 ([18]). *Let $k > 0$ and $d > 0$ be integers. Let G be a plane hypergraph with each edge of G incident to at most $k - 1$ vertices and $|V(G)| + |E(G)| = n$. If there is an edge e_0 such that, for any vertex v of G, $\mathrm{nd}_G(e_0, v) \leq d$, then, given e_0, a branch-decomposition of G with width at most $k - 1 + 2d$ can be constructed in $O(n)$ time.*[3]

The following lemma (see the Appendix for a self-contained proof), which is an application of a result of [17], gives a base for constructing the cylinder minors.

Lemma 3.3 ([17]). *Let G be a plane graph and let $k, k' > 0$ be integers. Let X and Y be edge sets of G satisfying the following conditions.*

(1) *Each of separations $(X, \overline{X})$ and $(Y, \overline{Y})$ is noose-induced.*
(2) *$G[Y]$ is biconnected.*
(3) $\mathrm{nd}_G(X, Y) \geq k'$.
(4) *There is no noose of G with length $< k$ that separates X and Y.*

Then G has a cylinder $C_k \times P_{k'}$ as a minor, and, given $(G|X)|Y$, such a minor can be constructed in time linear in $|V(\overline{X} \cap \overline{Y})|$.

Let G be a plane hypergraph and e_0 an edge of G. For any positive integers h and k, a collection $\mathcal{A}$ of noose-induced edge subsets of G is (k, h)-*shallowing* for (G, e_0), if it satisfies the following conditions.

(1) $e_0 \in \overline{A}$ for every $A \in \mathcal{A}$.
(2) $G|\overline{A}$ is biconnected for every $A \in \mathcal{A}$.
(3) $A \cap B = \emptyset$ for every pair of distinct elements $A, B \in \mathcal{A}$.
(4) $|\partial(A)| < k$ for every $A \in \mathcal{A}$.
(5) For each vertex v of $G|\mathcal{A}$, $\mathrm{nd}_G(e_0, v) \leq h$.

[3] Let $p > 0$ be an integer. If for any vertex v of G there is a vertex u of G such that $\mathrm{nd}_G(u, v) \leq p$, then, given such a vertex u, a branch-decomposition of G with width at most $\lceil \frac{k}{2} \rceil + 2p$ can be found in linear time [27]. In the preliminary version of this paper the algorithm of [27] was used. Because $2p$ can be as large as $k + 2d$, the algorithm of [18] gives a better upper bound on the width of the branch-decomposition found for G than that by the algorithm of [27]. We use the algorithm of [18] in this version. As a result, the approximation ratios α and β of the algorithms are improved from $\alpha > c + 1.5$ and $\beta > 2c + 1.5$ to $\alpha > c + 1$ and $\beta > 2c + 1$.

A (k, h)-shallowing collection $\mathcal{A}$ of noose-induced edge subsets is used to reduce the problem of decomposing G, via Lemma 3.1, to subproblems of decomposing $G|\overline{A}$ for each $A \in \mathcal{A}$. Based on this notion of (k, h)-shallowing collection, we give a recursive procedure used in our algorithm. In this procedure, the input k to the algorithm is global. The precise value of parameter h used in the algorithm will be specified later. For now, we simply remark that $h = O(k)$. The parameter λ is either $\frac{1}{2}$ or 1.

Procedure Branch-Grid($G|U$)
Input: A biconnected plane hypergraph $G|U$ with $\partial(U)$ specified, $|\partial(U)| < k$ and every other edge has exactly two vertices.
Output: Either a branch-decomposition of $G|U$ of width at most $k - 1 + 2h$ or a $k \times \lceil \lambda k \rceil$ cylinder minor of G.
Steps:

(1) If $\mathrm{nd}_{G|U}(\partial(U), v) \le h$ for each $v \in V(G|U)$, then apply Lemma 3.2 to find a branch-decomposition of $G|U$. Otherwise, proceed to the next step.
(2) Try to find a collection $\mathcal{A}$ of noose-induced edge subsets of $G|U$ that is (k, h)-shallowing for $(G|U, \partial(U))$. When unsuccessful, we are able to apply Lemma 3.3 to find a cylinder minor of G and terminate the algorithm, as we prove later.
If we find such a collection, proceed to the next step.
(3) For each $A \in \mathcal{A}$, call Branch-Grid($G|\overline{A}$) to construct a branch-decomposition T_A or a cylinder minor of $G|\overline{A}$.
If we find a branch-decomposition T_A for every $A \in \mathcal{A}$, apply Lemma 3.2 to $(G|U)|\mathcal{A}$ to construct a branch-decomposition T_0 of $(G|U)|\mathcal{A}$ and use Lemma 3.1 to combine these branch-decompositions T_A, $A \in \mathcal{A}$, and T_0 into a branch-decomposition T of $G|U$ and return T.

To bound the number of recursive calls in which each fixed vertex is involved in the computation of Step 2, we enforce some "progress" when we recurse on each noose-induced edge subset in $\mathcal{A}$. Let e_0 be an edge of G and let $d > 0$ be arbitrary. We say that a noose-induced edge subset A of G is *d-progressive for* (G, e_0) if it satisfies the following conditions.

(1) $e_0 \in \overline{A}$.
(2) For any vertex v of G, if $\mathrm{nd}_G(e_0, v) \le d$, then $v \in V(\overline{A})$.

We say a collection of noose-induced edge subsets is d-progressive for (G, e_0) if each of its members is d-progressive for (G, e_0). Informally, if a noose-induced edge subset A of $G|U$ is d-progressive for $(G|U, \partial(U))$ and Branch-Grid($G|U$) makes a recursive call Branch-Grid($G|\overline{A}$), then each vertex of $G|U$ gets closer to $\partial(\overline{A})$ in $(G|U)|\overline{A}$ than to $\partial(U)$ in $G|U$ by the amount of at least d, as long as it appears in $(G|U)|\overline{A}$. This is how we enforce a progress in the recursion.

4. Algorithm details

We now give the details of our algorithm, including the precise value of parameter h which depends on a positive integer c and a positive constant δ.

For a plane hypergraph G, an edge e_0 of G, and a nonnegative integer d, let

$$\mathrm{reach}_G(e_0, d) = \bigcup \{v \in V(G) | \mathrm{nd}_G(e_0, v) \le d\}$$

denote the set of vertices with normal distance d or smaller from edge e_0. Let $\lambda = \frac{1}{2}$ or 1. We define $d_1 = \delta k$, and for positive integer $i \ge 2$, $d_i = d_i(k) = d_1 + (i - 1)(\lceil \lambda k \rceil - 1)$.

Theorem 1.1 relies on the following lemma, which will be proved in Section 5, and guarantees that we can find a sufficiently shallowing and progressive collection of noose-induced edge subsets during the recursion, as long as $\mathrm{bw}(G) < k$.

Lemma 4.1. *Let $c \ge 1$ be a fixed integer and δ an arbitrary positive constant. Let G be a biconnected plane graph, k a positive integer, and U a noose-induced edge subset of G with $|\partial(U)| < k$. Let M denote the number of vertices of $G|U$ in $\mathrm{reach}_{G|U}(\partial(U), d_{c+1})$. If $\mathrm{reach}_{G|U}(\partial(U), d_{c+1}) \ne V(G|U)$, we can in $O(M^{1+\frac{1}{c}})$ time either*

(1) *find a δk-progressive and (k, d_{c+1})-shallowing collection $\mathcal{A}$ of noose-induced edge subsets for $(G|U, \partial(U))$, or*
(2) *find a $k \times \lceil \lambda k \rceil$ cylinder minor of $G|U$.*

In executing Step 2 of Procedure Branch-Grid, we invoke Lemma 4.1 and obtain a d-progressive and (k, h)-shallowing collection of noose-induced edge subsets with $d = \delta k$ and $h = d_{c+1}$. When the search for such a collection is unsuccessful, Lemma 4.1 ensures that a $k \times \lceil \lambda k \rceil$ cylinder minor of G is found. Theorem 1.1 follows from the following lemma, which in turn is proved assuming that Lemma 4.1 is true.

Lemma 4.2. *Given a biconnected plane graph G and an integer $k \ge 3$, suppose Branch-Grid($G|e_0$) is called, where e_0 is an arbitrary edge of G. The algorithm gives either a branch-decomposition of G with width at most $k - 1 + 2d_{c+1}$ or a $k \times \lceil \lambda k \rceil$ cylinder minor of G. The execution time of this call is $O(n^{1+\frac{1}{c}})$, where n is the number of vertices of G.*

Proof. When the algorithm fails in finding a δk-progressive and (k, d_{c+1})-shallowing collection of noose-induced edge subsets for $(G|U, \partial(U))$, at any point in the execution, Lemma 4.1 ensures that a $k \times \lceil \lambda k \rceil$ cylinder minor is found. Suppose that this does not happen and that the initial call returns a branch-decomposition of G. We show by induction on the recursion structure that each Branch-Grid($G|U$) call in the recursion returns a branch-decomposition of $G|U$ with width at most $k-1+2d_{c+1}$. For the base case, suppose that Branch-Grid($G|U$) does not make any recursive calls. This means that $\mathrm{nd}_{G|U}(\partial(U), v) \le d_{c+1}$ for each v of $G|U$. Moreover, $|e| < k$ for each $e \in E(G|U)$, since $|e| < k$ for each $e \in E(G)$ and $|\partial(U)| < k$ by the choice of U. Therefore, by Lemma 3.2, this call returns a branch-decomposition of width at most $k-1+2d_{c+1}$. For the induction step, suppose that Branch-Grid($G|U$) makes a recursive Branch-Grid($G|\overline{A}$) call for each $A \in \mathcal{A}$. Because $G|\overline{A}$ is biconnected, by the induction hypothesis, each call returns a branch-decomposition of $G|\overline{A}$ with width at most $k-1+2d_{c+1}$. Moreover, again by Lemma 3.2, the branch-decomposition we obtain for $(G|U)|\mathcal{A}$ has width at most $k-1+2d_{c+1}$, since $\mathcal{A}$ is (k, d_{c+1})-shallowing for $(G|U, \partial(U))$ and $(G|U)|\mathcal{A}$ is biconnected. The combined branch-decomposition for $G|U$ has width at most $k-1+2d_{c+1}$ by Lemma 3.1. This completes the induction.

We now analyze the running time. We say that a vertex v of G is involved in the call Branch-Grid($G|U$), if v is in $\mathrm{reach}_{G|U}(\partial(U), d_{c+1})$. We first observe that a vertex is involved in two Branch-Grid($G|U$) and Branch-Grid($G|U'$) calls only if these two calls are in the ancestor–descendant relationship in the tree of recursive calls, since otherwise $\overline{U}$ and $\overline{U'}$ are disjoint.

Let b be an arbitrary positive integer and suppose that there is a chain of $b+1$ calls Branch-Grid($G|U_0$), Branch-Grid($G|U_1$), ..., Branch-Grid($G|U_b$), where Branch-Grid($G|U_i$) for each $0 \le i < b$ directly calls Branch-Grid($G|U_{i+1}$). Suppose, furthermore, that a vertex v of G is involved in all of these calls. Since the collections of noose-induced edge subsets in the algorithm are δk-progressive, $\mathrm{nd}_{(..(G|U_0)..|U_i)|U_{i+1}}(\partial(U_{i+1}), v) \le \mathrm{nd}_{(..(G|U_0)..|U_i)}(\partial(U_i), v) - \delta k$, for $0 \le i < b$, and since $\mathrm{nd}_{G|U_0}(\partial(U_0), v) \le d_{c+1}$, $b \le \frac{d_{c+1}}{\delta k} = O(1)$. Therefore, each vertex of G may be involved in $O(1)$ calls. Since the running time of call Branch-Grid($G|U_i$), excluding what is spent in the recursive calls, is dominated by the $O(M^{1+\frac{1}{c}})$ time spent in the application of Lemma 4.1, where M is the number of vertices of G involved in this call, it follows that the total running time is $O(n^{1+\frac{1}{c}})$, where n is the number of vertices of G. □

5. Proof of the main lemma

In this section, we prove Lemma 4.1 to complete the proof of Theorem 1.1. We first describe the main ideas. Let G and U be as in Lemma 4.1. For each subgraph X of $G|U$ that is at distance h away from $\partial(U)$ (a precise definition is given later), we try to find a separation of order smaller than k that separates X from $\partial(U)$, using the linear-time algorithm for the vertex-disjoint paths for planar graphs [16]. If we fail to find such a separation for any X, then, by Lemma 3.3, we obtain a cylinder minor of G that certifies $\mathrm{bw}(G) \ge k$. If we do obtain separation $(A_X, \overline{A_X})$ of order smaller than k for each X that separates X from $\partial(U)$, then we hope that these edge subsets A_X constitute a (k, h)-shallowing collection for $(G|U, \partial(U))$. There are two issues to be resolved in this approach.

(1) These subsets may not be disjoint from each other as required by the definition of (k, h)-shallowing collections.
(2) Even though the algorithm for the vertex-disjoint paths runs in linear time, the computation must be repeated for each X and may result in a quadratic running time.

The first issue is resolved by Lemma 5.2. The second issue is resolved by a layered tree approach described below combined with Lemma 5.1 that helps in localizing the graph on which the vertex-disjoint path algorithm is executed. Lemma 5.1 is a modified version of Lemma 3.6 in [17]. We include a self-contained proof for this lemma here.

Lemma 5.1 (*[17]*). *Let G be a plane hypergraph and h a positive integer. Let $X \subseteq E(G)$ and $e_0 \in X$. For $i = 0, 1$ let c_i be a cycle or an edge of G such that c_0 and c_1 are edge-disjoint, and one of the regions separated by c_i contains c_{1-i} as well as all edges of X (see Fig. 1). Suppose further that, for $i = 0, 1$ and for each $v \in V(c_i)$, $\mathrm{nd}_G(e_0, v) = h$.*

Let $(A, \overline{A})$ be a noose-induced separation that satisfies the following conditions.

(1) *$E(c_0) \subseteq A$ and $X \subseteq \overline{A}$.*
(2) *$|\partial(A)|$ is the smallest subject to condition* (1).
(3) *A is minimal subject to conditions* (1) *and* (2).

Then, either $E(c_1) \subseteq A$ or $A \cap E(c_1) = \emptyset$.

Proof. We prove the lemma for the case that both c_0 and c_1 are cycles of G. The proofs for the other cases are similar and simpler. Let ν be a noose that induces separation $(A, \overline{A})$. Suppose for contradiction that $E(c_1) \not\subseteq A$ and $A \cap E(c_1) \ne \emptyset$. Then, ν must intersect at least two distinct vertices of c_1. For each $i = 0, 1$, we call the region separated by c_i that does not contain X the *interior region* of c_i. Let ν' be a maximal open segment of ν that does not intersect any vertices of c_1 or the interior region of c_1. Let the endpoints of ν' be v_1 and v_2, $v_1, v_2 \in V(c_1)$ (see Fig. 1).

For each vertex $v \in V(c_0)$, there is a vertex $u_v \in e_0$ such that $\mathrm{nd}_G(u_v, v) = h$ because $\mathrm{nd}_G(e_0, v) = h$. Let μ_v be a normal curve of length h between v and u_v. We assume without loss of generality that, for any pair of distinct vertices $v, w \in V(c_0)$, μ_v and μ_w do not intersect each other. For each $v \in V(c_0)$, let p_v denote the point at which μ_v and ν intersects; we choose the one closest to v if there are multiple intersections. Note that, because of the distance conditions, p_v lies on the segment ν' of ν. For $i = 1, 2$, let $w_i \in V(c_0)$ be the vertex such that p_{w_i} is the closest to v_i on ν' (see Fig. 1). We claim that w_1 and w_2 are adjacent on c_0. For, otherwise, there are two vertices $u_1, u_2 \in V(c_0)$ such that u_1, w_1, u_2, w_2 appear in this order on c_0, while

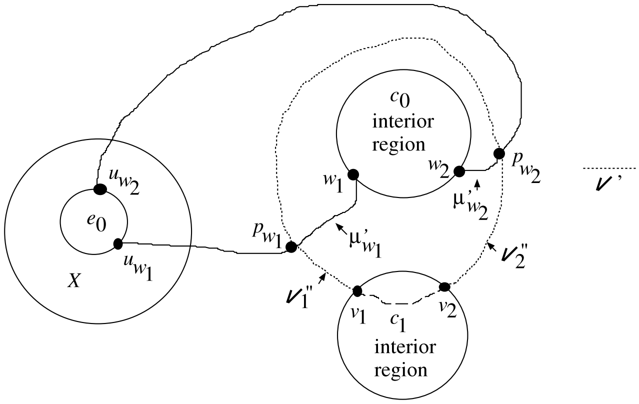

Fig. 1. Explanations on the proof of Lemma 5.1.

p_{u_1} and p_{u_2} appear between p_{w_1} and p_{w_2} on ν', and hence μ_{u_i} and μ_{w_j} must cross for at least one pair $(i, j) \in \{1, 2\} \times \{1, 2\}$, contradicting our assumption. For $i = 1, 2$, let μ'_{w_i} be the segment of μ_{w_i} between w_i and p_{w_i} and let ν''_i be the segment of ν' between v_i and p_{w_i} (see Fig. 1). Since $\mathrm{nd}_G(e_0, w_i) = \mathrm{nd}_G(e_0, v_i) = h$, the length of the normal curve ν''_i cannot be smaller than that of μ'_{w_i}. Let $\hat{\nu}$ be a noose obtained by concatenating μ'_{w_1}, the segment of ν' between p_{w_1} and p_{w_2}, μ'_{w_2} and finally a normal curve of length 1 along an edge of c_0 that closes the loop. From what we have proved on the lengths of μ'_{w_1} and μ'_{w_2}, we conclude that the length of $\hat{\nu}$ is no greater than that of ν. But then the separation induced by $\hat{\nu}$ contradicts the minimality condition (2) or (3) of A. $\square$

Lemma 5.2. *Let G be a biconnected plane hypergraph, $e_0 \in E(G)$, $k > 0$ an integer, and $0 < d < d'$ integers such that each edge in $G[V(G) \setminus \mathrm{reach}_G(e_0, d)]$ has exactly two vertices. Let $X \subseteq E(G)$ be a biconnected component of $G[V(G) \setminus \mathrm{reach}_G(e_0, d)]$ and $X_i \subseteq E(G)$, for $1 \le i \le r$, biconnected components of $G[V(G) \setminus \mathrm{reach}_G(e_0, d')]$ that are contained in X. Suppose that, for each $1 \le i \le r$, there is a separation $(A_i, \overline{A_i})$ of G with $|\partial(A_i)| < k$ such that $\overline{X} \subseteq \overline{A_i}$ and $X_i \subseteq A_i$. Then, there is a mutually disjoint collection $\mathcal{A}$ of edge subsets of G such that*

(1) *$|\partial(A)| < k$ for each $A \in \mathcal{A}$,*
(2) *$G|\overline{A}$ is biconnected for each $A \in \mathcal{A}$, and*
(3) *for each $1 \le i \le r$, there is some $A \in \mathcal{A}$ such that $X_i \subseteq A$ and $\overline{X} \subseteq \overline{A}$.*

Moreover, we can construct such a collection $\mathcal{A}$ in $O(r|X \setminus \bigcup_{1 \le i \le r} X_i|)$ time.

Proof. We first choose separation $(A_i, \overline{A_i})$ for each $1 \le i \le r$ so that it satisfies the following conditions.

(1) $X_i \subseteq A$ and $\overline{X} \subseteq \overline{A}$.
(2) $|\partial(A_i)|$ is the smallest subject to condition (1).
(3) A_i is minimal subject to conditions (1) and (2).

We first claim that, for $1 \le i, j \le r$ and $i \ne j$, either $X_j \subseteq A_i$ or $X_j \cap A_i = \emptyset$. To apply Lemma 5.1, we observe the following. Since $G[X_i]$ is biconnected for each i, either $G[X_i]$ has a cycle c_i bounding its outer face or is a single edge, which we also denote by c_i. Observe also that $\mathrm{nd}_G(e_0, v) = d' + 1$ for every $v \in c_i$ for every $1 \le i \le r$; certainly $\mathrm{nd}_G(e_0, v) \ge d' + 1$ from the definition of X_i and $\mathrm{nd}_G(e_0, v) \le d' + 1$, since v must be incident to a face which in turn is incident to a vertex v' with $\mathrm{nd}_G(e_0, v') \le d'$. Therefore, we can apply Lemma 5.1 to verify the claim.

Let I be a minimal subset of $\{1, \ldots, r\}$ such that, for every $1 \le i \le r$, there is some $j \in I$ with $X_i \subseteq A_j$. I is well defined, because $X_i \subseteq A_i$ for each i. Then, for any pair $i, j \in I$ of distinct indices, $X_i \not\subseteq A_j$ holds, and hence, by the claim above, $X_i \cap A_j = \emptyset$ holds. Based on this, we prove below that $A_i \cap A_j = \emptyset$ for each distinct $i, j \in I$ and $G|\overline{A_i}$ is biconnected for each $i \in I$. Then we are done, since $\mathcal{A} = \{A_i \mid i \in I\}$ satisfies the condition of the lemma.

Fix $i, j \in I$. For notational convenience, and without loss of generality, we assume that $i = 1$ and $j = 2$. Let ν_1 and ν_2 be nooses that induce separations $(A_1, \overline{A_1})$ and $(A_2, \overline{A_2})$, respectively. Suppose for contradiction that $A_1 \cap A_2 \ne \emptyset$. Since $1, 2 \in I$, neither A_1 nor A_2 is a subset of the other. Therefore, ν_1 and ν_2 must intersect. We count one maximal segment of ν_2 that is contained in the region separated by ν_1 and containing c_1 as one intersection between ν_1 and ν_2. We assume that the number of intersections between ν_1 and ν_2 is the smallest over all the possible choices of ν_1 and ν_2 that induce the desired separations. From the above claim and the choice of I, $A_1 \cap X_2 = \emptyset$ and $A_2 \cap X_1 = \emptyset$.

Let σ_2 be a maximal segment of ν_2 that is contained in the region separated by ν_1 and containing c_1. Let σ_1 be the segment of ν_1 that shares endpoints with σ_2 and, together with σ_2, forms a noose that separates c_2 from c_1. Let $\sigma'_1 = \nu_1 \setminus \sigma_1$. If the length of σ_2 is at most that of σ'_1, then we may replace ν_1 by the noose consisting of σ_1 and σ_2, which either contradicts the minimality of A_1 or the minimality of the number of intersections between ν_1 and ν_2. Therefore, we must have that the length of σ_2 is strictly larger than that of σ'_1. Let ν'_2 be the curve obtained from ν_2 by replacing the segment σ_2 by σ'_1.

This curve ν_2' may be self-intersecting. However, some closed subcurve of ν_2' does bound a region containing X_2, as the region separated by ν_1 and containing X_1 does not contain any edge of X_2. This subcurve is a noose of G and induces a separation that contradicts the minimality of A_2. Therefore $A_1 \cap A_2 = \emptyset$.

Let u, v, x be arbitrary distinct vertices of $V(A_i)$. Then u, v, x are also vertices of $V(X)$. Since $G[X]$ is biconnected, there is a path p between u and v in $G[X]$ that does not pass through x. Let p' be the path obtained by replacing each subpath q (possibly none) of p, where every edge of q is in $\overline{A_i}$ and the end vertices of q are in $\partial(\overline{A_i})$, by the edge $\partial(\overline{A_i})$. Then p' is a path between u and v in $G|\overline{A_i}$ that does not pass through x. That is, $G|\overline{A_i}$ is biconnected.

The mutual disjoint collection $\mathcal{A}$ can be computed as follows. We replace $\partial(\overline{X})$ and $\partial(X_i)$ in $(G|\overline{X})|X_i$ with star graphs $S_{\overline{X}}$ and S_i, respectively, where $V(S_{\overline{X}}) = \{v_{\overline{X}}\} \cup \{v|v \in \partial(\overline{X})\}$, $v_{\overline{X}}$ is a new vertex, and $E(S_{\overline{X}}) = \{\{v_{\overline{X}}, v\}|v \in \partial(\overline{X})\}$; $V(S_i) = \{v_i\} \cup \{v|v \in \partial(X_i)\}$, v_i is a new vertex, and $E(S_i) = \{\{v_i, v\}|v \in \partial(X_i)\}$. Then we compute a set of maximum number of vertex-disjoint paths $p_1, \ldots, p_r$ between $v_{\overline{X}}$ and v_i. By Menger's theorem, r is equal to the size of the minimum vertex-cut separating $v_{\overline{X}}$ and v_i. From this, there is a noose of length r that intersects exactly one vertex other than $v_{\overline{X}}$ or v_i of each path p_j $(1 \le j \le r)$. To find the desired separation, we put a pebble on a vertex of each path p_j $(1 \le j \le r)$, initially on the vertex adjacent to v_i. We say that two paths p_{j_1} and p_{j_2} are adjacent if there is a vertex of p_{j_1} and a vertex of p_{j_2} incident to a same face. For each pair of adjacent paths, we check if the pebbles on the paths are incident to a same face. If not, we move pebbles towards to $v_{\overline{X}}$ to make the two pebbles incident to a same face. The process must terminate with all pebbles on a noose in $(G|\overline{X})|X_i$ that induces the separation $(A_i, \overline{A_i})$ such that $\partial(A_i)$ is a minimum vertex-cut set separating $\partial(\overline{X})$ and $\partial(X_i)$, and A_i is minimal. The total number of moves for pebbles is bounded by the total number of vertices in paths $p_1, \ldots, p_j$.

Lemma 5.1 implies that, for any vertex $v \in \partial(A_i)$, $A_i \in \mathcal{A}$, $\text{nd}_G(e_0, v) \le d'$. This helps us to bound the running time for computing A_i. Let $\mathcal{B} = \{X_i | 1 \le i \le r\}$ and let H be the hypergraph obtained from hypergraph $(G|\overline{X})|\mathcal{B}$ with $\partial(\overline{X})$ and $\partial(X_i)$ replaced by the corresponding star graphs. From Lemma 5.1, a minimum noose in H separating $\partial(\overline{X})$ and $\partial(X_i)$ is also a minimum noose in G separating $\overline{X}$ and X_i. Hypergraph H is planar and $|V(H)| \le M$, where M is the number of vertices of G in $\text{reach}_G(e_0, d')$. It is shown in [16] that the maximum vertex-disjoint paths problem on H can be converted in linear time to the maximum edge-disjoint paths problem on a planar graph $M(H)$, called the *medial graph* [26], derived from H. Since $|V(M(H))| = O(|V(H)|)$ [26] and the maximum edge-disjoint paths problem in planar graphs can be solved in linear time [8,29], the maximum vertex-disjoint paths problem and thus the separation $(A_i, \overline{A_i})$ can be computed in $O(M)$ time. Because this computation may be repeated r times, the time for computing $\mathcal{A}$ is $O(r|X \setminus \bigcup_{1 \le i \le r} X_i|)$. □

Let G and e_0 be as above. The *layer tree* for (G, e_0), denoted by $\text{LT}(G, e_0)$, is defined as follows. Recall that $d_1 = \delta k$, and for positive integer $i \ge 2$, $d_i = d_i(k) = d_1 + (i-1)(\lceil \lambda k \rceil - 1)$.

(1) The root of the tree is $V(G)$.
(2) Each biconnected component of $G[V(G) \setminus \text{reach}_G(e_0, d_1)]$ is in level 1 of the tree and is a child node of the root.
(3) For each i, $2 \le i \le c$, each biconnected component X of $G[V(G) \setminus \text{reach}_G(e_0, d_i)]$ is in level i of the tree and is a child node of the connected component of $G[V(G) \setminus \text{reach}_G(e_0, d_{i-1})]$ that contains X.

Let X be a non-root node of $\text{LT}(G, e_0)$ and Y a child node of X. Since $\text{nd}_G(\overline{X}, Y) \ge d_i - d_{i-1} + 1 \ge \lceil \lambda k \rceil$ and $G[Y]$ is biconnected, we can apply Lemma 3.3 to obtain a noose-induced edge subset A with $\partial(A) < k$ separating $\overline{X}$ and Y, assuming that $\text{bw}(G) < k$. Our strategy is to find at least one such noose along the path from each leaf in level $c+1$ to the root.

Let m denote the number of leaves in level $c+1$ of $\text{LT}(G, e_0)$. We classify nodes of $\text{LT}(G, e_0)$ as *crowded* or *uncrowded* by induction on its tree structure. We classify each leaf in level $c+1$ as crowded. We classify a node in other levels as crowded if it has more than $m^{\frac{1}{c}}$ crowded child nodes. Otherwise it is uncrowded. We call a parent–child pair (X, Y) *processable* if X is uncrowded, Y is crowded, and no ancestor of X is crowded.

Lemma 5.3. *In* $\text{LT}(G, e_0)$*, for every leaf* Z *in level* $c+1$*, the path from the root to* Z *contains exactly one processable parent–child pair.*

Proof. Observe that each node in the first level is uncrowded, since otherwise it would have more than m descendants in level $c+1$. The lemma immediately follows from this observation and the definition of processable pairs. □

To "process" a parent–child pair (X, Y), that is, to find a minimum noose separating $\overline{X}$ and Y, we solve the maximum vertex-disjoint paths problem for planar hypergraphs as we described in the proof of Lemma 5.2. To localize the problem, we define a hypergraph $H(X, Y)$ for each parent–child pair X, Y in $\text{LT}(G, e_0)$ as follows. Let i be the level of node X in $\text{LT}(G, e_0)$. Then, $H(X, Y) = (G|\overline{X})|\mathcal{B}$, where $\mathcal{B}$ is the collection of biconnected components of $X \setminus \text{reach}_G(e_0, d_{i+1})$. Let ν be a minimum noose in G separating $\overline{X}$ and a $Y \in \mathcal{B}$ in G, and suppose that $|\nu| < k$. By Lemma 5.1, ν is also a minimum noose separating the edge $\partial(\overline{X})$ and the edge $\partial(Y)$ in $H(X, Y)$. Thus, running the algorithm for the maximum vertex-disjoint paths problem in $H(X, Y)$ is sufficient for deciding whether there is a noose of G of length smaller than k separating $\overline{X}$ and Y and, if there is, finding such a noose.

Lemma 5.4. *Let* G *be a biconnected plane graph and* U *a noose-induced edge subset of* G*. Let* M *denote the number of vertices of* $G|U$ *in* $\text{reach}_{G|U}(\partial(U), d_{c+1})$*. Then, in* $O(M^{1+\frac{1}{c}})$ *time, we can either find a collection* $\mathcal{A}$ *of noose-induced edge subsets of* G *satisfying the following conditions or a* $k \times \lceil \lambda k \rceil$ *cylinder minor of* G*.*

(1) $\mathrm{reach}_{G|U}(\partial(U), d_1) \subseteq \overline{A}$ *for each* $A \in \mathcal{A}$.
(2) $G|\overline{A}$ *is biconnected for each* $A \in \mathcal{A}$.
(3) *For any* $A, B \in \mathcal{A}$, *either* $A \subseteq B$, $B \subseteq A$, *or* $A \cap B = \emptyset$.
(4) $|\partial(A)| < k$ *for each* $A \in \mathcal{A}$.
(5) *For each biconnected component* Z *of* $V(G|U) \setminus \mathrm{reach}_{G|U}(\partial(U), d_{c+1})$, *there is* $A \in \mathcal{A}$ *with* $Z \subseteq A$.

Proof. We process each processable pair in the layer tree $\mathrm{LT}(G|U, \partial(U))$. Let v be an arbitrary vertex of $G|U$ and let i be such that $d_i < \mathrm{nd}_{G|U}(\partial(U), v) \le d_{i+1}$. Let W (respectively, Z) be the only node of $\mathrm{LT}(G|U, \partial(U))$ at level $i-1$ (respectively, level i) such that $v \in V(W)$ (respectively, $v \in V(Z)$). Then, v is a vertex of $H(X, Y)$ for a parent–child pair X, Y in $\mathrm{LT}(G|U, \partial(U))$ only if $X = W$ or $X = Z$. Therefore, there are at most $2m^{\frac{1}{c}}$ processable pairs (X, Y) for which v is in $H(X, Y)$, where $m \le M$ is the number of leaves in level $c+1$ of $\mathrm{LT}(G|U, \partial(U))$. The time for processing pair (X, Y) is dominated by the time $O(M_{X,Y})$, where $M_{X,Y}$ is the number of vertices of $H(X, Y)$, for computing the maximum vertex-disjoint paths as described in the proof of Lemma 5.2. We conclude that the total time for processing all processable pairs is $\sum_{(X,Y):\text{processable}} O(M_{X,Y}) = O(M^{1+\frac{1}{c}})$. If for any pair the minimum noose found has length k or larger, then by Lemma 3.3 we find a $k \times \lceil \lambda k \rceil$ cylinder minor. If this does not happen, then there is a noose of length smaller than k that separates $\overline{X}$ and Y for each processable pair (X, Y). From Lemmas 5.2 and 5.3, and the definition of the layer tree $\mathrm{LT}(G|U, \partial(U))$, we can find a desired collection of noose-induced edge subsets. □

We are now ready to prove Lemma 4.1 (the main lemma).

Proof. Given G and U, assume that $\mathrm{bw}(G) < k$ and let $\mathcal{A}$ denote the collection of noose-induced edge subsets of G obtained by Lemma 5.4.

We show that $\mathcal{A}$ is (k, d_{c+1})-shallowing and (δk)-progressive for $(G|U, \partial(U))$. Let v be an arbitrary vertex of $G|U$. Suppose that v is in $(G|U)|\mathcal{A}$. The construction of $\mathcal{A}$ ensures that the shortest normal curve from $\partial(U)$ to v in $G|U$ remains in $(G|U)|\mathcal{A}$, and therefore we have $\mathrm{nd}_{(G|U)|\mathcal{A}}(\partial(U), v) \le \mathrm{nd}_{G|U}(\partial(U), v) \le d_{c+1}$. From this and Lemma 5.4, $\mathcal{A}$ is (k, d_{c+1})-shallowing for $(G|U, \partial(U))$.

Let $A \in \mathcal{A}$ be arbitrary and let v be an arbitrary vertex of $G|U$. If $\mathrm{nd}_{G|U}(\partial(U), v) \le \delta k$ then $v \notin V(A)$. Suppose that v is a vertex of $V(A)$. Let u be a vertex of $\partial(U)$ such that $\mathrm{nd}_G(\partial(U), v) = \mathrm{nd}_G(u, v)$ and let w be the vertex of $\partial(\overline{A})$ on the shortest normal curve from u to v in $G|U$. Then, from $\mathrm{nd}_{G|U}(u, w) \ge \delta k$,

$$\begin{aligned}\mathrm{nd}_{(G|U)|\overline{A}}(\partial(\overline{A}), v) &\le \mathrm{nd}_{(G|U)|\overline{A}}(w, v)\\ &= \mathrm{nd}_{(G|U)}(u, v) - \mathrm{nd}_{(G|U)}(u, w)\\ &\le \mathrm{nd}_{G|U}(\partial(U), v) - \delta k.\end{aligned}$$

This shows that A is δk-progressive for $(G|U, \partial(U))$. □

6. Concluding remarks

We have given $O(n^{1+\epsilon})$-time constant-factor approximation algorithms for the optimal branch-decompositions and largest grid minors of planar graphs. It is interesting to develop linear-time constant-factor approximation algorithms for these problems. Another interesting open problem is that, given a graph G and an integer k, it is not known whether $\mathrm{gm}(G) \ge k$ can be decided in polynomial time even for planar G. An optimal branch-decomposition of a planar graph can be computed in $O(n^3)$ time [15,26]. It is worth developing $o(n^3)$-time exact algorithms for the optimal branch decompositions of planar graphs. It is also interesting to develop efficient $(1+\epsilon)$-approximation algorithms for the optimal branch-decompositions and largest grid minors of planar graphs.

Appendix

We give a proof for Lemma 3.3. Readers may refer to [17] for more details.

Lemma 3.3 ([17]). *Let* G *be a plane graph and let* $k, k' > 0$ *be integers. Let* X *and* Y *be edge sets of* G *satisfying the following conditions.*

(1) *Each of separations* $(X, \overline{X})$ *and* $(Y, \overline{Y})$ *is noose-induced.*
(2) $G[Y]$ *is biconnected.*
(3) $\mathrm{nd}_G(X, Y) \ge k'$.
(4) *There is no noose of* G *with length* $< k$ *that separates* X *and* Y.

Then G *has a cylinder* $C_k \times P_{k'}$ *as a minor, and, given* $(G|X)|Y$, *such a minor can be constructed in time linear in* $|V(\overline{X} \cap \overline{Y})|$.

Proof. Let ν be the noose which induces the edge subset Y and $Y(\nu) \subseteq Y$ such that each edge of $Y(\nu)$ is incident with a face intersected by ν. Because $G[Y]$ is biconnected, $Y(\nu)$ forms a cycle. Let H be the plane graph obtained from the plane graph G with sets X and $Y \setminus Y(\nu)$ removed. Let f_1 and f_2 be the faces in H for which the corresponding disks in G contain X and Y, respectively. Then, from the third condition of the lemma, for any $u \in V(f_1)$ and $v \in V(f_2)$, the normal distance between u and v in H is at least $k' - 1$ (i.e., $\mathrm{nd}_H(u, v) \geq k' - 1$). Let $F(H)$ be the set of faces of H, $R_0 = \{f_1\}$ and $V_0 = V(f_1)$. For $i \geq 1$, define a set $R_i \subseteq F(H)$ of faces and a set $V_i \subseteq V(G)$ of vertices inductively as follows.

(1) R_i is the set of faces incident with a vertex in V_{i-1} but not belonging to R_j for any $j < i$.
(2) V_i is the set of vertices incident with a face in R_i but not belonging to V_j for any $j < i$.

Let ρ be the embedding of H. For $i \geq 0$, let $H[V_i]$ be the plane graph induced by V_i from H (the embedding ρ_i for $H[V_i]$ is obtained from ρ for H by restricting it to the vertices and edges of $H[V_i]$). Note that each connected component of $H[\bigcup_{j>i} V_j]$ must lie, under embedding ρ, in a single face of $H[V_i]$.

We identify k' vertex-disjoint cycles $c_0, \ldots, c_{k'-1}$ of H in the reverse order as follows. First, we let $c_{k'-1}$ be the cycle bounding face f_2. Because of the fourth condition of the lemma, every vertex of $V(c_{k'-1})$ must belong to V_i for some $i \geq k' - 1$. Therefore, there must be a face of $H[V_{k'-2}]$ that contains $\rho(c_{k'-1})$. We let $c_{k'-2}$ be the cycle bounding this face. More generally, for $0 \leq i \leq k' - 2$, we define c_i to be the cycle bounding the face of $H[V_i]$ that contains $\rho(c_{i+1})$.

We claim that there are k vertex-disjoint paths between $V(c_0)$ and $V(c_{k'-1})$. Suppose otherwise. Then, by Menger's theorem, there must be a separation $(A, \overline{A})$ of G with order smaller than k such that $V(c_0) \subseteq V(A)$ and $V(c_{k'-1}) \subseteq V(\overline{A})$. If we choose this separation so that $\partial(A)$ is minimal, then it is induced by some noose in G. By the construction of the cycles, it follows that $X \subseteq V(A)$ and $Y \subseteq V(\overline{X})$, a contradiction. □

References

[1] S. Arnborg, D. Cornell, A. Proskurowski, Complexity of finding embedding in a k-tree, SIAM J. Discrete Math. 8 (1987) 277–284.
[2] S. Arnborg, J. Lagergren, D. Seese, Easy problems for tree-decomposable graphs, J. Algorithms 12 (1991) 308–340.
[3] E. Amir, Approximation algorithms for treewidth, Algorithmica 56 (2010) 448–479.
[4] H.L. Bodlaender, A. Grigoriev, A.M.C.A. Koster, Treewidth lower bounds with brambles, Algorithmica 51 (1) (2008) 81–98.
[5] H.L. Bodlaender, A tourist guide through treewidth, Acta Cybernet. 11 (1993) 1–21.
[6] H.L. Bodlaender, A linear time algorithm for finding tree-decomposition of small treewidth, SIAM J. Comput. 25 (1996) 1305–1317.
[7] H.L. Bodlaender, D. Thilikos, Constructive linear time algorithm for branchwidth, in: Proc. of the 24th International Colloquium on Automata, Languages, and Programming, 1997, pp. 627–637.
[8] L. Coupry, A simple linear time algorithm for the edge-disjoint (s, t)-paths problem in undirected planar graphs, Inform. Process. Lett. 64 (1997) 83–86.
[9] F. Dorn, F.V. Fomin, D.M. Thilikos, Catalan Structures and Dynamic Programming in H-minor-free graphs, in: Proc. of the 2008 Symposium on Discrete Algorithms, SODA 2008, 2008, pp. 631–640.
[10] E.D. Demaine, M.T. Hajiaghayi, Graphs excluding a fixed minor have grids as large as treewidth, with combinatorial and algorithmic applications through bidimensionality, in: Proc. of the 2005 Symposium on Discrete Algorithms, SODA 2005, 2005, pp. 682–689.
[11] E.D. Demaine, M.T. Hajiaghayi, Bidimensionality, map graphs, and grid minors, arXiv:Computer Science, DM/052070, v1, 2005.
[12] E.D. Demaine, M.T. Hajiaghayi, K. Kawarabayashi, Algorithmic graph minor theory: decomposition, approximation, and coloring, in: Proc. of the 2005 IEEE Symposium on Foundation of Computer Science, FOCS 2005, 2005, pp. 637–646.
[13] U. Feige, M.T. Hajiaghayi, J.R. Lee, Improved approximation algorithms for minimum weight vertex separators, SIAM J. Comput. 38 (2) (2008) 629–657.
[14] A. Grigoriev, Tree-width and large grid minors in planar graphs, (2008) (submitted for publication).
[15] Q.P. Gu, H. Tamaki, Optimal branch decomposition of planar graphs in $O(n^3)$ time, ACM Trans. Algorithms 4 (3) (2008) 1–13. article No. 30.
[16] Q.P. Gu, H. Tamaki, Efficient reduction of vertex-disjoint Menger problem to edge-disjoint Menger problem in undirected planar graphs, Technical Report, SFU-CMPT-TR 2009-11, May 2009.
[17] Q.P. Gu, H. Tamaki, Improved bound on the planar branchwidth with respect to the largest grid minor size, Technical Report, SFU-CMPT-TR 2009-17, July 2009.
[18] Q.P. Gu, H. Tamaki, A radius-based linear-time-constructive upper bound on the branchwidth of planar hypergraphs, Technical Report, SFU-CMPT-TR 2009-21, Nov. 2009.
[19] Q.P. Gu, H. Tamaki, Constant-factor approximations of branch-decomposition and largest grid minor of planar graphs in $O(n^{1+\epsilon})$ time, in: Proc. of the 2009 International Symposium on Algorithms and Computation, ISAAC 2009, 2009, pp. 984–993.
[20] T. Kloks, J. Kratochvíl, H. Müller, Computing the branchwidth of interval graphs, Discrete Appl. Math. 145 (2) (2005) 266–275.
[21] N. Robertson, P.D. Seymour, Graph minors I. Excluding a forest, J. Combin. Theory Ser. B 35 (1983) 39–61.
[22] N. Robertson, P.D. Seymour, Graph minors II. Algorithmic aspects of tree-width, J. Algorithms 7 (1986) 309–322.
[23] N. Robertson, P.D. Seymour, Graph minors X. Obstructions to tree decomposition, J. Combin. Theory Ser. B 52 (1991) 153–190.
[24] N. Robertson, P.D. Seymour, Graph minors XIII. The disjoint paths problem, J. Combin. Theory Ser. B 63 (1995) 65–110.
[25] N. Robertson, P.D. Seymour, R. Thomas, Quickly excluding a planar graph, J. Combin. Theory Ser. B 62 (1994) 323–348.
[26] P.D. Seymour, R. Thomas, Call routing and the ratcatcher, Combinatorica 14 (2) (1994) 217–241.
[27] H. Tamaki, A linear time heuristic for the branch-decomposition of planar graphs, in: Proc. of ESA2003, 2003, pp. 765–775.
[28] R. Thomas, Tree decompositions of graphs, p. 3, 2009. www.math.gatech.edu/thomas/SLIDE/slide.ps.
[29] K. Weihe, Edge-disjoint (s, t)-paths in undirected planar graphs in linear time, J. Algorithms 23 (1997) 121–138.