1.5-Approximation for Treewidth of Graphs Excluding a Graph with One Crossing as a Minor*

Erik D. Demaine[1], MohammadTaghi Hajiaghayi[1], and Dimitrios M. Thilikos[2]

[1] Laboratory for Computer Science, Massachusetts Institute of Technology, 200 Technology Square, Cambridge, MA 02139, U.S.A.
[2] Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya,
Campus Nord – Mòdul C5, Desp. 211b, c/Jordi Girona Salgado, 1-3. E-08034, Barcelona, Spain

Abstract. We give polynomial-time constant-factor approximation algorithms for the treewidth and branchwidth of any H-minor-free graph for a given graph H with crossing number at most 1. The approximation factors are 1.5 for treewidth and 2.25 for branchwidth. In particular, our result directly applies to classes of nonplanar graphs such as K_5-minor-free graphs and $K_{3,3}$-minor-free graphs. Along the way, we present a polynomial-time algorithm to decompose H-minor-free graphs into planar graphs and graphs of treewidth at most c_H (a constant dependent on H) using clique sums. This result has several applications in designing fully polynomial-time approximation schemes and fixed-parameter algorithms for many NP-complete problems on these graphs.

1 Introduction

Treewidth plays an important role in the complexity of several problems in graph theory. The notion was first defined by Robertson and Seymour in [RS84] and served as one of the cornerstones of their lengthy proof of the Wagner conjecture, now known as the Graph Minors Theorem. (For a survey, see [RS85].) Treewidth also has several applications in algorithmic graph theory. In particular, a wide range of otherwise-intractable combinatorial problems are polynomially solvable, often linearly solvable, when restricted to graphs of bounded treewidth [ACP87,Bod93].

Roughly speaking, the *treewidth* of a graph is the minimum k such that the graph can be "decomposed" into a tree structure of bags, with each vertex of graph spread over a connected subtree of bags, so that edges only connect two

* The work of the third author was supported by the IST Programme of the EU under contract number IST-1999-14186 (ALCOM-FT), the Spanish CICYT project TIC2000-1970-CE, and the Ministry of Education and Culture of Spain (Resolución 31/7/00 – BOE 16/8/00). Emails: {hajiagha,edemaine}@theory.lcs.mit.edu, and sedthilk@lsi.upc.es

vertices occupying a common bag, and at most $k+1$ vertices occupy each bag. (For the precise definition, see Section 2.)

Much research has been done on computing and approximating the treewidth of a graph. Computing treewidth is NP-complete even if we restrict the input graph to graphs of bounded degree [BT97], cocomparability graphs [ACP87,HM94], bipartite graphs [Klo93], or the complements of bipartite graphs [ACP87]. On the other hand, treewidth can be computed exactly in polynomial time for chordal graphs, permutation graphs [BKK95], circular-arc graphs [SSR94], circle graphs [Klo93], and distance-hereditary graphs [BDK00].

From the approximation viewpoint, Bodlaender et al. [BGHK95] gave an $O(\log n)$-approximation algorithm for treewidth on general graphs. A famous open problem is whether treewidth can be approximated within constant factor. Treewidth can be approximated within constant factor on AT-free graphs [BT01] (see also [BKMT]) and on planar graphs. The approximation for planar graphs is a consequence of the polynomial-time algorithm given by Seymour and Thomas [ST94] for computing the parameter *branchwidth*, whose value approximates treewidth within a factor of 1.5. To our knowledge, until now it remained open whether treewidth could be approximated within a constant factor for other kinds of graphs.

In this paper, we make a significant step in this direction. We prove that, if H is a graph that can be drawn in the plane with a single crossing (a *single-crossing* graph), then there is a polynomial-time algorithm that computes the treewidth of any H-minor-free graph. The two simplest examples of such graph classes are K_5-minor-free graphs and $K_{3,3}$-minor-free graphs.

Our result is based on a structural characterization of the graphs excluding a single-crossing graph as a minor. This characterization allows us to decompose such a graph into planar graphs and graphs of small treewidth according to clique sums. This decomposition theorem is a generalization of the current decomposition results for graphs excluding special single-crossing graph such as $K_{3,3}$ [Asa85] and K_5 [KM92]. We also show how this decomposition can be computed in polynomial time.

Our decomposition theorem has two main applications. First, we show how the tree decomposition and treewidth of each component in the decomposition can be combined in order to obtain an approximation for the whole input graph. Second, we show how the constructive decomposition can be applied to obtain fully polynomial-time schemes and fixed-parameter algorithms for a wide variety of NP-complete problems on these graphs.

This paper is organized as follows. First, in Section 2, we introduce the terminology used throughout the paper, and formally define the parameters treewidth and branchwidth. In Section 3, we introduce the concept of clique-sum graphs and prove several results on the structure of graphs excluding single-crossing graphs as minors. The main approximation algorithm is described in Section 4. In Section 5, we present several applications of clique-sum decompositions in designing algorithms for these graphs. Finally, in Section 6, we conclude with some remarks and open problems.

2 Background

2.1 Preliminaries

All the graphs in this paper are undirected without loops or multiple edges. The reader is referred to standard references for appropriate background [BM76].

Our graph terminology is as follows. A graph G is represented by $G = (V, E)$, where V (or $V(G)$) is the set of vertices and E (or $E(G)$) is the set of edges. We denote an edge e between u and v by $\{u, v\}$. We define n to be the number of vertices of a graph when this is clear from context.

The *(disjoint) union* of two disjoint graphs G_1 and G_2, $G_1 \cup G_2$, is the graph G with merged vertex and edge sets: $V(G) = V(G_1) \cup V(G_2)$ and $E(G) = E(G_1) \cup E(G_2)$.

One way of describing classes of graphs is by using *minors*, introduced as follows. *Contracting* an edge $e = \{u, v\}$ is the operation of replacing both u and v by a single vertex w whose neighbors are all vertices that were neighbors of u or v, except u and v themselves. A graph G is a *minor* of a graph H if H can be obtained from a subgraph of G by contracting edges. A graph class $\mathcal{C}$ is a *minor-closed* class if any minor of any graph in $\mathcal{C}$ is also a member of $\mathcal{C}$. A minor-closed graph class $\mathcal{C}$ is H*-minor-free* if $H \notin \mathcal{C}$.

For example, a planar graph is a graph excluding both $K_{3,3}$ and K_5 as minors.

2.2 Treewidth

The notion of treewidth was introduced by Robertson and Seymour [RS86] and plays an important role in their fundamental work on graph minors. To define this notion, first we consider the representation of a graph as a tree, which is the basis of our algorithms in this paper. A *tree decomposition* of a graph $G = (V, E)$, denoted by $TD(G)$, is a pair (χ, T) in which $T = (I, F)$ is a tree and $\chi = \{\chi_i | i \in I\}$ is a family of subsets of $V(G)$ such that: (1) $\bigcup_{i \in I} \chi_i = V$; (2) for each edge $e = \{u, v\} \in E$ there exists an $i \in I$ such that both u and v belong to χ_i; and (3) for all $v \in V$, the set of nodes $\{i \in I | v \in \chi_i\}$ forms a connected subtree of T. To distinguish between vertices of the original graph G and vertices of T in $TD(G)$, we call vertices of T *nodes* and their corresponding χ_i's *bags*. The maximum size of a bag in $TD(G)$ minus one is called the *width* of the tree decomposition. The *treewidth* of a graph G ($\mathrm{tw}(G)$) is the minimum width over all possible tree decompositions of G.

2.3 Branchwidth

A *branch decomposition* of a graph G is a pair (T, τ), where T is a tree with vertices of degree 1 or 3 and τ is a bijection from the set of leaves of T to $E(G)$. The *order* of an edge e in T is the number of vertices $v \in V(G)$ such that there are leaves t_1, t_2 in T in different components of $T(V(T), E(T) - e)$ with $\tau(t_1)$ and $\tau(t_2)$ both containing v as an endpoint. The *width* of (T, τ) is the maximum order over all edges of T, and the *branchwidth* of G is the minimum width over all branch decompositions of G. The following result implies that branchwidth is a 1.5-approximation on treewidth:

Theorem 1 ([RS91], Section 5). *For any graph G with m edges, there exists an $O(m^2)$-time algorithm that*

1. *given a branch decomposition (T, τ) of G of width $\leq k+1$, constructs a tree decomposition (χ, T) of G that has width $\leq \frac{3}{2}k$; and*
2. *given a tree decomposition (χ, T) of G that has treewidth $k+1$, constructs a branch decomposition (T, τ) of G of width $\leq k$.*

While the complexity of treewidth on planar graphs remains open, the branchwidth of a planar graph can be computed in polynomial time:

Theorem 2 ([ST94], Sections 7 and 9). *One can construct an algorithm that, given a planar graph G,*

1. *computes in $O(n^3)$ time the branchwidth of G; and*
2. *computes in $O(n^5)$ time a branch decomposition of G with optimal width.*

Combining Theorems 1 and 2, we obtain a polynomial-time 1.5-approximation for treewidth in planar graphs:

Theorem 3. *One can construct an algorithm that, given a planar graph G,*

1. *computes in $O(n^3)$ time a value k with $k \leq \mathrm{tw}(G) + 1 \leq \frac{3}{2}k$; and*
2. *computes in $O(n^5)$ time a tree decomposition of G with width k.*

This approximation algorithm will be one of two "base cases" in our development in Section 4 of a 1.5-approximation algorithm for nonplanar graphs excluding a single-crossing graph as a minor.

3 Computing Clique-Sum Decompositions for Graphs Excluding a Single-Crossing-Graph Minor

This section describes the general framework of our results, using the key tool of *clique-sums*; see [HNRT01,Haj01].

3.1 Clique Sums

Suppose G_1 and G_2 are graphs with disjoint vertex-sets and $k \geq 0$ is an integer. For $i = 1, 2$, let $W_i \subseteq V(G_i)$ form a clique of size k and let G'_i $(i = 1, 2)$ be obtained from G_i by deleting some (possibly no) edges from $G_i[W_i]$ with both endpoints in W_i. Consider a bijection $h : W_1 \to W_2$. We define a *k-sum* G of G_1 and G_2, denoted by $G = G_1 \oplus_k G_2$ or simply by $G = G_1 \oplus G_2$, to be the graph obtained from the union of G'_1 and G'_2 by identifying w with $h(w)$ for all $w \in W_1$. The images of the vertices of W_1 and W_2 in $G_1 \oplus_k G_2$ form the *join set*.

In the rest of this section, when we refer to a vertex v of G in G_1 or G_2, we mean the corresponding vertex of v in G_1 or G_2 (or both). It is worth mentioning that $\oplus$ is not a well-defined operator and it can have a set of possible results. See Figure 1 for an example of a 5-sum operation.

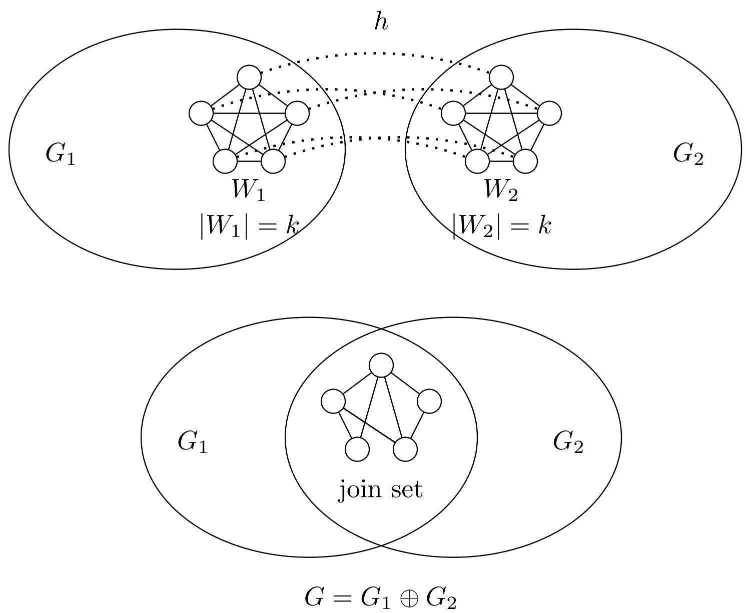

Fig. 1. Example of 5-sum of two graphs.

3.2 Connection to Treewidth

The following lemma shows how the treewidth changes when we apply a clique-sum operation, which will play an important role in our approximation algorithms in Section 4. This lemma is mentioned in [HNRT01] without proof. For the sake of completeness, we present the proof here.

Lemma 1 ([HNRT01]). *For any two graphs G and H,* $\mathrm{tw}(G \oplus H) \leq \max\{\mathrm{tw}(G), \mathrm{tw}(H)\}$.

Proof. Let W be the set of vertices of G and H identified during the $\oplus$ operation. Since W is a clique in G, in every tree decomposition of G, there exists a node α such that W is a subset of χ_α [BM93]. Similarly, the same is true for W and a node α' of each tree decomposition of H. Hence, we can construct a tree decomposition of G and a tree decomposition of H and add an edge between α and α'. □

3.3 Computing Clique-Sum Decompositions

The main theorem of this section is an algorithmic version of the following existential theorem of Robertson and Seymour:

Theorem 4 ([RS93]). *For any single-crossing graph H, there is an integer $c_H \geq 4$ (depending only on H) such that every graph with no minor isomorphic to H can be obtained by* 0-, 1-, 2- *or* 3-*sum of planar graphs and graphs of treewidth at most c_H.*

We use Theorem 4 in our constructive algorithm. For a graph $G = (V, E)$, we call a subset S of $V(G)$ a *k-cut* if the induced subgraph $G[V - S]$ is disconnected and $|S| = k$. A k-cut is *strong* if $G - S$ has more than two connected components, or it has two connected components and each component has more than one vertex. This definition is less strict than the notion of strong cuts introduced in [KM92], where a similar (but consequently weaker) version of Lemma 2 is obtained. Let $S \subseteq V(G)$ be a cut that separates G into $h \geq 2$ components $G_1, \cdots, G_h$. For $1 \leq i \leq h$, we denote by $G_i \cup K(C)$ the graph obtained from $G[V(G_i) \cup C]$ by adding an edge between any pair of nonadjacent vertices in C. The graphs $G_i \cup K(C)$, $1 \leq i \leq k$, are called the *augmented components* induced by C. Theorem 5 below describes a constructive algorithm to obtain a clique-sum decomposition of a graph G as in Theorem 4 with the additional property that the decomposition graphs are minors of the original graph G. In this sense, the result is even stronger than Theorem 4. This additional property, that each graph in the clique-sum series is a minor of the original graph, is crucial for designing approximation algorithms in the next section; Theorem 4 alone would not suffice. First we illustrate the important influence of strong cuts on augmented components:

Lemma 2. *Let C be a strong 3-cut of a 3-connected graph $G = (V, E)$, and let $G_1, G_2, \cdots, G_h$ denote the h induced components of $G[V - C]$. Then each augmented component of G induced by C, $G_i \cup K(C)$, is a minor of G.*

Proof. Suppose $C = \{x, y, z\}$. First consider the case that C disconnects the graph into at least 3 components. By symmetry, it suffices show that $G_1 \cup K(C)$ is a minor of G. Contract all edges of G_2 and G_3 to obtain super-vertices y' and z'. Because G is 3-connected, both y' and z' are adjacent to all vertices in C. Now contract edges $\{y, y'\}$ and $\{z, z'\}$ to obtain super-vertices y'' and z'', respectively. Then x, y'', and z'' form a clique, so we have arrived at the augmented component $G_1 \cup K(C)$ via contractions. Next consider the case in which $G[V - C]$ has only two components G_1 and G_2, and both have at least two vertices. Again it suffices to show that $G_1 \cup K(C)$ is a minor of G. First suppose that G_2 is a tree. Because G is 3-connected, there is a vertex x' in G_2 that neighbors x, and similarly a vertex z' in G_2 that neighbors z. Now contract every other vertex of G_2 arbitrarily to either x' or z', to obtain super-vertices x'' and z''. Because G_2 is connected, there is an edge between x'' and z''. Because G_2 has no cycle, there cannot be more than one edge between the components corresponding to super-vertices x'' and z''. Because G is 3-connected, there is an edge between y and either x'' or z'', say x''. Again because G is 3-connected, z'' is connected to a vertex of C other than z. If z'' is adjacent to y, contract the edges $\{x'', x\}$ and $\{z'', z\}$, and if z'' is adjacent to x, contract the edges $\{x'', y\}$ and $\{z'', z\}$, to form a clique on the vertices of C. Finally suppose that G_2 has a cycle C'. We claim that there are three vertex-disjoint paths connecting three vertices of C' to three vertices of C in G_2. By contracting these paths and then contracting edges of C' to form a triangle, we have a clique on the vertices of C as desired. To prove the claim, augment the graph G by adding a vertex v_1 connected to every vertex in C, and by adding a vertex v_2 connected to every vertex in C'. Because $|C| = 3$ and $|C'| \geq 3$, the augmented graph is still vertex 3-connected. Therefore there exist at least three vertex-disjoint paths from v_1 to v_2. Each of these paths must be in G_2, begin by entering a vertex of C, and end by leaving a vertex of C', and these vertices of C and C' must be different among the three paths (because they are vertex-disjoint). Thus, if we remove the first vertex v_1 and last vertex v_2 from each path, we obtain the desired paths. □

Theorem 5. *For any graph G excluding a single-crossing graph H as a minor, we can construct in $O(n^4)$ time a series of clique-sum operations $G = G_1 \oplus G_2 \oplus \cdots \oplus G_m$ where each G_i, $1 \leq i \leq m$, is a minor of G and is either a planar graph or a graph of treewidth at most c_H. Here each $\oplus$ is a 0-, 1-, 2- or 3-sum.*

Proof. The algorithm works basically as follows. Given a graph G, compute its connectivity. If it is disconnected, consider each of its connected components separately. If it has a 1-cut or 2-cut, recursively apply the algorithm on the augmented components induced by that 1-cut or 2-cut. If its connectivity is at least three, find a strong 3-cut and recursively apply the algorithm on the augmented components induced by that strong 3-cut. If the graph is 3-connected but has no strong 3-cut, then we claim that it is either planar or has treewidth at most c_H.

We first prove the correctness of the algorithm above, and later fill in the algorithmic details and analyze the running time. If G has a 1-cut or 2-cut, then each augmented component is a minor of G, and thus by Theorem 4 we can recurse on each augmented component. The same holds for strong 3-cuts if G is 3-connected, because Theorem 2 implies that the property of excluding graph H as a minor is inherited by all its augmented components. Now suppose that the graph G is 3-connected yet it has no strong 3-cut. It remains to show that either the treewidth of G is greater than c_H or that G is planar. Suppose for contradiction that neither of these properties hold. By Theorem 4, G can be obtained by 3-sums of a sequence of elementary graphs $\mathcal{C} = (J_1, \ldots, J_r)$. (Because G is 3-connected, we have no k-sums for $k \leq 2$.) We claim that one of the graphs in $\mathcal{C}$ must be a planar graph with at least five vertices. If this were not the case, then all the graphs in $\mathcal{C}$ would have treewidth at most c_H so, by Lemma 1, G would also have treewidth $\leq c_H$, which is a contradiction. Notice also that we can not have more than one graph in $\mathcal{C}$ with at least five vertices because we do not have strong 3-cuts. Therefore, $\mathcal{C}$ contains a planar graph with at least 5 vertices and all the other graphs in $\mathcal{C}$ are K_4's. We claim that G itself must be planar, establishing a contradiction. Suppose to the contrary that, during the clique-sum operations forming G, there is a 3-sum $G'' = G' \oplus K_4$ with join set C such that G' is planar but G'' is not planar. Consider a planar embedding of G'. Because C is a triangle in G' and $G' \oplus K_4$ is not planar, there are some vertices inside triangle C and some vertices outside triangle C. Thus $G'' - C$ has at least three components so C is a strong 3-cut in G''. Because G'' is a graph in the clique-sum sequence of G, C is also a strong 3-cut in G, which is again a contradiction.

To analyze the running time of the algorithm, first we claim that, for a H-minor-free graph G where H is single-crossing, we have $|E(G)| = O(|V(G)|)$.

This claim follows because the number of edges in planar graphs and graphs of treewidth at most c_H is a linear function in the number of vertices, and the total number of vertices of graphs in a clique-sum sequence forming G is linear in $|V(G)|$ (we have linear number of k-sums and $k \leq 3$). In linear time we can obtain all 1-cuts [Tar72] and we can obtain all 2-cuts using the algorithms of Hopcroft and Tarjan [HT73] or Miller and Ramachandran [MR92]. The number of 3-cuts in a 3-connected graph is $O(n^2)$ and we can obtain all 3-cuts in $O(n^2)$ time [KR91]. We can check whether each 3-cut is strong in $O(n)$ time using a depth-first search. All other operations including checking planarity and having treewidth at most c_H can be performed in linear time [Wil84,Bod96]. Now, if the algorithm makes no recursive calls, the running time of the algorithm, $T(n)$, is $O(n)$. If it makes recursive calls for a 1-cut, we have that $T(n) = T(n_1) + T(n - n_1 + 1) + O(n), \quad n_1 \geq 2$, where n_1 and $n - n_1 + 1$ are the sizes of the two augmented components. (We only split the graph into two 2-connected components at once, possibly leaving the same 1-cut for the recursive calls.) Similarly, for recursive calls for a 2-cut, we have $T(n) = T(n_1) + T(n - n_1 + 2) + O(n), \quad n_1 \geq 3$. For recursive calls for a strong 3-cut with exactly two components, we have $T(n) = T(n_1) + T(n - n_1 + 3) + O(n^3), \quad n_1 \geq 4$. Finally, if we have recursive calls for a strong 3-cut with at least three components, we have that $T(n) = T(n_1) + T(n_2) + T(n - n_1 - n_2 + 6) + O(n^3), \quad 4 \leq n_1, n_2, n - n_1 - n_2 + 6 \leq n - 2$, where n_1, n_2, and $n - n_1 - n_2 + 6$ are the sizes of the augmented components. (Again, we only split the graph into three 3-connected components, possibly leaving the same 3-cut for the recursive calls.) Solving this recurrence concludes a worst-case running time of $O(n^4)$. □

We can also parallelize this algorithm to run in $O(\log^2 n)$ time using an approach similar to that described by Kezdy and McGuinness [KM92]. The details are omitted from this paper.

3.4 Related Work

Theorems 2 and 5 generalize a characterization of $K_{3,3}$-minor-free graphs and K_5-minor-free graphs by Wagner [Wag37]. He proved that a graph has no minor isomorphic to $K_{3,3}$ if and only if it can be obtained from planar graphs and K_5 by 0-, 1-, and 2-sums. He also showed that a graph has no minor isomorphic to K_5 if and only if it can be obtained from planar graphs and V_8 by 0-, 1-, 2-, and 3-sums. Here V_8 denotes the graph obtained from a cycle of length 8 by joining each pair of diagonally opposite vertices by an edge. We note that both K_5 and V_8 have treewidth 4, i.e., $c_H = 4$. Constructive algorithms for obtaining such clique-sum series have also been developed. Asano [Asa85] showed how to construct in $O(n)$ time a series of clique-sum operations for $K_{3,3}$-minor-free graphs. Kézdy and McGuinness [KM92] presented an $O(n^2)$-time algorithm to construct such a clique-sum series for K_5-minor-free graphs.

4 Approximating Treewidth

We are now ready to prove our final result, a 1.5-approximation algorithm on treewidth:

Theorem 6. *For any single-crossing graph H, we can construct an algorithm that, given an H-minor-free graph as input, outputs in $O(n^5)$ time a tree decomposition of G of width k where* $\mathrm{tw}(G) \leq k + 1 \leq \frac{3}{2}\mathrm{tw}(G)$.

Proof. The algorithm consists of the following four steps:
Step 1: Let G be a graph excluding a single-crossing graph H. By Theorem 5, we can obtain a clique-sum decomposition $G = G_1 \oplus G_2 \oplus \cdots \oplus G_m$ where each G_i, $1 \leq i \leq m$, is a minor of G and is either a planar graph or a graph of treewidth at most c_H. According to the same theorem, this step requires $O(n^4)$ time. Let B be the set of bounded treewidth components and P be the set of planar components: $B = \{i \mid 1 \leq i \leq m,\ \mathrm{tw}(G_i) \leq c_H\}$, $P = \{1, \ldots, m\} - B$.
Step 2: By Theorem 3, we can construct, for any $i \in P$, a tree decomposition D_i of G_i with width k_i and such that

$$k_i \leq \mathrm{tw}(G_i) + 1 \leq \tfrac{3}{2}k_i \quad \text{for all } i \in P. \tag{1}$$

The construction of each of these tree decompositions requires $O(|V(G_i)|^5)$ time. As $m = O(n)$ and $\sum_{1 \leq i \leq m} |V(G_i)| = O(n)$, the total time for this step is $O(n^5)$.
Step 3: Using Bodlaender's algorithm in [Bod96], for any $i \in B$, we can obtain a tree decomposition of G_i with minimum width k_i, in linear time where the hidden constant depends only on c_H. Combining (1) with the fact that $\mathrm{tw}(G_i) = k_i$ for each $i \in B$, we obtain

$$k_i \leq \mathrm{tw}(G_i) + 1 \leq \tfrac{3}{2}k_i \quad \text{for all } i \in \{1, \ldots, m\}. \tag{2}$$

Step 4: Now that we have tree decompositions D_i of each G_i, we glue them together using the construction given in the proof of Lemma 1. In this way, we obtain a tree decomposition of G that has size $k = \max\{k_i \mid 1 \leq i \leq m\}$. Combining this equality with (2), we have

$$k \leq \max\{\mathrm{tw}(G_i) \mid i = 1, \ldots, m\} + 1 \leq \tfrac{3}{2}k. \tag{3}$$

The details of implementation of this step in $O(n^4)$ time is similar to the details described by Demaine et al. [DHT02] and hence omitted.

Finally, we prove that the algorithm is a 1.5-approximation. By Lemma 1, we have that $\mathrm{tw}(G) \leq \max\{\mathrm{tw}(G_i) \mid i = 1, \ldots, m\}$. By Theorem 5, each G_i is a minor of G and therefore $\mathrm{tw}(G_i) \leq \mathrm{tw}(G)$. Thus, $\mathrm{tw}(G) = \max\{\mathrm{tw}(G_i) \mid i = 1, \ldots, m\}$ and from (3) we conclude that $k \leq \mathrm{tw}(G) + 1 \leq \frac{3}{2}k$ and the theorem follows. □

Notice that, in the theorem above, if we just want to output the value k without the corresponding tree decomposition, then we just make use of Theorem 3(1) in Step 2 and skip Step 4, and the overall running time drops to $O(n^4)$. Using the same approach as Theorem 6, one can prove a potentially stronger theorem:

Theorem 7. *If we can compute the treewidth of any planar graph in polynomial time, then we can compute the treewidth of any H-minor-free graph, where H is single-crossing, in polynomial time.*

Proof. We just use the polynomial-time algorithm for computing treewidth of planar graphs in Step 2 of the algorithm described in the proof of Theorem 6. □

5 Other Applications of Constructing Clique-Sum Decompositions

In this section, we show that the constructive algorithm described in Section 5 has many other important applications in algorithm design for the class of graphs excluding a single-crossing graph as a minor. Roughly speaking, because both planar graphs and graphs of bounded treewidth have good algorithmic properties, clique-sum decompositions into these graphs enable the design of efficient algorithms for many NP-complete problems.

5.1 Polynomial-Time Approximation Schemes (PTASs)

Much work designs PTASs for NP-complete problems restricted to certain special graphs. Lipton and Tarjan [LT80] were the first who proved various NP-optimization problems have PTASs over planar graphs. Alon et al. [AST90] generalized Lipton and Tarjan's ideas to graphs excluding a fixed minor. Because these PTASs were impractical [CNS82], Baker [Bak94] developed practical PTASs for the problems considered by Lipton and Tarjan and Alon et al. Eppstein [Epp00] showed that Baker's technique can be extended by replacing "bounded outerplanarity" with "bounded local treewidth." Intuitively, a graph has *bounded local treewidth* if the treewidth of an r-neighborhood (all vertices of distance at most r) of each vertex $v \in V(G)$ is a function of r, $r \in \mathbb{N}$, and not the number of vertices. Unfortunately, Eppstein's algorithms are impractical for nonplanar graphs. Hajiaghayi et al. [HNRT01,Haj01] designed practical PTASs for both minimization and maximization problems on graphs excluding one of K_5 or $K_{3,3}$ as a minor, which is a special class of graphs with bounded local treewidth. Indeed, they proved the following more general theorem:

Theorem 8 ([HNRT01,Haj01]). *Given the clique-sum series of an H-minor-free graph G, where H is a single-crossing graph, there are PTASs with approximation ratio $1+1/k$ (or $1+2/k$) running in $O(c^k n)$ time (c is a small constant) on graph G for hereditary maximization problems (see [Yan78] for exact definitions) such as maximum independent set and other problems such as maximum triangle matching, maximum H-matching, maximum tile salvage, minimum vertex cover, minimum dominating set, minimum edge-dominating set, and subgraph isomorphism for a fixed pattern.*

Applying Theorem 5, we obtain the following corollary:

Corollary 1. *There are PTASs with approximation ratio $1 + 1/k$ (or $1 + 2/k$) with running time $O(c^k n + n^4)$ for all problems mentioned in Theorem 8 on any H-minor-free graph H where H is single-crossing.*

5.2 Fixed-Parameter Algorithms (FPTs)

Developing fast algorithms for NP-hard problems is an important issue. Recently, Downey and Fellows [DF99] introduced a new approach to cope with this NP-hardness, called *fixed-parameter tractability*. For many NP-complete problems, the inherent combinatorial explosion can be attributed to a certain aspect of the problem, a *parameter*. The parameter is often an integer and small in practice. The running times of simple algorithms may be exponential in the parameter but polynomial in the rest of the problem size. For example, it has been shown that k-vertex cover (finding a vertex cover of size k) has an algorithm with running time $O(kn + 1.271^k)$ [CKJ99] and hence this problem is fixed-parameter tractable. Alber et al. [ABFN00] demonstrated a solution to the planar k-dominating set in time $O(4^{6\sqrt{34k}} n)$. This result was the first nontrivial results for the parameterized version of an NP-hard problem where the exponent of the exponential term grows sublinearly in the parameter (see also [KP02] for a recent improvement of the time bound of [ABFN00] to $O(2^{27\sqrt{k}} n)$). Using this result, others could obtain exponential speedup of fixed parameter algorithms for many NP-complete problems on planar graphs (see e.g. [KC00,CKL01,KLL01,AFN01]). Recently, Demaine et al. [DHT02] extended these results to many NP-complete problems on graphs excluding either K_5 or $K_{3,3}$ as a minor. In fact, they proved the following general theorem:

Theorem 9 ([DHT02]). *Given the clique-sum series of an H-minor-free graph G, where H is a single-crossing graph, there are algorithms that in $O(2^{27\sqrt{k}} n)$ time decide whether graph G has a subset of size k dominating set, dominating set with property P, vertex cover, edge-dominating set, minimum maximal matching, maximum independent set, clique-transversal set, kernel, feedback vertex set and a series of vertex removal properties (see [DHT02] for exact definitions).*

Again, applying Theorem 5, we obtain the following corollary:

Corollary 2. *There are algorithms that in $O(2^{27\sqrt{k}} n + n^4)$ time decide whether any H-minor-free graph G, where H is single-crossing, has a subset of size k with one of the properties mentioned in Theorem 9.*

6 Conclusions and Future Work

In this paper, we obtained a polynomial-time algorithm to construct a clique-sum decomposition for H-minor-free graphs, where H is a single-crossing graph. As mentioned above, this polynomial-time algorithm has many applications in designing approximation algorithms and fixed-parameter algorithms for these kinds of graphs [Haj01,HNRT01,DHT02]. Also, using this result, we obtained a 1.5-approximation algorithm for treewidth on these graphs. Here we present several open problems that are possible extensions to this paper.

One topic of interest is finding characterization of other kinds of graphs such as graphs excluding a double-crossing graph (or a graph with a bounded number of crossings) as a minor. We suspect that we can obtain such characterizations using k-sums for $k > 3$. Designing polynomial-time algorithms to construct such decompositions would be instructive.

It would also be interesting to find other problems than those mentioned by Hajiaghayi et al. [Haj01,HNRT01,DHT02] for which the technique of obtaining clique-sum decomposition can be applied. We think that this approach can be applied for many other NP-complete problems that have good (approximation) algorithms for planar graphs and graphs of bounded treewidth.

From Theorem 2, the treewidth is a 1.5-approximation on the branchwidth. A direct consequence of this fact and our result is the existence of a 2.25-approximation for the branchwidth of the graphs excluding a single-crossing graph. One open problem is how one can use clique-sum decomposition to obtain a better approximation or an exact algorithm for the branchwidth of this graph class.

Acknowledgments

We thank Prabhakar Ragde and Naomi Nishimura for their encouragement and help on this paper.

References

[ABFN00] Jochen Alber, Hans L. Bodlaender, Henning Fernau, and Rolf Niedermeier. Fixed parameter algorithms for planar dominating set and related problems. In *Algorithm theory—Scandinavian Workshop on Algorithm Theory 2000 (Bergen, 2000)*, pages 97–110. Springer, Berlin, 2000.

[ACP87] Stefan Arnborg, Derek G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM J. Algebraic Discrete Methods*, 8(2):277–284, 1987.

[AFN01] Jochen Alber, Henning Fernau, and Rolf Niedermeier. Parameterized complexity: Exponential speed-up for planar graph problems. In *Electronic Colloquium on Computational Complexity (ECCC)*. Germany, 2001.

[Asa85] Takao Asano. An approach to the subgraph homeomorphism problem. *Theoret. Comput. Sci.*, 38(2-3):249–267, 1985.

[AST90] Noga Alon, Paul Seymour, and Robin Thomas. A separator theorem for for graphs with excluded minor and its applications. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (Baltimore, MD, 1990)*, pages 293–299, 1990.

[Bak94] Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. Assoc. Comput. Mach.*, 41(1):153–180, 1994.

[BDK00] Hajo J. Broersma, Elias Dahlhaus, and Ton Kloks. A linear time algorithm for minimum fill-in and treewidth for distance hereditary graphs. *Discrete Appl. Math.*, 99(1-3):367–400, 2000.

[BGHK95] Hans L. Bodlaender, John R. Gilbert, Hjálmtýr Hafsteinsson, and Ton Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *J. Algorithms*, 18(2):238–255, 1995.

[BKK95] Hans L. Bodlaender, Ton Kloks, and Dieter Kratsch. Treewidth and pathwidth of permutation graphs. *SIAM J. Discrete Math.*, 8(4):606–616, 1995.

[BKMT] Vincent Bouchitté, Dieter Kratsch, Haiko Mller, and Ioan Todinca. On treewidth approximations. In *Cologne-Twente Workshop on Graphs and Combinatorial Optimization (CTW'01)*.

[BM76] John A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. American Elsevier Publishing Co., Inc., New York, 1976.

[BM93] Hans L. Bodlaender and Rolf H. Möhring. The pathwidth and treewidth of cographs. *SIAM J. Discrete Math.*, 6(2):181–188, 1993.

[Bod93] Hans L. Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11:1–23, 1993.

[Bod96] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[BT97] Hans L. Bodlaender and Dimitrios M. Thilikos. Treewidth for graphs with small chordality. *Discrete Appl. Math.*, 79(1-3):45–61, 1997.

[BT01] Vincent Bouchitté and Ioan Todinca. Treewidth and minimum fill-in: grouping the minimal separators. *SIAM J. Comput.*, 31(1):212–232 (electronic), 2001.

[CKJ99] Jianer Chen, Iyad A. Kanj, and Weijia Jia. Vertex cover: further observations and further improvements. In *Graph-theoretic concepts in computer science (Ascona, 1999)*, pages 313–324. Springer, Berlin, 1999.

[CKL01] Maw-Shang Chang, Ton Kloks, and Chuan-Min Lee. Maximum clique transversals. In *Proceedings of the 27th International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 300–310. Mathematical Programming Society, Boltenhagen, Germany, 2001.

[CNS82] Norishige Chiba, Takao Nishizeki, and Nobuji Saito. An approximation algorithm for the maximum independent set problem on planar graphs. *SIAM J. Comput.*, 11(4):663–675, 1982.

[DF99] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Springer-Verlag, New York, 1999.

[DHT02] Erik D. Demaine, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Exponential speedup of fixed parameter algorithms on $K_{3,3}$-minor-free or $K_5 - minor - free$ graphs. Technical Report MIT-LCS-TR-838, M.I.T, March 2002.

[Epp00] David Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3-4):275–291, 2000.

[Haj01] MohammadTaghi Hajiaghayi. Algorithms for Graphs of (Locally) Bounded Treewidth. Master's thesis, University of Waterloo, September 2001.

[HM94] Michel Habib and Rolf H. Möhring. Treewidth of cocomparability graphs and a new order-theoretic parameter. *ORDER*, 1:47–60, 1994.

[HNRT01] Mohammadtaghi Hajiaghayi, Naomi Nishimura, Prabhakar Ragde, and Dimitrios M. Thilikos. Fast approximation schemes for $K_{3,3}$-minor-free or K_5-minor-free graphs. In *Euroconference on Combinatorics, Graph Theory and Applications 2001 (Barcelona, 2001)*. 2001.

[HT73] J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2:135–158, 1973.

[KC00] Tom Kloks and Leizhen Cai. Parameterized tractability of some (efficient) Y-domination variants for planar graphs and t-degenerate graphs. In *International Computer Symposium (ICS)*. Taiwan, 2000.

[KLL01] Tom Kloks, C.M. Lee, and Jim Liu. Feedback vertex sets and disjoint cycles in planar (di)graphs. In *Optimization Online*. Mathematical Programming Society, Philadelphia, 2001.

[Klo93] Ton Kloks. Treewidth of circle graphs. In *Algorithms and computation (Hong Kong, 1993)*, pages 108–117. Springer, Berlin, 1993.

[KM92] André Kézdy and Patrick McGuinness. Sequential and parallel algorithms to find a K_5 minor. In *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms (Orlando, FL, 1992)*, pages 345–356, 1992.

[KP02] Iyad A. Kanj and Ljubomir Perkovic. Improved parameterized algorithms for planar dominating set. In *27th International Symposium on Mathematical Foundations of Computer Science, MFCS 2002*. Warszawa - Otwock, Poland, August 26-30, 2002. To appear.

[KR91] Arkady Kanevsky and Vijaya Ramachandran. Improved algorithms for graph four-connectivity. *J. Comput. System Sci.*, 42(3):288–306, 1991. Twenty-Eighth IEEE Symposium on Foundations of Computer Science (Los Angeles, CA, 1987).

[LT80] Richard J. Lipton and Robert Endre Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9(3):615–627, 1980.

[MR92] Gary L. Miller and Vijaya Ramachandran. A new graph triconnectivity algorithm and its parallelization. *Combinatorica*, 12(1):53–76, 1992.

[RS84] Neil Robertson and Paul D. Seymour. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory Series B*, 36:49–64, 1984.

[RS85] Neil Robertson and Paul D. Seymour. Graph minors — a survey. In I. Anderson, editor, *Surveys in Combinatorics*, pages 153–171. Cambridge Univ. Press, 1985.

[RS86] Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

[RS91] Neil Robertson and Paul D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory Series B*, 52:153–190, 1991.

[RS93] Neil Robertson and Paul Seymour. Excluding a graph with one crossing. In *Graph structure theory (Seattle, WA, 1991)*, pages 669–675. Amer. Math. Soc., Providence, RI, 1993.

[SSR94] Ravi Sundaram, Karan S. Singh, and Pandu C. Rangan. Treewidth of circular-arc graphs. *SIAM J. Discrete Math.*, 7(4):647–655, 1994.

[ST94] Paul D. Seymour and Robin Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.

[Tar72] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.

[Wag37] Kehrer Wagner. Über eine Eigenschaft der eben Komplexe. *Deutsche Math.*, 2:280–285, 1937.

[Wil84] S. G. Williamson. Depth-first search and Kuratowski subgraphs. *J. Assoc. Comput. Mach.*, 31(4):681–693, 1984.

[Yan78] Mihalis Yannakakis. Node- and edge-deletion NP-complete problems. In *Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, CA, 1978)*, pages 253–264. ACM press, New York, 1978.