数理解析研究所講究録
950 巻 1996 年 80-86

Sequential and Parallel Approximation of Maximum Induced-Subgraph Problems on Sparse Graphs

陳 致中 (Zhi-Zhong Chen)
Dept. of Math. Sci., Tokyo Denki Univ., Hatoyama, Saitama 350-03, Japan

Abstract

We show that for an integer $k \geq 2$ and an n-vertex graph G without a $K_{3,3}$ (resp., K_5) minor, we can compute k induced subgraphs of G with treewidth $\leq 3k-4$ (resp., $\leq 6k-7$) in $O(kn)$ (resp., $O(kn+n^2)$) time such that each vertex of G appears in exactly $k-1$ of these subgraphs. This leads to *practical* polynomial-time approximation schemes for various maximum induced-subgraph problems on graphs without a $K_{3,3}$ or K_5 minor. The result extends a well-known result of Baker that there are practical polynomial-time approximation schemes for various maximum induced-subgraph problems on *planar* graphs.

1 Introduction

Let π be a property on graphs. π is *hereditary* if, whenever a graph G satisfies π, every induced subgraph of G also satisfies π. Suppose π is a hereditary property. The *maximum induced subgraph problem* associated with π (MISP(π)) is the following: Given a graph $G=(V,E)$, find a maximum subset U of V that induces a subgraph satisfying π. Yannakakis showed that various natural MISP(π)'s are NP-hard even if the input graph is restricted to a planar graph [12]. Thus, it is of interest to design efficient approximation algorithms for these MISP(π)'s.

An approximation algorithm A for a maximization problem Π achieves a *performance ratio* of ρ if for every instance I of Π, the ratio of the optimal value for I to the solution value returned by A is at most ρ. A *polynomial-time approximation scheme* (PTAS) for problem Π is an approximation algorithm which given an instance I of Π and an $\epsilon > 0$, returns a solution s within time polynomial in the size of I such that the ratio of the optimal value for I to the value of s is at most $(1+\epsilon)$.

Much work has been devoted to designing PTASs for MISP(π)'s restricted to certain special instances [1, 4, 10]. Lipton and Tarjan were the first who proved that various MISP(π)'s restricted to *planar* instances have PTASs [10]. In their approach, they applied their planar separator theorem. Unfortunately, their schemes are known to be nonpractical. That is, to achieve a reasonable performance ratio (e.g., 2), the number of vertices in the input graph and/or the running time of the schemes has to be enormous ($\approx 2^{2^{400}}$). Later, Baker gave practical PTASs for the same problems using a different approach [4]. By extending Lipton & Tarjan's approach, Alon et al. [1] showed that various MISP(π)'s restricted to graphs without an excluded minor have polynomial-time approximation schemes. Like Lipton and Tarjan's schemes, Alon et al.'s schemes have the shortage of being *very* nonpractical. Very recently, Eppstein proved that if $\mathcal{F}$ is a family of graphs without an excluded minor and does not contain all apex graphs, then there is a function f such that every graph in $\mathcal{F}$ with diameter at most D has treewidth $f(D)$ [5]. Combining this result together with Baker's approach leads to PTASs for MISP(π)'s restricted to graphs in such a family $\mathcal{F}$. Unfortunately, Eppstein's proof is based on Robertson & Seymour's "planar obstruction theorem" and $f(D)$ is *extremely* large (even if D is small) [5]. Consequently, the resulting PTASs are nonpractical.

Since neither Alon et al.'s schemes nor the schemes implied by Eppstein's result above are practical, it is natural to ask whether practical PTASs exist for MISP(π)'s restricted to graphs without an excluded minor. In this paper, we give an affirmative answer to this question when the minor is $K_{3,3}$ or K_5. Since neither a $K_{3,3}$ minor nor a K_5 minor can exist in a planar graph, our result extends Baker's result above. Our schemes can be viewed as a modification of Baker's schemes. Recall that Baker's schemes consist of three steps. First, decompose the input planar graph G into k $(k-1)$-outerplanar (induced) subgraphs G_1, $\cdots$, G_k such that each vertex of G appears in exactly $k-1$ of these subgraphs. Next, compute an optimal solution s_i in each G_i using dynamic programming. Finally, output the best one among s_1, $\cdots$, s_k as a (nearly optimal) solution in the original graph G. In [4], Baker shows that the output solution has size at least $(k-1)/k$ optimal. Our schemes differ from Baker's only in the first step. This difference is essential because it is impossible to perform the first step above when G is not planar. In our schemes, the input graph G without a $K_{3,3}$ (resp., K_5) minor is decomposed into k induced subgraphs with treewidth $\leq 3k-4$ (resp., $\leq 6k-7$) in $O(kn)$ (resp., $O(kn+n^2)$) time such that each vertex of G appears in exactly $k-1$ of these subgraphs. This decomposition is based on the nice structures of graphs without a $K_{3,3}$ or K_5 minor that were developed in [2, 6, 9]. Roughly speaking, these nice structures say that a graph without a $K_{3,3}$ (resp., K_5) minor must have very special 3-connected (resp., 4-connected) components each of which can easily be decomposed into induced subgraphs of bounded treewidth. The problem is how to combine the decompositions of these components into a (single) decomposition of the original graph G. We solve this problem by organizing these components into a suitable tree. The other two steps in our schemes are the same as those in Baker's, and therefore can be done in practical polynomial (often linear) time because various MISP(π)'s restricted to graphs of bounded treewidth can be computed in practical polynomial (often linear) time by dynamic programming [11]. Besides their practicality, our schemes also have the advantage of being easy to parallelize.

2 Preliminaries

Throughout this paper, a graph is always connected. Unless stated explicitly, a graph is always simple, i.e., has neither multiple edges nor self-loops. Let $G=(V,E)$ be a graph. For convenience, we allow $V=\emptyset$. If $V=\emptyset$, then we call G an *empty* graph. We sometimes write $V(G)$ instead of V and $E(G)$ instead of E. The *neighborhood* of a vertex v in G is the set of vertices in G adjacent to v. For $U\subseteq V$, the *subgraph of G induced by U* is the graph (U,F) with $F=\{\{u,v\}\in E : u,v\in U\}$ and is denoted by $G[U]$. When $U\subseteq V$, we sometimes write $G-U$ instead of $G[V-U]$.

A *contraction* of an edge $\{u,v\}$ in G is made by identifying u and v with a new vertex whose neighborhood is the union of the neighborhoods of u and v (resulting multiple edges and self-loops are deleted). A *contraction* of G is a graph obtained from G by a sequence of edge contractions. A graph H is a *minor* of G if H is the contraction of a subgraph of G. G is H-*free* if G has no minor isomorphic to H. In this paper, we deal with $K_{3,3}$-free graphs and K_5-free graphs. Recall that a planar graph must be both $K_{3,3}$-free and K_5-free by Kuratowski's Theorem.

A *tree-decomposition* of G is a pair $(\{X_i : i\in I\},T)$, where $\{X_i : i\in I\}$ is a family of subsets of V and T is a tree with $V(T)=I$ such that the following hold:

(a) $\cup_{i\in I}X_i=V$.

(b) For every edge $\{v,w\}\in E$, there is a subset X_i, $i\in I$ with $v\in X_i$ and $w\in X_i$.

(c) For all $i,j,k\in I$, if j lies on the path from i to k in T, then $X_i\cap X_k\subseteq X_j$.

The *treewidth* of a tree-decomposition $(\{X_i : i\in I\},T)$ is $\max\{|X_i|-1 : i\in I\}$. The *treewidth* of G, denoted by tw(G), is the minimum treewidth of a tree-decomposition of G, taken over all possible tree-decompositions of G. The treewidth of an empty graph is defined to be 0.

Lemma 2.1 [Robertson & Seymour] Let $G=(V,E)$ be a graph, and R_1 and R_2 be two subsets of V such that (i) $R_1\cap R_2=\emptyset$ or $G[R_1\cap R_2]$

is a clique and (ii) there is no $\{u_1, u_2\} \in E$ with $u_1 \in R_1 - R_2$ and $u_2 \in R_2 - R_1$. Then, $\text{tw}(G[R_1 \cup R_2]) \leq \max\{\text{tw}(G[R_1]), \text{tw}(G[R_2])\}$.

A set $S \subseteq V$ is a *cutset* if $G - S$ is disconnected. A cutset S is a *k-cut* if $|S| = k$. A k-cut is *strong* if $G - S$ has at least three connected components. A graph with at least k vertices is *k-connected* if it has no $(k-1)$-cut. A *biconnected component* of G is a maximal 2-connected subgraph of G.

Let C be a cutset of G, and G_1, ..., G_p be the connected components of $G - C$. For $1 \leq i \leq p$, let $G_i \cup K(C)$ be the graph obtained from $G[V(G_i) \cup C]$ by adding an edge between every pair of non-adjacent vertices in C. The graphs $G_1 \cup K(C)$, ..., $G_p \cup K(C)$ are called the *augmented components* induced by C. Clearly, if G is k-connected and C is a k-cut of G, then all the augmented components induced by C are also k-connected.

It is well known that the biconnected components of a graph are unique. Let $\mathcal{C}^1$ be the set of all 1-cuts of G, and $\mathcal{B}$ be the set of all biconnected components of G. Consider the bipartite graph $H = (\mathcal{C}^1 \cup \mathcal{B}, F)$, where $F = \{\{C, B\} : C \in \mathcal{C}^1, B \in \mathcal{B}$, and $C \subseteq V(B)\}$. It is known that H is a tree. Suppose that $\mathcal{B} = \{B_1, ..., B_q\}$. Let $I = \{1, ..., q\}$. Root the tree H at B_1 and define $\mathcal{T}^1(G)$ to be the tree whose vertex set is I and edge set is $\{\{i, i'\} : B_i$ is the grandparent of $B_{i'}$ in the rooted tree $H\}$. (Note that $\mathcal{T}^1(G)$ is undirected.) The following fact is easy to prove.

Fact 1 $(\{V(B_i) : i \in I\}, \mathcal{T}^1(G))$ is a tree-decomposition of G and can be computed from G in $O(|V|)$ time.

Suppose that G is 2-connected. Further suppose that G contains a 2-cut. Replacing G by the augmented components induced by a 2-cut is called *splitting* G. Suppose G is split, the augmented components are split, and so on, until no more splits are possible. The graphs constructed in this way are 3-connected and the set of the graphs are called a *2-decomposition* of G. Each element of a 2-decomposition of G is called a *split component* of G. It is possible for G to have two or more 2-decompositions. A split component of G must be either a triangle or a 3-connected graph with at least 4 vertices. Let $\mathcal{D}$ be a 2-decomposition of G. We use $\mathcal{C}^2(\mathcal{D})$ to denote the set of the 2-cuts used to split G into the split components in $\mathcal{D}$. Consider the bipartite graph $H = (\mathcal{C}^2(\mathcal{D}) \cup \mathcal{D}, F)$, where $F = \{\{C, D\} : C \in \mathcal{C}^2(\mathcal{D}), D \in \mathcal{D}$, and $C \subseteq V(D)\}$. It is known that H is a tree. Suppose that $\mathcal{D} = \{D_1, ..., D_q\}$. Let $I = \{1, ..., q\}$. Root the tree H at D_1 and define $\mathcal{T}^2(G, \mathcal{D})$ to be the tree whose vertex set is I and edge set is $\{\{i, i'\} : D_i$ is the grandparent of $D_{i'}$ in the rooted tree $H\}$. (Note that $\mathcal{T}^2(G, \mathcal{D})$ is undirected.) Construct a supergraph $G^2(\mathcal{D})$ of G as follows: For each $\{u, v\} \in \mathcal{C}^2(\mathcal{D})$ with $\{u, v\} \notin E$, add the edge $\{u, v\}$ to G. Then, we have the following fact:

Fact 2 $(\{V(D_i) : i \in I\}, \mathcal{T}^2(G, \mathcal{D}))$ is a tree-decomposition of $G^2(\mathcal{D})$.

3 A technical lemma

Let S be a set. For an integer $k \geq 2$, a *k-cover* of S is a list of k subsets of S such that each element of S is contained in exactly $k - 1$ subsets in the list.

Lemma 3.1 Let $G = (V, E)$ be a graph. Let k and b be two integers with $k \geq 2$, and τ be a property on k-covers of subsets of V. Suppose that G has a tree-decomposition $(\{X_j : j \in I\}, T)$ and T has a rooted version such that the following three conditions are satisfied:

(1) For every $j' \in I$ and every child j of j' in T, $G[X_{j'} \cap X_j]$ is a clique.

(2) For the root $r \in I$ of T, we can compute a k-cover $\langle R_1, ..., R_k \rangle$ of X_r in $f(k, |X_r|)$ time such that

(2a) for every $1 \leq l \leq k$, $\text{tw}(G[R_l]) \leq b$ and

(2b) for every child j'' of r in T, $\langle R_1 \cap X_{j''}, ..., R_k \cap X_{j''} \rangle$ is a k-cover of $X_r \cap X_{j''}$ satisfying τ.

(3) For every $j' \in I$ and every child j of j' in T and every k-cover $\langle Y_1, ..., Y_k \rangle$ of $X_{j'} \cap X_j$ satisfying τ, we can compute a k-cover $\langle Z_1, ..., Z_k \rangle$ of X_j in $f(k, |X_j|)$ time such that

(3a) for every $1 \leq l \leq k$, $Y_l = Z_l \cap X_{j'}$,

(3b) for every $1 \leq l \leq k$, $\text{tw}(G[Z_l]) \leq b$, and

(3c) for every child j'' of j, $\langle Z_1 \cap X_{j''}, ..., Z_k \cap X_{j''} \rangle$ is a k-cover of $X_j \cap X_{j''}$ satisfying τ.
Then, we can compute a k-cover $\langle V_1, ..., V_k \rangle$ of V in $O(\sum_{j \in I} f(k, |X_j|))$ time such that for each $1 \leq l \leq k$, $\mathrm{tw}(G[V_l]) \leq b$ and $V_l \cap X_r = R_l$.

Proof. Consider the following algorithm for computing $\langle V_1, ..., V_k \rangle$:

Algorithm 1

1. Set V_1, ..., V_k to be the empty set.

2. While traversing T (starting at its root r) in a breadth-first manner, perform the following steps:

2.1. If the current vertex j is r, then compute a k-cover $\langle R_1, ..., R_k \rangle$ of X_r satisfying the two conditions (2a) and (2b) above, and further add the vertices in each R_l, $1 \leq l \leq k$, to V_l.

2.2. If the current vertex j is not r, then find the parent j' of j in T, set $\langle Y_1, ..., Y_k \rangle = \langle V_1 \cap (X_{j'} \cap X_j), ..., V_k \cap (X_{j'} \cap X_j) \rangle$, compute a k-cover $\langle Z_1, ..., Z_k \rangle$ of X_j satisfying the conditions (3a), (3b), and (3c) above, and add the vertices in each Z_l, $1 \leq l \leq k$, to V_l.

3. Output $\langle V_1, ..., V_k \rangle$.

Next, we prove that the output $\langle V_1, ..., V_k \rangle$ of Algorithm 1 satisfies that $\mathrm{tw}(G[V_l]) \leq b$ and $V_l \cap X_r = R_l$ for each $1 \leq l \leq k$. First note that the while-loop in Algorithm 1 is executed $|I|$ times. W.l.o.g., we may assume that $I = \{1, ..., |I|\}$ and that $j+1$ is traversed by Algorithm 1 right after j for each $1 \leq j \leq |I| - 1$. Then, $r = 1$. For each $1 \leq j \leq |I|$ and each $1 \leq l \leq k$, let V_l^j be the content of the variable V_l right after the jth iteration of the while-loop. We claim that for each $1 \leq j \leq |I|$, $\langle V_1^j, ..., V_k^j \rangle$ is a k-cover of $\cup_{1 \leq i \leq j} X_i$ satisfying the following three conditions:

(C1) $\mathrm{tw}(G[V_l^j]) \leq b$ and $V_l^j \cap X_1 = R_l$ for each $1 \leq l \leq k$.

(C2) For each son j'' of j in T, $\langle V_1^j \cap (X_j \cap X_{j''}), ..., V_k^j \cap (X_j \cap X_{j''}) \rangle$ is a k-cover of $X_j \cap X_{j''}$ satisfying τ.

(C3) For each $1 \leq i \leq j$ and each child i' of i in T, $\langle V_1^j \cap (X_i \cap X_{i'}), ..., V_k^j \cap (X_i \cap X_{i'}) \rangle = \langle V_1^i \cap (X_i \cap X_{i'}), ..., V_k^i \cap (X_i \cap X_{i'}) \rangle$.

The lemma follows from the claim. We can prove the claim by induction on j. ∎

Let $G = (V, E)$ be a graph, and U be a subset of V. A k-cover L of U is *completely unbalanced* if exactly one set in L is empty and the others are equal to U. A k-cover L of U is *weakly unbalanced* if there are one vertex $u \in U$ and two sets U_1 and U_2 in L such that $U_1 = \{u\}$, $U_2 = U - \{u\}$, and all the sets in L except U_1 and U_2 are equal to U. A k-cover of U is *unbalanced* if it is either completely unbalanced or weakly unbalanced. Note that if $|U| \leq 2$, then every k-cover of U must be unbalanced. Hereafter, the property τ in Lemma 3.1 means "unbalanced", i.e., a k-cover L of U satisfies τ if and only if L is unbalanced.

4 Approximating MISP(π)'s on $K_{3,3}$-free graphs

Lemma 4.1 Let $G = (V, E)$ be a connected planar graph, and k be an integer ≥ 2. Suppose that s_1 and s_2 are two adjacent vertices in G and $\langle Y_1, ..., Y_k \rangle$ is an unbalanced k-cover of $\{s_1, s_2\}$. Then, we can compute a k-cover $\langle Z_1, ..., Z_k \rangle$ of V in $O(k|V|)$ time such that $\mathrm{tw}(G[Z_l]) \leq 3k - 4$ and $Z_l \cap \{s_1, s_2\} = Y_l$ for each $1 \leq l \leq k$.

Lemma 4.2 [2, 6]. Each split component of a 2-connected $K_{3,3}$-free graph is either isomorphic to K_5 or planar.

Lemma 4.3 Let $G = (V, E)$ be a 2-connected $K_{3,3}$-free graph. Then, for any $k \geq 2$, we can compute a k-cover $\langle V_1, ..., V_k \rangle$ of V in $O(k|V|)$ time such that $\mathrm{tw}(G[V_l]) \leq 3k - 4$ for each $1 \leq l \leq k$.

Proof. Let $\mathcal{D} = \{D_1, ..., D_q\}$ be a 2-decomposition of G, and let $I = \{1, ..., q\}$. It is known that $\mathcal{D}$ can be computed in $O(|V|)$ time [7]. Moreover, $\sum_{i \in I} |V(D_i)| = O(|V|)$ [7]. W.l.o.g., we may assume that $G^2(\mathcal{D}) = G$ because a k-cover $\langle V_1, ..., V_k \rangle$ of V such that the subgraph of $G^2(\mathcal{D})$ induced by V_l has treewidth $\leq 3k-4$ for each $1 \leq l \leq k$ is also a k-cover $\langle V_1, ..., V_k \rangle$ of V such that $\text{tw}(G[V_l]) \leq 3k-4$ for each $1 \leq l \leq k$. Then, by Fact 2, $(\{V(D_j) : j \in I\}, \mathcal{T}^2(G,\mathcal{D}))$ is a tree-decomposition of G. For convenience, let $T = \mathcal{T}^2(G,\mathcal{D}))$, $b = 3k-4$, and $X_j = V(D_j)$ and $f(k,|X_j|) = O(k|X_j|)$ for each $j \in I$. We want to apply Lemma 3.1 to the graph G and the tree-decomposition $(\{X_j : j \in I\}, T)$. To this end, we first (arbitrarily) choose an $r \in I$ and root T at r.

Clearly, the condition (1) in Lemma 3.1 is satisfied by G and $(\{X_j : j \in I\}, T)$. By Lemma 4.2, $G[X_r] = D_r$ is either isomorphic to K_5 or planar. Let us first suppose that $G[X_r]$ is isomorphic to K_5. Then, we set $R_1 = \emptyset$ and $R_2 = \cdots = R_k = X_r$ if $k \geq 3$; otherwise $(k=2)$, we arbitrarily choose two vertices v_1 and v_2 in X_r and set $R_1 = \{v_1, v_2\}$ and $R_2 = X_r - R_1$. Obviously, $\langle R_1, ..., R_k \rangle$ is a k-cover of X_r satisfying the condition (2a) in Lemma 3.1. $\langle R_1, ..., R_k \rangle$ also satisfies the condition (2b) in Lemma 3.1 since $|X_r \cap X_{j''}| = 2$ for every child j'' of r in T. Next, suppose that $G[X_r]$ is a planar graph. Then, we arbitrarily choose an edge $\{s_1, s_2\}$ in $G[X_r]$, set $Y_1 = \emptyset$ and $Y_2 = \cdots = Y_k = \{s_1, s_2\}$, and use Lemma 4.1 to compute a k-cover $\langle R_1, ..., R_k \rangle$ of X_r in $O(k|X_r|)$ time such that $\text{tw}(G[R_l]) \leq 3k-4$ for each $1 \leq l \leq k$. Clearly, $\langle R_1, ..., R_k \rangle$ satisfies the condition (2a) in Lemma 3.1. $\langle R_1, ..., R_k \rangle$ also satisfies the condition (2b) in Lemma 3.1 since $|X_r \cap X_{j''}| = 2$ for every child j'' of r in T.

Fix a $j' \in I$ and a child j of j' in T. Let $\langle Y_1, ..., Y_k \rangle$ be an unbalanced k-cover of $X_{j'} \cap X_j$. W.l.o.g., we may assume that $|Y_l| \leq |Y_{l+1}|$ for each $1 \leq l \leq k-1$. By Lemma 4.2, $G[X_j] = D_j$ is either isomorphic to K_5 or planar. Let us first suppose that $G[X_j]$ is isomorphic to K_5. If $k \geq 3$, then we set $Z_1 = Y_1$ and $Z_l = Y_l \cup (X_j - X_{j'})$ for each $2 \leq l \leq k$. Otherwise $(k=2)$, we arbitrarily choose a vertex $v \in X_j - X_{j'}$ and set $Z_1 = Y_1 \cup (X_j - (X_{j'} \cup \{v\}))$ and $Z_2 = Y_2 \cup \{v\}$. Then, no matter what k is, $\langle Z_1, ..., Z_k \rangle$ is a k-cover of X_j satisfying the conditions (3a), (3b), and (3c) in Lemma 3.1. Next, suppose that $G[X_j]$ is planar. Let $X_{j'} \cap X_j = \{s_1, s_2\}$. Note that s_1 and s_2 are adjacent in G. We use Lemma 4.1 to compute a k-cover $\langle Z_1, ..., Z_k \rangle$ of X_j. It should be easy to see that $\langle Z_1, ..., Z_k \rangle$ is a k-cover of X_j satisfying the conditions (3a), (3b), and (3c) in Lemma 3.1. ∎

Theorem 4.4 Let $G = (V,E)$ be a $K_{3,3}$-free graph. Then, for any $k \geq 2$, we can compute a k-cover $\langle V_1, ..., V_k \rangle$ of V in $O(k|V|)$ time such that $\text{tw}(G[V_l]) \leq 3k-4$ for $1 \leq l \leq k$.

Corollary 4.5 Let π be a hereditary property on graphs. Suppose that MISP(π) restricted to n-vertex graphs of treewidth $\leq k$ can be solved in $T_\pi(k,n)$ time. Then, given an integer $k \geq 2$ and a $K_{3,3}$-free graph $G = (V,E)$, we can compute a subset U of V in $O(k|V| + T_\pi(3k-4, |V|))$ time such that $G[U]$ satisfies π and $|U|$ is at least $(k-1)/k$ optimal.

For various properties π, $T_\pi(k,n) = 2^{p(k)}q(n)$ where p and q are polynomials of low degree (often, of degree 1) [11]. Hence, for such properties π, MISP(π) restricted to $K_{3,3}$-free graphs has a practical polynomial-time approximation scheme by Corollary 4.5.

5 Approximating MISP(π)'s on K_5-free graphs

We start by giving several definitions. Suppose that G is 3-connected. Further suppose that G contains a *strong* 3-cut. Replacing G by the augmented components induced by a strong 3-cut is called *strongly splitting* G. Suppose G is strongly split, the augmented components are strongly split, and so on, until no more strong splits are possible. The set of the graphs constructed in this way are called a *strong 3-decomposition* of G.

Definition 5.1 We define W to be the graph obtained from a 8-cycle by adding 4 crossing edges. More precisely, $W = (\{1,...,8\}, E_1 \cup E_2)$, where $E_1 = \{\{i,i+1\} : 1 \leq i \leq 7\} \cup \{\{8,1\}\}$ and $E_2 = \{\{i,i+4\} : 1 \leq i \leq 4\}$. A K_5-free graph G is said to be *nice* if G is 3-connected, nonplanar, and is not isomorphic to $K_{3,3}$ or W.

Fact 3 [9] Suppose that G is a nice K_5-free graph. Let C be a strong 3-cut in G. Then, the augmented components induced by C are also nice K_5-free graphs. Moreover, G has another strong 3-cut C' if and only if C' is a strong 3-cut of some augmented component of G induced by C.

Fact 4 [9] A nice K_5-free graph has a *unique* strong 3-decomposition. Moreover, each graph in the strong 3-decomposition is planar.

Suppose that $G = (V, E)$ is a nice K_5-free graph. Let $\mathcal{D}^3(G)$ be the strong 3-decomposition of G, and $\mathcal{C}^3(G)$ be the set of all strong 3-cuts in G. Define $H(G)$ to be the bipartite graph $(\mathcal{D}^3(G) \cup \mathcal{C}^3(G), F)$, where $F = \{\{D, C\} : D \in \mathcal{D}^3(G),\ C \in \mathcal{C}^3(G)$, and $C \subseteq V(D)\}$.

Lemma 5.2 (1) Every edge of G is contained in some graph in $\mathcal{D}^3(G)$.

(2) If a subset S of V induces a triangle but $S \notin \mathcal{C}^3(G)$, then exactly one graph in $\mathcal{D}^3(G)$ contains the three vertices in S.

(3) $H(G)$ is a tree. Moreover, if some vertex $u \in V$ is contained in two graphs D and D' in $\mathcal{D}^3(G)$, then u is contained in every graph on the path between D and D' in $H(G)$.

Suppose that $\mathcal{D}^3(G) = \{D_1, ..., D_q\}$. Let $I = \{1, ..., q\}$. Root the tree $H(G)$ at D_1 and define $\mathcal{T}^3(G)$ to be the tree whose vertex set is I and edge set is $\{\{i, i'\} : D_i$ is the grandparent of $D_{i'}$ in the rooted tree $H(G)\}$. (Note that $\mathcal{T}^3(G)$ is undirected.) Construct a supergraph G^3 of G as follows: For each strong 3-cut C and each pair of nonadjacent vertices u and v in C, add the edge $\{u, v\}$ to G.

Corollary 5.3 $(\{V(D_i) : i \in I\}, \mathcal{T}^3(G))$ is a tree-decomposition of G^3.

Lemma 5.4 Let $G = (V, E)$ be a connected planar graph, and k be an integer ≥ 2. Suppose that S is a subset of V such that $G[S]$ is a triangle, and $\langle Y_1, ..., Y_k \rangle$ is an unbalanced k-cover of S. Then, we can compute a k-cover $\langle Z_1, ..., Z_k \rangle$ of V in $O(k|V|)$ time such that $\text{tw}(G[Z_l]) \leq 6k - 7$ and $Z_l \cap S = Y_l$ for each $1 \leq l \leq k$, and $\langle Z_1 \cap S', ..., Z_k \cap S' \rangle$ is an unbalanced k-cover of S' for all subsets S' of V with $G[S']$ being a triangle. ∎

Lemma 5.5 Let $G = (V, E)$ be a nice K_5-free graph, and k be an integer ≥ 2. Suppose that s_1 and s_2 are two adjacent vertices in G and $\langle U_1, ..., U_k \rangle$ is an unbalanced k-cover of $\{s_1, s_2\}$. Then, we can compute a k-cover $\langle V_1, ..., V_k \rangle$ of V in $O(k|V| + |V|^2)$ time such that $\text{tw}(G[V_l]) \leq 6k - 7$ and $V_l \cap \{s_1, s_2\} = U_l$ for each $1 \leq l \leq k$. ∎

Lemma 5.6 Let $G = (V, E)$ be a 2-connected K_5-free graph. Then, for any $k \geq 2$, we can compute a k-cover $\langle V_1, ..., V_k \rangle$ of V in $O(k|V| + |V|^2)$ time such that $\text{tw}(G[V_l]) \leq 6k-7$ for each $1 \leq l \leq k$.

Proof. Let $\mathcal{D} = \{D_1, ..., D_q\}$ be a 2-decomposition of G, and let $I = \{1, ..., q\}$. It is known that $\mathcal{D}$ can be computed in $O(|V|)$ time [7]. W.l.o.g., we may assume that $G^2(\mathcal{D}) = G$ because a k-cover $\langle V_1, ..., V_k \rangle$ of V such that the subgraph of $G^2(\mathcal{D})$ induced by V_l has treewidth $\leq 6k - 7$ for each $1 \leq l \leq k$ is also a k-cover $\langle V_1, ..., V_k \rangle$ of V such that $\text{tw}(G[V_l]) \leq 6k - 7$ for each $1 \leq l \leq k$. Then, by Fact 2, $(\{V(D_j) : j \in I\}, \mathcal{T}^2(G, \mathcal{D}))$ is a tree-decomposition of G. For convenience, let $T = \mathcal{T}^2(G, \mathcal{D})$, $b = 6k - 7$, and $X_j = V(D_j)$ and $f(k, |X_j|) = O(k|X_j| + |X_j|^2)$ for each $j \in I$. We want to apply Lemma 3.1 to the graph G and the tree-decomposition $(\{X_j : j \in I\}, T)$. To this end, we first (arbitrarily) choose an $r \in I$ and root T at r.

We only prove that the condition (3) in Lemma 3.1 is satisfied by G and $(\{X_j : j \in I\}, T)$. Fix a $j' \in I$ and a child j of j' in T. Let $\langle Y_1, ..., Y_k \rangle$ be an unbalanced k-cover of $X_{j'} \cap X_j$, and let $X_{j'} \cap X_j = \{s_1, s_2\}$. Recall that $\{s_1, s_2\}$ is an edge in both $G[X_{j'}]$ and $G[X_j]$. Moreover, by symmetry, we may assume that $|Y_l| \leq |Y_{l+1}|$ for all $1 \leq l \leq k - 1$. We distinguish four cases as follows:

Case 1': $G[X_j]$ is planar. Then, as stated in the proof of Lemma 4.3, we can compute a k-cover $\langle Z_1, ..., Z_k \rangle$ of X_j in $O(k|X_j|)$ time satisfying the conditions (3a), (3b), and (3c) in Lemma 3.1.

Case 2': $G[X_j]$ is isomorphic to $K_{3,3}$. Then, we set $Z_1 = Y_1$ and $Z_l = Y_l \cup (X_j - X_{j'})$ for each $2 \leq l \leq k$. Clearly, $\langle Z_1, ..., Z_k \rangle$ is a k-cover of X_j satisfying the conditions (3a), (3b), and (3c) in Lemma 3.1.

Case 3': $G[X_j]$ is isomorphic to the graph W (see Definition 5.1). If $k \geq 3$, then we set $Z_1 = Y_1$ and $Z_l = Y_l \cup (X_j - X_{j'})$ for each $2 \leq l \leq k$; otherwise ($k = 2$), we (arbitrarily) choose a subset A of $X_j - X_{j'}$ with $|A| = 3$ and set $Z_1 = Y_1 \cup A$ and $Z_2 = X_j - Z_1$. Then, it is easy to verify that $\langle Z_1, ..., Z_k \rangle$ is a k-cover of X_j satisfying the conditions (3a), (3b), and (3c) in Lemma 3.1.

Case 4': $G[X_j]$ is a nice K_5-free graph. Then, by Lemma 5.5, we can compute a k-cover $\langle Z_1, ..., Z_k \rangle$ of X_j in $O(k|X_j|+|X_j|^2)$ time such that $\text{tw}(G[Z_l]) \leq 6k-7$ and $Z_l \cap \{s_1, s_2\} = Y_l$ for each $1 \leq l \leq k$. From this, it should be clear that $\langle Z_1, ..., Z_k \rangle$ satisfies the conditions (3a), (3b), and (3c) in Lemma 3.1.

Note that one of the four cases must occur. Thus, by the discussions above and Lemma 3.1, we have the lemma. ▮

Theorem 5.7 Let $G = (V, E)$ be a K_5-free graph. Then, for any $k \geq 2$, we can compute a k-cover $\langle V_1, ..., V_k \rangle$ of V in $O(k|V| + |V|^2)$ time such that $\text{tw}(G[V_l]) \leq 6k-7$ for each $1 \leq l \leq k$.

Corollary 5.8 Let π be a hereditary property on graphs. Suppose that MISP(π) restricted to n-vertex graphs of treewidth $\leq k$ can be solved in $T_\pi(k, n)$ time. Then, given an integer $k \geq 2$ and a K_5-free graph $G = (V, E)$, we can compute a subset U of V in $O(k|V| + |V|^2 + T_\pi(6k-7, |V|))$ time such that $G[U]$ satisfies π and $|U|$ is at least $(k-1)/k$ optimal.

For various properties π, $T_\pi(k, n) = 2^{p(k)}q(n)$ where p and q are polynomials of low degree (often, of degree 1) [11]. Hence, for such properties π, MISP(π) restricted to K_5-free graphs has a practical polynomial-time approximation scheme by Corollary 5.8.

References

[1] N. Alon, P. Seymour, and R. Thomas, A separator theorem for graphs without an excluded minor and its applications, *STOC'90.*

[2] T. Asano, An approach to the subgraph homeomorphism problem, *TCS* **38** (1985).

[3] H.L. Bodlaender, Planar graphs with bounded treewidth, Technical Report.

[4] B.S. Baker, Approximation algorithms for NP-complete problems on planar graphs, *J. ACM* **41** (1994).

[5] D. Eppstein, Subgraph isomorphism in planar graphs and related problems, *SODA'95.*

[6] D.W. Hall, A note on primitive skew curves, *Bull. Amer. Math. Soc.* **49** (1943).

[7] J.E. Hopcroft and R.E. Tarjan, Dividing a graph into triconnected components, *SIAM-JC* **2** (1973), 135-158.

[8] A. Kanevsky and V. Ramachandran, Improved algorithms for graph four-connectivity, *FOCS'87.*

[9] A. Kézdy and P. McGuinness, Sequential and parallel algorithms to find a K_5 minor, *SODA'92.*

[10] R.J. Lipton and R.E. Tarjan, Applications of a planar separator theorem, *SIAM-JC* **9** (1980), 615-627.

[11] J.A. Telle and A. Proskurowski, Practical algorithms on partial k-trees with an application to domination-like problems, *WADS'93*, LNCS 709.

[12] M. Yannakakis, Node- and edge-deletion NP-complete problems, *STOC'78.*