Planarity allowing few error vertices in linear time

Ken-ichi Kawarabayashi
National Institute of Informatics
2-1-2, Hitotsubashi, Chiyoda-ku
Tokyo 101-8430, Japan
k_keniti@nii.ac.jp

Abstract— We show that for every fixed k, there is a linear time algorithm that decides whether or not a given graph has a vertex set X of order at most k such that $G-X$ is planar (we call this class of graphs k-*apex*), and if this is the case, computes a drawing of the graph in the plane after deleting at most k vertices. In fact, in this case, we shall determine the minimum value $l \leq k$ such that after deleting some l vertices, the resulting graph is planar. If this is not the case, then the algorithm gives rise to a minor which is not k-apex and is minimal with this property. This answers the question posed by Cabello and Mohar in 2005, and by Kawarabayashi and Reed (STOC'07), respectively.

Note that the case $k = 0$ is the planarity case. Thus our algorithm can be viewed as a generalization of the seminal result by Hopcroft and Tarjan (J. ACM 1974), which determines if a given graph is planar in linear time. Our algorithm can be also compared to the algorithms by Mohar (STOC'96 and Siam J. Discrete Math 2001) for testing the embeddability of an input graph in a fixed surface in linear time, by Kawarabayashi and Mohar (STOC'08) for testing polyhedral embeddability of an input graph in a fixed surface in linear time, and by Kawarabayashi and Reed (STOC'07) for testing the fixed crossing number in linear time. Note that deciding the genus of k-apex graphs is NP-complete, even for $k = 1$, as shown by Mohar. Thus k-apex graphs are very different from bounded genus graphs in a sense.

In addition, for any fixed c, k, we apply our algorithm to obtain a linear time approximation scheme for weighted TSP, and for minimum weighted c-edge-connected submultigraph, respectively, for k-apex graphs. (In this case, an embedding of a k-apex graph is not given in the input). The first result generalizes the recent planar result by Klein (FOCS'05), while the second result generalizes Czumaj et al. (SODA'04). We also extend several optimization results for planar graphs by Baker (J. ACM. 1994) and others to k-apex graphs.

Keywords-Planarity, Few errors, TSP, Approximation Algorithms and linear time

1. Introduction

1.1. Planarity Testing and Graph Embeddings

A seminal result of Hopcroft and Tarjan [24] from 1974 is that there is a linear time algorithm for testing planarity of graphs. This is just one of a host of results on embedding graphs in surfaces. These problems are of both practical and theoretical interest. The practical issues arise, for instance, in problems concerning VLSI, and also in several other applications of "nearly planar" networks, because planar graphs and graphs embedded in low genus surfaces can be handled more easily. Theoretical interest comes from the facts that there are many generally hard problems which can be solved in polynomial time (often, even linear time) when considering planar graphs or "nearly" planar graphs, e.g., MAXIMUM CLIQUE, SUBGRAPH ISOMORPHISM [18]. Even for problems that remain NP-hard on planar graphs, we often have efficient approximation algorithms, e.g, INDEPENDENT SET, VERTEX COVER, DOMINATION SET etc [4], [31]. Recently, some apparently new and nontrivial linear time algorithms concerning graph embeddings appear: One [25] is concerning drawing a given graph into the plane with at most k crossings (for any fixed k), and another one [26] is concerning embedding a given graph into a surface with face-width at least k with an application to the graph isomorphism problem (for any fixed k). Both algorithms depend on the linear time algorithm for embedding a given graph into a given surface [27], [33], [34]. Computing the genus of graphs in certain minor closed families has mainly theoretical (but also some practical) importance. We refer to Mohar and Schrijver [36] for a discussion on this topic. While some minor closed families of graphs allow polynomial time genus computation, there are some classes where genus testing is NP-complete. The simplest such family is the class of all 1-apex graphs [35], i.e, there is a vertex v in a given graph G such that $G - v$ is planar.

In this paper, we are interested in k-apex graphs, which is a generalization of 1-apex graphs. Let us say that a graph is k-*apex* if it contains a set of at most k vertices whose removal yields a planar graph. In addition to practical importance, the class of k-apex graphs is important in a theoretical sense. K-apex graphs are important in the seminal structure theorem in graph minor project [44], [45]. In addition, determining whether or not a given graph is k-apex for any fixed k can be classified in the framework of parameterized complexity developed by Downey and Fellows [17]. In the parameterized complexity literature, a lot of similar vertex-deletion problems have been studied, for example, the feedback set problem. Let us observe that if k is as part of input, then this problem is still NP-hard [30]. Thus k must be fixed. Note that the case $k = 0$ is the planarity case. Thus determining whether or not a given graph is k-apex for any fixed k is a quite natural question in this context. Moreover, k-apex graphs are exactly the obstructions for " bounded local tree-width graphs", which unify "planar graphs and bounded genus graphs" and "bounded tree-width graph", and are drawn much attention by many researchers [19], [22].

We define the *apex number* of a graph G as the minimum k for which G is k-apex. Our main result is to extend the seminal result of Hopcroft and Tarjan [24] on the planarity testing to k-apex graphs for any fixed k. We give a linear time algorithm to test whether or not a given graph G is k-apex for any fixed k. In addition, we give an embedding of a k-apex graph if one exists in G, and determine the apex number (of a k-apex graph). Moreover, we use our main result to generalize some optimization results for planar graphs [4], [18], [31] to k-apex graphs. Furthermore, we generalize some linear time approximation schemes for planar graphs [5], [11], [29] to k-apex graphs.

The class of k-apex graphs seems very different from bounded genus graphs, and planar graphs with at most k edges (i.e, after deleting at most k edges, the resulting graph becomes planar). There are linear time algorithms to test whether or not a given graph can be embedded into a surface of bounded genus [27], [33], [34]. Also, there is a linear time algorithm to figure out whether or not a given graph is a planar graph with at most k edges. This was shown in [25]. Let us observe that its proof depends on the above mentioned linear time algorithms [27], [33], [34]. Note that planar graphs with at most k edges can be embedded into a surface of Euler genus k. On the other hand, we cannot embed a given k-apex graph on a surface with low genus, since it is known to be NP-complete to decide the genus of a k-apex graph (even 1-apex graphs, see [35]). The difficulty is that some 1-apex graph has huge Euler genus. For example, Euler genus of the bipartite graph $K_{3,n-3}$ is $\lceil (n-5)/2 \rceil$. This makes a huge difference between bounded genus graphs, planar graphs with at most k edges, and k-apex graphs, in terms of designing a linear time algorithm for embedding k-apex graphs, because we cannot use the above mentioned linear time algorithms [27], [33], [34].

On the other hand, it is easy to check in quadratic time if a graph is 1-apex - simply test for each vertex v, if $G - v$ is planar. In fact, it is easy to check in $O(n^{k+1})$ time if a graph is k-apex. Robertson and Seymour [44] have already proved that all classes of graphs that are closed under taking minors are recognizable in cubic time. Since k-apex graphs are closed under taking minors, this implies that there is an $O(n^3)$ algorithm for deciding whether G is k-apex for fixed k. Moreover, Marx and Scholotter [32] gives an $O(n^2)$ time algorithm for recognizing k-apex graphs for any fixed k. Our algorithm improves the time complexity of their result to linear time.

1.2. Our main results

In this paper, we prove the following result.

Theorem 1.1: For every k and any graph G, there is a linear time algorithm which returns one of the following:

(1) an embedding of G into a plane after deleting at most k vertices, or
(2) a minor of G which is not k-apex and is minimal with this property.

In fact, if the output is (1), then our algorithm determines the apex number $l \leq k$. The running time is $O(f(k)n)$, where $f(k)$ is double exponential of k.

This proves a conjecture by Cabello and Mohar [9] and by Kawarabayashi and Reed ([25], STOC'07), and improves the $O(n^2)$ algorithm given by Marx and Scholotter [32]. This problem was also discussed by Fellows and Langston [20]. Our proof implies the uniformity, which means that we have an algorithm which works for all $l \leq k$.

We have learned that Cabello and Mohar (private communication) gave a linear time algorithm for the case $k = 1$ in Theorem 1.1, i.e, 1-apex graphs, but their method is completely different from ours.

Our algorithm has several appealing features. It applies techniques which were used in [40], [41] to obtain a linear time algorithm to solve the k disjoint paths problem for planar graphs (improving the $O(n^3)$ algorithm of Robertson and Seymour [44] for any fixed k). We also use a result by Mohar [35] concerning face-cover in a plane graph.

Some applications to TSP and related problems.: The Traveling Salesman Problem is a classic problem that has served as a testbed for almost every new algorithmic idea over the past 50 years. It has been considered extensively in planar graphs and its generalizations, starting with a PTAS for unweighted planar graphs [21], then a PTAS for weighted planar graphs [2], recently improved to linear time [29]. There is also a PTAS [14] for bounded genus graphs.

Our algorithm of Theorem 1.1 can be used to give linear time approximation schemes.

Theorem 1.2: For any fixed k, any constant $c \geq 2$, and any $0 < \epsilon \leq 1$, given a graph which is k-apex (but an embedding is not given), there is a linear time $(1+\epsilon)$-approximation algorithm for weighted TSP, and for minimum-weight c-edge-connected submultigraph, respectively. (The minimum-weight c-edge-connected submutigraph problem allows using multiple copies of an edge in the input graph, and hence *submultigraph*, but the solution must pay for every copy.)

In fact, the proof of Theorem 1.2 follows from our algorithm of Theorem 1.1, together with the results by Demaine, Hajaghayi and Mohar [14]. Theorem 1.2 generalizes the linear time approximation scheme for weighted TSP for planar graphs by Klein [29] and the polynomial time approximation scheme for minimum-weight c-edge-connected submultigraph for planar graphs when $c = 2$ by Berger et al. [5] and Czumaj et al. [11].

TSP and minimum c-edge-connected submultigraph are examples of a general class of problems called *contraction-closed problems*, where the optimal solution only improves when contracting an edge. Many other classic problems are contraction-closed, for example, dominating set (and its many variations) and minimum chordal completion. The same proof of Theorem 1.2 implies that there are linear time $(1+\epsilon)$-approximation algorithms for these problems.

We also extend several optimization results for planar graphs to k-apex graphs, using the results in [4], [12], [13]. Details will be in Section 6. Proofs of Theorem 1.2 will be also give in Section 6.

1.3. Overview of the algorithm

We now turn to our algorithm for constructing an embedding of a given graph G into a plane after deleting at most k vertices (which we call *apex vertex set*), if one exists. Otherwise, we discover a minimal forbidden minor for k-apex graphs contained in G. The crux of the matter is to understand which vertices are irrelevant with respect to *any* embedding of G into a plane after deleting at most k vertices. In order to describe this idea, it will be convenient to define the following;

1) Let A^U be a maximum set of vertices in G such that no matter how we embed G into a plane after deleting at most k vertices from G, A^U is contained in an apex vertex set. By *maximum*, we mean that there is no vertex v in $G - A^U$ that can be in A^U. Sometimes, we call A^U the *universal apex vertex set* for G. Any vertex in A^U is sometimes called a *universal apex vertex* for G.
2) Let A_P be a set of vertices in G such that $A_P \cap A^U = \emptyset$, but for each vertex $x \in V(A_P)$, there is a vertex set A' with $x \in A'$ whose order is exactly the apex number $l \leq k$ of G, such that $G - A'$ can be embedded into a plane. (Let us observe that $A^U \subseteq A'$.)

When we say that a vertex v in G is irrelevant, we mean that $v \notin (A^U \cup A_P)$, and furthermore, no matter how we embed $G - v$ into a plane after deleting $l \leq k$ vertices (where l is the apex number of G), we can put the vertex v to this embedding so that the resulting embedding is still planar. We shall show the following;

1) If a vertex v in $G - A^U$ is contained in a planar subgraph Q of $G - A^U$, and v is surrounded by a huge wall (a wall of height at least $2k$), then $v \notin A_P$, and v is irrelevant
2) The removal of all these irrelevant vertices, say a set Y, yields a graph of bounded tree-width if G is k-apex. Moreover, if the universal apex vertex set A^U is given, there is a linear time algorithm to find the vertex set Y in $G - A^U$.

Thus one key is to detect the vertex set A^U.

Let us describe our algorithm more precisely. We may assume that the minimum degree of G is at least 2. For some small but constant $\epsilon > 0$, we iteratively find a sequence of graphs $G = G_0, G_1, \dots, G_b$ such that G_i is obtained from G_{i-1} by either

1) contracting an induced matching M_i with at least $\epsilon|G_{i-1}|$ edges, or
2) deleting a stable set I of $\epsilon|G_{i-1}|$ vertices, each of degree l, where $l \leq k+3$.

In the latter case, for every vertex x in I, the following holds:

1) there are at least l vertices in I such that each of them has the exactly same neighbors as x.
2) In addition, there are $k+3$ vertices in G_{i-1} that are not in I, such that each of them has the exactly same neighbors as x.

In this case, we also put a clique to G_{i-1} for the neighbors of every $x \in I$. Then G_i is the resulting graph. Note that the resulting graph G_i is a minor of G_{i-1}.

We stop after b steps, where b is minimum value such that G_b has fewer than B vertices for some constant B. Clearly $b \leq log_{1/\epsilon}(n/B)$ and hence it turns out that the sum of the sizes of all encountered graphs G_i is $O(\frac{1}{\epsilon}n)$.

For each i, we will either construct a desired embedding for G_i with an apex vertex set A_i of order at most k, or give a minimal forbidden minor of k-apex graphs in G_i, in time $O(|G_i|)$. It is easy to do this for G_b in constant time, since it has bounded size. We will work backwards from b to 1 using the embedding of $G_{i+1} - A_{i+1}$ to help in constructing an embedding for $G_i - A_i$, if one exists. The key idea is that when we construct the embedding of G_{i+1}, we make a reduction to get a subgraph G'_{i+1} of G_{i+1}, which has bounded tree-width. The important property of G'_{i+1} is that the apex number $l \leq k$ of G_{i+1} is the exactly same as that of G'_{i+1}. Moreover, all the vertices in A^U must be in G'_{i+1}, and no vertex in $V(G_{i+1}) - V(G'_{i+1})$ is in A_P. Actually, every vertex v in $V(G_{i+1}) - V(G'_{i+1})$ is irrelevant, i.e, G_{i+1} is k-apex if and only if $G_{i+1} - v$ is.

For the reader's convenience, we shall give a sketch how to construct the graph G'_{i+1} from G_{i+1}. Suppose a planar embedding of $G_{i+1} - A_{i+1}$ is given for some apex vertex set A_{i+1} of order at most k. Then there are two steps to find the graph G'_{i+1}.

Step 1. Detecting the vertex set A^U for G'_{i+1}.

To find such a vertex set, we need an idea by Mohar [35].

Roughly, it says the following: Suppose the apex number of G is k, and let an apex vertex set be A with $|A| = k$. Suppose a planar embedding of $G - A$ is given. Let x be a vertex in A, and let U be the neighbors of x in $G - A$. A set $\mathcal{F}$ of facial walks of $G - A$ is a *face cover* of U if each vertex of U belongs to a member of $\mathcal{F}$. If a vertex x is contained in the apex vertex set A, then either

1) face-cover of its neighbors in $G - A$ is large, and the neighbors of x in $G - A$ are covered by bounded number of disks of $G - A$, each of whose graphs inside these disks is of small radius in $G - A$ (in which case, it is easy to fix the embedding because the graph in

the union of these disks has bounded tree-width), or

2) face-cover of its neighbors in $G-A$ is large, and there are many pairwise disjoint faces in $G-A$ that have a neighbor of x. Moreover, face-distance of any two of them is at least 7. (in which case, no matter how we embed this graph into a plane after deleting at most k vertices, x must be in the apex vertex set A. Thus x must be in A^U.).
3) face-cover of its neighbors in $G-A$ is small.

In the second case, we shall conclude that x must be in the set A^U because x is contained in at least $k+1$ copies of kuratowski graphs, i.e, a $K_{3,3}$-minor or a K_5-minor, that share only x (thus if x is not in the apex vertex set, then one of the kuratowski graphs still remains). Let us remark that the idea to detect the universal apex vertices, i.e, a vertex set A^U, was adapted in Graph Minors papers [44], [45] to detect the universal apex vertex set in the seminal Graph Minor structure theorem [45] (The universal apex vertex set corresponds to the "tips of horns" in the languages of [45]). Detecting the vertex set A^U in G is one of the keys in our algorithm, too.

Step 2. Performing the reduction.

Once we get the universal apex vertex set A^U, then we can perform the reduction in $G_{i+1}-A^U$ to get a subgraph of bounded tree-width. If $G_{i+1}-A^U$ has large tree-width, then there is a huge wall W, and almost all subwalls of W must be planarly embedded in $G_{i+1}-A^U$. Note that $G_{i+1}-A^U$ may not be planar, but each non-planar part must be local because there is no universal apex vertex in $G_{i+1}-A^U$. Thus large portion of $G_{i+1}-A^U$ has a planar embedding, except for small number of areas. In fact, we shall prove the following;

> There are only bounded number of disks in $G_{i+1}-A_{i+1}$, each of whose graphs inside them is of bounded radius, such that they cover all the neighbors of the vertices of $A_{i+1}-A^U$ in $G_{i+1}-A_{i+1}$.

Thus after deleting these disks and all the vertices in $A_{i+1}-A^U$, the resulting graph G' is a planar graph with the universal apex vertex set A^U.

We also prove that the deleted graph $G_{i+1}-G'$ (that consists of bounded number of planar graphs of bounded radius in the disks, together with all the vertices in $A_{i+1}-A^U$) has bounded tree-width. Intuitively, this is because each disk we have deleted in $G_{i+1}-A_{i+1}$ is a planar graph of bounded radius (and we have only bounded number of the deleted disks), thus the deleted graph $G_{i+1}-G'$ cannot have a huge wall. This fact is needed when we make a reduction in G'.

We now make a reduction to delete the vertices in deep inside the subwall of W in $G'-A^U$ to get a bounded tree-width graph. Let us remind that $G'-A^U$ has a planar embedding. Thus each subwall W' of W in $G'-A^U$ induces a planar graph, i.e, letting C be the outer face boundary of W', the graph drawn inside C is planar. In Section 3, we shall prove that each vertex v surrounded by a $2k$-wall is irrelevant, i.e, G_{i+1} is k-apex if and only if $G_{i+1}-v$ is. Thus each of them is not in A_P. It can be shown that after performing the reduction, we can destroy all the walls that have $2k$ nested cycles in $G'-A^U$. The resulting graph, together with the deleted graph $G_{i+1}-G'$ (that consists of bounded number of planar graphs of bounded radius in the disks, together with all the vertices in $A_{i+1}-A^U$), is G'_{i+1}. We shall prove that this graph G'_{i+1} has tree-width at most $h(k)$ for some function h of k. Intuitively, this is because each disk we have deleted to make up the graph G' has bounded radius, and we have deleted only bounded number disks (and hence the deleted graph $G_{i+1}-G'$ cannot have a huge wall), thus the graph G'_{i+1} cannot have a huge wall.

Let us observe that $G_{i+1}-G'_{i+1}$ consists of disjoint planar graphs such that each of them is contained in some disk in G'_{i+1}. In addition, each vertex in $V(G_{i+1})-V(G'_{i+1})$ is not in A^U nor in A_P. Actually, every vertex v in $V(G_{i+1})-V(G'_{i+1})$ is irrelevant, i.e, G_{i+1} is k-apex if and only if $G_{i+1}-v$ is. Since G'_{i+1} has tree-width at most $h(k)$, so we can compute the apex number of G'_{i+1} and all the vertices of G'_{i+1} that are in A_P. Since we only delete the vertices of G that are not in $A_P \cup A^U$, thus the apex number of G'_{i+1} is exactly same as that of G_{i+1}. Moreover, we can compute all the vertices of G_{i+1} that are in A_P (from the graph G'_{i+1}).

In the above overview, we did not care about the uncontraction of an induced matching. In this paper, we need to uncontract the induced matching M_i of G_i in G_{i+1} to obtain the graph G_i. Note that G_{i+1} may be obtained from G_i by deleting a stable set I of $\epsilon|G_i|$ vertices of degree l ($l \leq k+3$). In this case, it is actually easy to construct an embedding of G_i from G_{i+1} since all the deleted vertices of degree l are adjacent to all the vertices of a clique of order exactly l in G_{i+1}. In addition, for each vertex x in I, there are $k+3$ vertices of G_i that are not in the stable set I, but each of these vertices have exactly the same neighbors as x. Thus it is easy to fix the embedding of G_i from the embedding of G_{i+1}. Therefore, we shall only concentrate on the case when G_{i+1} is obtained from G_i by contracting the induced matching M_i.

It remains to find a subgraph G'_i of G_i with the property, which is exactly the same as that of G'_{i+1}, i.e., G'_i has tree-width at most $h(k)$, and the apex number $l \leq k$ of G_i is the exactly same as that of G'_i. Moreover, all the vertices in A^U must be in G'_i, and no vertex in $V(G_i)-V(G'_i)$ is in A_P. Actually, every vertex v in $V(G_i)-V(G'_i)$ is irrelevant, i.e, G_i is k-apex if and only if G_i-v is. Otherwise, we need to find a minimal forbidden minor for k-apex graphs in G_i. This is our main challenge in this algorithm, and we shall describe more details in the full version of this paper. We continue with this procedure until we reach $G = G_0$.

In order to achieve properties as claimed for our main algorithm, we need the following ingredients:

1) Finding an induced matching of large size. This was originated in Bodlaender [8].
2) Finding a universal apex vertex set. We need the idea by Mohar [35].
3) Reducing the tree-width by deleting irrelevant vertices.

For the second ingredient, Mohar [35] used the idea to prove that determining Euler genus of 1-apex graphs is NP-complete. For the third ingredient, we first use the idea in Graph Minors [44], [45]. We shall then use the technique in [40], [41]. The results in [40], [41] show that there is a linear time algorithm for the k disjoint paths problem for any fixed k when an input graph is planar. This algorithm handles planar graphs more quickly than the seminal algorithm of Robertson and Seymour in [44] which solves the same problem for arbitrary graphs in cubic time. The proof in [40], [41] uses several ideas underlying Robertson and Seymour's algorithm.

This paper is organized as follows. In Section 2, we discuss how to detect the universal apex vertex set (and its relation to face-cover). This section contains one of the most important ingredients in the algorithm of Theorem 1.1. In Section 3, we show how to delete irrelevant vertices quickly to reduce the tree-width. This technique was also used in [25], [26], [27] to design linear time algorithms for some graph embedding problems. In Section 4, we give our reduction theorem. In Section 5, we describe our linear time algorithm for Theorem 1.1. In Section 6, we prove Theorem 1.2, and give more applications.

2. Face Cover and Universal Apex

Suppose the apex number of G is k, and let $A = \{x_1, \dots, x_k\}$ be an apex vertex set. Thus $G - A$ has a planar embedding. In this section, we shall look at face-cover of the neighbors of each x_i in $G - A$. We need some definition.

A *facial walk* of a face is a walk taken around the face in clockwise order, recording the edges and vertices bordering the face as they are encountered. Given a graph Z embedded in a plane and a subset U of its vertices, a set $\mathcal{F}$ of facial walks of Z is a *face cover* of U if each vertex of U belongs to a member of $\mathcal{F}$. Let $\tau(U)$ be the size of the smallest face cover of U. It is known (see [35]) that computing $\tau(U)$ is NP-complete if U is as part of the input.

Face-distance of two points u, w (not necessarily vertices) in a plane graph G is the number of faces that the shortest curve between u and w in G passes. *Face-distance radius* can be defined as "radius" in a usual sense, except distance is replaced by face-distance.

Our purpose of this section is to prove the following.

> For each $x_i \in A$, if face-cover of its neighbors in $G - A$ is large, and there are many disjoint faces that have a neighbor of x_i in $G - A$ (and moreover, face-distance between any two of them is at least 7), then $x_i \in A^U$.

Intuitively, this is because x_i is contained in $k + 1$ copies of kuratowski graphs, i.e, a $K_{3,3}$-minor or a K_5-minor, that share only x_i. Thus if x_i is not in the apex vertex set, then one of the kuratowski graphs still remains. Thus $x_i \in A^U$.

We now try to detect all the vertices in A^U. Detecting the universal apex set A^U is a significant step in our algorithm, as this allows us to make a reduction, see overview. This idea is inspired by the arguments in the proof of the main Graph Minor structure theorem [45]. (The universal apex vertex set corresponds to the "tips of horns" in the languages of [45]). Detecting the universal apex vertices is one of the keys in our algorithm, too.

We need the following result in [35].

Theorem 2.1: Let G be a 3-connected planar graph and let F be a collection of facial cycles of G. Then F contains a subset F' such that any two cycles in F' either is disjoint or shares at most one vertex, and $|\mathcal{F}'| \leq \frac{|\mathcal{F}|}{40}$.

We now prove the following result:

Theorem 2.2: Let G be a 3-connected plane graph and let $U \subseteq V(G)$. We now add a vertex v to G so that v is adjacent to all the vertices in U. Let G' be the resulting graph. Suppose there are at least $2k + 1$ pairwise disjoint faces F_i that are needed to cover all the vertices in U in G. Then either

1) no matter how we delete k vertices from G, there is a kuratowski graph, i.e, a K_5-minor or a $K_{3,3}$-minor, in the resulting graph of G', or
2) there are at most $2k$ disks in G, each of whose graphs inside them is of face-distance radius at most 8, such that they cover all the vertices in U.

Proof of Theorem 2.2. In order to prove it, we need the following lemma. If C is a cycle in a graph on a plane, then $int(C)$ denotes the set of vertices and edges inside C (but not on C). By *inside* here, we will specify in each case. The union of $int(C)$ and C is denoted $Int(C)$. Hereafter, a *kuratowski graph* means either a K_5-minor or a $K_{3,3}$-minor.

We need one more lemma, which is easy to prove.

Lemma 2.3: Suppose G is a 3-connected planar graph. Let C_1, C_2, C_3 be three disjoint cycles such that C_1 is a face in $int(C_2)$ (and in G, too), and $Int(C_2)$ is a subgraph of $int(C_3)$. (Thus, C_1, C_2, C_3 are nested cycles in this order.) Let x be a vertex in C_1, and y be a vertex in C_3. If we add the edge xy, then $Int(C_3)$ together with the edge xy gives rise to a non-planar graph, and hence it contains a kuratowski graph.

Proof. Let $R = Int(C_3) + xy$. Since G is 3-connected, there are three disjoint paths P_1, P_2, P_3 from $V(C_1)$ to $V(C_3)$ in $Int(C_3)$. By a result in [28] (or see [16]), the paths

P_1, P_2, P_3 and the cycle C_2 can be chosen so that each path of P_1, P_2, P_3 intersects C_2 by a path (possibly only one vertex). Then it is easy to see that this configuration together with the edge xy, for any vertex x in C_1 and for any vertex y in C_3, gives rise to either a K_5-minor or a $K_{3,3}$-minor (hence a kuratowski graph). □

We are now ready to prove Theorem 2.2. Until the end of the proof, we shall focus on the graph G. We now greedily construct disjoint disks $D_1, \dots, D_l$ such that each disk D_i has a center in a face F_i that contains a vertex in U, and in addition, each disk D_i contains all the vertices whose face-distance is at most 4 from the center in F_i. (Let us observe that the center here is a point contained in the face F_i.) We also assume that each disk does not contain a vertex whose face-distance from the center of the disk is at least 5. Since G is 3-connected, each disk D_i contains three disjoint cycles C_1, C_2, C_3 that satisfy the assumption of Lemma 2.3 with $C_1 = F_i$. We take such disjoint disks $D_1, \dots, D_l$ so that no more such a disk is possible to create.

Suppose $l \leq 2k$. Then by our construction, every face, which is outside these disks $D_1, \dots, D_l$ and has a vertex in U, has face-distance at most 4 from some disk D_i. Then we can add all these faces (call them F'), together with the faces that have face-distance at most three from a face in F', and the faces that have face-distance at most three from a face in some of disks $D_1, \dots, D_l$, to some disks in $D_1, \dots, D_l$, so that all the vertices in the resulting disk D_i have face-distance at most 8 from the center of D_i. In this way, all the faces that have a vertex in U are contained in some of the new disks $D_1, \dots, D_l$. Then we are done, as the second conclusion of Theorem 2.2 holds.

Suppose finally $l \geq 2k+1$. It is well-known that every 3-connected planar graph is $\frac{1}{2}$-tough (see [6]). Thus no matter how we delete k vertices from G, there are two disks of $D_1, \dots, D_l$, say D_1, D_2, such that they do not contain any deleted vertex, and they are in the same component R of the resulting graph. Moreover, the disk D_2 contains three disjoint cycles C_1, C_2, C_3 that satisfy the assumption of Lemma 2.3 with $C_1 = F_2$. By contracting vertices in $V(R) - V(Int(C_3))$ to the cycle C_3 in the disk D_2, and then contracting the vertex v to C_3 as well, we can get the configuration as described in Lemma 2.3. So, no matter how we delete k vertices from G, the resulting graph of G' has a kuratowski graph that involves the vertex v. This completes the proof of Theorem 2.2. □

We now prove our main theorem in this section.

Theorem 2.4: There is a nondecreasing integer function $f_1 : \mathbb{N} \to \mathbb{N}$ such that the following holds. Let G be a 3-connected plane graph and let $U \subseteq V(G)$. We now add a vertex v to G so that v is adjacent to all the vertices in U. Let G' be the resulting graph. Suppose there are at least $f_1(k)$ faces that are needed to cover all the vertices in U in G. Then either

1) no matter how we delete k vertices from G, there is a kuratowski graph, which involves the vertex v in the resulting graph of G', or
2) there are at most $2k$ disks in G, each of whose graphs inside them is of face-distance radius at most 9, such that they cover all the vertices in U.

Proof of Theorem 2.4. Until the end of the proof, we shall focus on the graph G. Set $f_1(l) \geq 80k+40$. By Theorem 2.1, there are at least $\frac{|f_1(k)|}{40} \geq 2k+1$ faces that contain at least one vertex in U such that any two faces either is disjoint or shares at most one vertex. Let F be such faces. We may assume that each face that contains a vertex in U, but is not contained in F, shares at least one vertex with a face in F. If there are at least $2k+1$ pairwise disjoint faces in F, then by Theorem 2.2, we get a desired conclusion. Otherwise, we now greedily construct disjoint disks $D_1, \dots, D_l$ such that each disk D_i has a center in a face F_i in F, and in addition, each disk D_i contains all the vertices whose face-distance is at most 4 from the center in F_i. (Let us observe that the center here is a point contained in the face F_i.) We also assume that each disk does not contain a vertex whose face-distance from the center of the disk is at least 5. We take such disks $D_1, \dots, D_l$ so that no more disk is possible to create. As we did in the proof of Theorem 2.2, if $l \geq 2k+1$, we are done by Lemma 2.3. Otherwise, as we did in the proof of Theorem 2.2, there are at most $2k$ disks, each of whose graphs inside them is of face-distance radius at most 8, such that they cover all the faces in F. We now need to add all the faces that contain a vertex in U, but are not contained in F. As remarked above, each of those faces must share a vertex with a face in F. Thus by adding these faces to some disks, we can get at most $2k$ disks $D_1, \dots, D_l$ ($l \leq 2k$), each of whose face-distance radius is at most 9, such that they cover all the vertices in U. □

3. Bounding the tree-width inside flat parts

Suppose that G is k-apex and the universal apex vertex set A^U for G is given. In this section, we are given a planar subgraph Q of $G - A^U$. We want to bound its tree-width by deleting many vertices at once, and our goal is to do this in linear time. Moreover, we want that the deleted vertex set U is *irrelevant* in the sense that $U \cap A_P = \emptyset$, and in addition, no matter how we embed $G - U$ into a plane after deleting at most k vertices, the vertices in U can be put back to this planar embedding so that the resulting embedding is still planar. We prove the following result.

Theorem 3.1: Suppose the universal apex vertex set A^U for G is given, where $|A^U| \leq k$. Suppose that $G - A^U$ contains a planar subgraph Q and that C is the outer cycle of a planar embedding of Q. Suppose also that for every vertex in $Q \setminus C$, all its neighbors in the graph $G - A^U$ are contained in Q. If Q contains an h-wall W, where $h \geq 2k+1$, then the set X of vertices of W, that have face-distance in the wall W at least k from the outer cycle of W, is irrelevant.

In particular, every vertex surrounded by a $2k$-wall in Q is irrelevant. Furthermore, no matter how we embed $G-A-X$ into a plane for any apex vertex set A, where $A_U \subseteq A$ and $|A| \leq k$, the embedding of $G-A-X$ can be changed at vertices of Q such that any vertex in X is contained in a disk of the embedding of $G-A-X$. Furthermore, we can put all the vertices of X back to the above mentioned disk so that the resulting embedding of $G-A$ is still planar.

Proof. Suppose $G-A$ has an embedding in a plane. Since $|A| \leq k$, thus $W-A$ still has a k-wall W'. The embedding of the wall W' in a plane is unique, since W' is a subdivision of a 3-connected graph (it is well known that every 3-connected planar graph has a unique embedding). Thus each face of the wall W' bounds a disk. This implies that no matter how we embed $G-A-X$ into a plane, the wall W' induces a planar graph, i.e, the outer face boundary of W' bounds a disk D in $G-A-X$ (so the subgraph of $G-A-X$ embedded in this disk is a plane graph). In particular, the embedding in the disk D can be changed (using Whitney switch, see [37]. Note that given two planar embeddings M, N of a planar graph, M can be obtained from N by repeatedly applying Whitney switch.), so that it becomes a subembedding of Q. Thus for each vertex x in X, there is a face in $G-A-X$ that corresponds to a disk D' of Q that contains x. Moreover, since the graph embedded in this disk D' is a plane graph in Q, thus x can be put back to this disk D', so that the resulting embedding of $G-A$ is still planar. Thus the vertices in X are irrelevant. □

Theorem 3.1 says that for any planar subgraph Q (in $G-A^U$, where G is k-apex), if there is a $(2k+1)$-wall in Q, then we can delete the middle vertex which is irrelevant, and actually, we can keep deleting irrelevant vertices until there is no $(2k+1)$-wall in the resulting graph of Q. So, as long as there is a $2k$-wall in Q, we can delete all the vertices deep inside this $2k$-wall. The problem here is that, how can we perform this operation in linear time? Fortunately, there is a way to do it. This method was first adapted by Reed, Robertson, Seymour and Schrijver [40], [41], who proved that there is a linear time algorithm for the k disjoint paths problem for planar graphs (for any fixed k). So we shall use this method to delete vertices until the resulting graph has no $(2k+1)$-wall. Let us state this as a lemma.

Lemma 3.2: Let G, A^U, k be as in Theorem 3.1. Suppose that $G-A^U$ contains a planar subgraph Q and that C is the outer cycle of a planar embedding of Q. Suppose also that for every vertex in $Q \setminus C$, all its neighbors in the graph $G-A^U$ are contained in Q. Then, given the graph Q, there is a linear time algorithm to find a maximal vertex set $X \subseteq V(Q)$ such that

1) the graph $Q-X$ does not contain a $(2k+1)$-wall,
2) deleting the vertices of X from $G-A^U$ does not change the problem of finding an embedding of $G-A$ into a plane for any apex vertex set A, where $A^U \subseteq A$, $|A| \leq k$ and $A \cap X \neq \emptyset$, by a sequence of applications of Theorem 3.1, and
3) no matter how we embed $G-A-X$ into a plane for any apex vertex set A, where $A^U \subseteq A$, $|A| \leq k$ and $A \cap X \neq \emptyset$, the embedding of $G-A-X$ can be changed at vertices of Q such that each vertex in X is contained in some disk of the embedding of $G-A-X$, and we can put all the vertices of X back to some of the above mentioned disks so that the resulting embedding of $G-A$ is still planar.

(by a *maximal* vertex set here, we mean that any vertex of $Q-X$ that is adjacent to a vertex in X is not surrounded by a $2k$-wall in $Q-X$. It follows that each vertex in $Q-X$ that is adjacent to a vertex in X is surrounded by a $(2k-1)$-wall.)

The proof of Lemma 3.2 is inspired by Reed, Robertson, Schrijver and Seymour's work [40], [41]. In fact, Lemma 3.2 was used in [25], [26], [27] to design linear time algorithms for the graph embedding problems. Let us observe that when Lemma 3.2 is needed, the universal vertex set A^U and the planar subgraph Q are already given. Thus we do not have to take the universal apex set A^U into account. We just need to focus on the planar graph Q.

We shall need the following extension of Lemma 3.2.

Lemma 3.3: Let $l \geq 0$ be a fixed integer and G, A^U, k be as in Theorem 3.1. Suppose that $G-A^U$ contains a planar subgraph Q and that C is the outer cycle of a planar embedding of Q. Suppose also that for every vertex in $Q \setminus C$, all its neighbors in the graph $G-A^U$ are contained in Q. Let $C_1, \ldots, C_l$ be disjoint faces of Q. Then, given the graph Q, there is a linear time algorithm to find a maximal vertex set $X \subseteq V(Q)$ such that

1) each vertex in X is surrounded by a $2k$-wall in $Q-X$ that does not contain any face of $C_1, \ldots, C_l$,
2) deleting the vertices of X from $G-A^U$ does not change the problem of finding an embedding of $G-A$ into a plane for any apex vertex set A, where $A^U \subseteq A$, $|A| \leq k$ and $A \cap X \neq \emptyset$, by a sequence of applications of Theorem 3.1,
3) $Q-X$ contains no $2kl$-walls, and
4) no matter how we embed $G-A-X$ into a plane for any apex vertex set A, where $A^U \subseteq A$, $|A| \leq k$ and $A \cap X \neq \emptyset$, the embedding of $G-A-X$ can be changed at vertices of Q such that each vertex in X is contained in some disk of the embedding of $G-A-X$, and we can put all the vertices of X back to some of the above mentioned disks so that the resulting embedding of $G-A$ is still planar.

(by a *maximal* vertex set here, we mean that any vertex of $Q-X$ that is adjacent to a vertex in X is not surrounded by a $2k$-wall that does not contain any face of $C_1, \ldots, C_l$ in $Q-X$. It follows that each vertex in $Q-X$ that is adjacent to a vertex in X is surrounded by a $(2k-1)$-wall that does not contain any face of $C_1, \dots, C_l$ in $Q - X$.)

The proof of Lemma 3.3 is similar to the proof of Reed, Robertson, Schrijver and Seymour, and that of Lemma 3.2. Lemma 3.3 was also used in [25], [26], [27] to design linear time algorithms for some graph embedding problems.

4. Finding a relevant subgraph of bounded tree-width

Let us recall that our algorithm produces first a sequence of graphs $G = G_0, G_1, \dots, G_b$, where each G_{i+1} is obtained from G_i either by contracting a large induced matching M_i or by deleting a large set of vertices of degree $l \leq k + 3$ (in which case we shall write $M_i = \emptyset$). In this section, an integer i is fixed such that $M_i \neq \emptyset$ and the following hypothesis is assumed.

Hypothesis 4.1: The apex number of the graph G_{i+1} is $l \leq k$. The graph G_i is obtained from G_{i+1} by uncontracting the matching M_i and $|M_i| \geq \epsilon |G_i|$, where $\epsilon > 0$ is some small but fixed constant. Moreover, we have a subgraph G'_{i+1} of G_{i+1}, and the set A^U_{i+1} satisfying the following conditions:

1) A^U_{i+1} is the universal apex set for G_{i+1}, and the apex number of G'_{i+1} is l. Actually, we slightly modify the definition of the universal apex vertex set A^U_{i+1}. Each vertex in A^U_{i+1} is obtained as in the first conclusion of Theorem 2.4.
2) G'_{i+1} has tree-width at most $h(k)$ for some function h of k (to be determined later, but $h(k) = O(k^3)$).
3) $G_{i+1} - G'_{i+1}$ is planar. More precisely, it consists of disjoint disks $D_1, D_2, \dots, D_l$ for some l, and all the vertices in these disks are irrelevant for the graph G_{i+1}, and hence they are not in A_P for G_{i+1}.

The objective of this section is to start with $G_{i+1}, G'_{i+1}, A^U_{i+1}$ satisfying the hypothesis, and then to either construct G'_i satisfying the hypothesis for i with the universal apex vertex set A^U_i, or to find a minimal forbidden minor for k-apex graphs in time $O(|G_i|)$.

A proof for constructing G'_i, A^U_i will be given in the full paper.

5. Algorithm

Finally, we are ready to present the complete algorithm.

Algorithm for Theorem 1.1

We assume a positive integer k.

Input: A graph G of order n.

Output: Either an embedding of G in a plane after deleting at most $l \leq k$ vertices or a minor of G which is not k-apex and is minimal with this property, where l is the apex number of G.

Running time: $O(f(k)n)$ for some function $f : \mathbb{N} \to \mathbb{N}$.

Description:

Initially, we delete all vertices of degree at most 1. Hereafter, we assume that G has minimum degree at least 2.

Step 1. Find a sequence of graphs $G = G_0, G_1, ..., G_b$ such that G_i is obtained from G_{i-1} by either contracting an induced matching M_{i-1} with at least $\epsilon |G_{i-1}|$ edges for some small but constant $\epsilon > 0$, or deleting a stable set I of $\epsilon |G_{i-1}|$ vertices, each of degree $l \leq k+3$. In the latter case, for every vertex x in I, the following holds:

1) x has the same neighbors as at least l other vertices in I.
2) In addition, there are $k+3$ vertices in G_{i-1} that are not in I such that each of them has exactly same neighbors as x.

In this case, we also put a clique to G_{i-1} for the neighbors of every $x \in I$.

This step can be done (details will be discussed in the full paper). If we find a minimal forbidden minor for k-apex graphs, we stop and output the minor. Otherwise, we keep doing it b steps, where b is minimum integer such that G_b has fewer than B vertices for some constant B. Then $b \leq log_{1/\epsilon} n$ and the sum of the sizes of all G_i is $O(n)$.

At each step i, we can either find a desired induced matching or a desired stable set in time $O(|G_i|)$. A short computation implies that we can do it in linear time.

Step 2. Apply a brute force algorithm to find either an embedding of G_b or a minimal forbidden minor for k-apex graphs in G_b. Since $|G_b| < B$, this can be done in constant time.

We recursively apply Step 3 for $i = b, b-1, \dots$.

Step 3. For the ith iteration, either find a subgraph G'_i of G_i satisfying Hypothesis 4.1 or a minimal forbidden minor for k-apex graphs. We also need to detect the universal apex vertex set A^U_i as in Hypothesis 4.1.

Step 4. Extend the embedding of G'_0 to $G_0 = G$.

This can be done in time $O(V(|G|))$ by applying the planarity algorithm [7], [10], [24], [48]. Note that all vertices of $G - G'_0$ can be embedded into disks $D_1, D_2, \dots$ in any planar embedding of $G'_0 - A^U_0$. Let us observe that any vertex in A^U_0 is contained in $V(G'_0)$. Therefore, we just need the planarity algorithm for this task, and hence we can do this in linear time.

Since $b \leq log_{1/\epsilon} n$ and the sum of the sizes of the G_i is $O(n)$, we can get a linear time algorithm to output a desired conclusion in Theorem 1.1.

Let us finally estimate the constant $f(k)$. The expensive part is the sum of the sizes of the G_i, which can be written as $c(\epsilon)n$ for some function c of ϵ. Another expensive part is to construct the graph G'_i, which takes $O(c'(k)|G_i|)$, where $c'(k)$ is double exponential of k. Essentially, the most expensive part is the dynamic programming part. It was mentioned in [32] that the time complexity of this part is $O(f'(k)n)$, where f' is double exponential of k. The tree-width bound in constructing the graph G'_i is $O(k^9)$. Therefore, $f(k)$ only depends on $c(f'(k))$, and the algorithm shows that $f(k) = Poly(f'(k))$, because $c(\epsilon)$ is single exponential. Thus, $f(k)$ is double exponential of k as we claimed. This completes the proof of Theorem 1.1. □

6. Some Applications

6.1. TSP and related problems

To obtain the linear time approximation schemes for weighted TSP and minimum c-edge-connected submultigraph for Theorem 1.2, as well as general family of approximation algorithms for contraction-closed problems, we study a structural decomposition problem introduced in [14]: partition the edges of a graph into l pieces such that contracting any one of the pieces results in a bounded-treewidth graph (where the bound depends on l). Such a result has been obtained for bounded-genus graphs [14] and for planar graphs with a variation of contraction called compression (deletion in the dual graph) [29]. In fact, the following result is already proved in [14] for k-apex graphs:

Theorem 6.1: Suppose an embedding of a k-apex graph is given for any fixed k. For any integer $l \geq 2$, the edges of G can be partitioned into l sets such that contracting any one of the sets results in a graph of treewidth at most $f(k,l)$ (for some function f of k,l). Furthermore, such a partition can be found in linear time.

Let us observe that the most expensive part in the proof of Theorem 1.1 in [14] is to find a shortest non-contractible cycle in the fixed surface. On the other hand, we only need Theorem 1.1 in [14] for planar graphs, which can be done in linear time (this was observed in [14], but the proof essentially follows from [29]). Note that dealing with the edges with one endvertex in the apex vertex set is easy. Thus Theorem 6.1 follows. In [14], it is shown how Theorem 6.1 leads to a general family of linear time approximation schemes for any contraction-closed problem satisfying a few simple criteria, including weighted TSP and minimum c-edge-connected submultigraph. Thus Theorem 6.1, together with Theorem 1.1, which gives an embedding of a given k-apex graph in linear time (for any fixed k), proves Theorem 1.2. Many other classic problems are contraction-closed, for example, dominating set (and its many variations) and minimum chordal completion. The same proof of Theorem 1.2 implies that there are linear time $(1+\epsilon)$-approximation algorithms for these problems.

6.2. Other applications

A fundamental way to design graph algorithms is using Lipton and Tarjan's divide-and-conquer separation decomposition for planar graphs [31]. A generalization of this decomposition approach leads to PTASs for many minimization and maximization problems, such as vertex cover, minimum color sum, and hereditary problems such as independent set and max-clique [4], [19]. This approach can be used to k-apex graphs. Firstly, Theorem 1.1 implies that there is a linear time algorithm to find an embedding of a k-apex graph G (for any fixed k). Secondly, after deleting the apex vertex set A (of order at most k), we can find a separator in $G - A$ by a result in [31]. Thus this separator, together with the apex vertex set A, gives rise to a separator in G. This separator allows us to use (and generalize) Lipton and Tarjan's divide-and-conquer separation decomposition [31] for k-apex graphs. Thus all the applications that are carried out in [4], [19] can be extended to k-apex graphs. In fact, the following result was proved in [13]:

Theorem 6.2: Suppose an embedding of a k-apex graph is given for any fixed k. For any integer $l \geq 2$, the vertices of G can be partitioned into l sets such that deleting any one of the sets results in a graph of tree-width at most $f(k,l)$ (for some function f of k,l). Furthermore, such a partition can be found in linear time.

Note that in [13], Theorem 6.2 was obtained for "almost" embeddable graphs (for the definition of "almost" embeddable graphs, see in [13]), which include k-apex graphs. In [13], it is shown how Theorem 6.2 leads to a general family of linear time approximation schemes for many minimization and maximization problems. Thus, together with Theorem 1.1, which gives an embedding of a given k-apex graph in linear time (for any fixed k), we get linear time approximation schemes for many such minimization and maximization problems for k-apex graphs, such as vertex cover, minimum color sum, and hereditary problems such as independent set and max-clique.

References

[1] S. Arnborg and A. Proskurowski, Linear time algorithms for NP-hard problems restricted to partial k-trees, *Discrete Appl. Math.* **23** (1989), 11–24.

[2] S. Arora, M. Grigni, D. Karger, P. Klein and A. Woloszyn, A polynomial-time approximation scheme for weighted planar graph TSP, *Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'98)*, 33–41, (1998).

[3] I. Adler, M. Grohe and S. Kreutzer, Computing excluded minors, *Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'08)*, 641–650, (2008).

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

[5] A. Berger, A. Czumaj, M. Grigni and H. Zhao, Approximation schemes for minimum 2-connected spanning subgraphs in weighted planar graphs, *Proc. 13th Annual European Symposium on Algorithms*, **3669**, *Lecture Notes in Computer Science*, 472–483, (2005).

[6] J. A. Bondy and U. S. R. Murty, Graph Theory, Springer, 2008.

[7] K. S. Booth and G.S. Lueker, Testing for the consecutive ones property, interval graphs and graph planarity using PQ-trees, *J. Comput. System Sci.*, **13** (1976), 335–379.

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

[9] S. Cabello and B. Mohar, Problem of months, Fall 2005, http://www.fmf.uni-lj.si/~mohar/

[10] N. Chiba, T. Nishizeki, S. Abe and T. Ozawa, A linear time algorithm for embedding planar graphs using PQ-trees, *J. Comput. System Sci.*, **30** (1985), 54–76.

[11] A. Czumaj, M. Grigni, P. Sissokho and H. Zhao, Approximation schemes for minimum 2-edge-connected and biconnected subgraphs in planar graphs. *Proc. 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'04)*, 496–505, (2004).

[12] E. D. Demaine, F. Fomin, M. Hajiaghayi, and D. Thilikos, Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs, *J. ACM*, **52** (2005), 1–29.

[13] E. D. Demaine, M. Hajiaghayi, and K. Kawarabayashi, Algorithmic graph minor theory: Decomposition, approximation and coloring, *Proc. 46th Symposium on Foundations of Computer Science, (FOCS'05)*, 637–646, (2005).

[14] E. D. Demaine, M. Hajiaghayi, and B. Mohar, Approximation algorithms via contraction decomposition, *Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'07)*, 278–287, (2007).

[15] R. Diestel, K. Yu. Gorbunov, T. R. Jensen, and C. Thomassen, Highly connected sets and the excluded grid theorem, *J. Combin. Theory Ser. B*, **75** (1999), 61–73.

[16] R. Diestel, K. Kawarabayashi and P. Wollan, The Erdős-Pósa property for clique minors in highly connected graphs, *submitted*.

[17] R.G. Downey and M.R. Fellows, Parameterized complexity, Springer-Verlag, 1999.

[18] D. Eppstein, Subgraph isomorphism in planar graphs and related problems, *J. Graph Algor. Appl.*, **3** (1999), 1–27.

[19] D. Eppstein, Diameter and tree-width in minor-closed graph families, *Algorithmica*, **27** 275–291, (2000).

[20] M.R. Fellows and M. A. Langston, On serach, decision and the efficiency of polynomial time algorithms (extended abstract), *Proc. 21th ACM Symposium on Theory of Computing (STOC'89)*, 501–512, (1989).

[21] M. Grigni, E. Koutsoupias, and C. Papadimitriou, An approximation scheme for planar graphs TSP, *Proc. 36th Symposium on Foundations of Computer Science, (FOCS'95)*, 640–646, (1995).

[22] M. Grohe, Local tree-width, excluded minors, and approximation algorithms. *Combinatorica* **23**, (2003), 613–632.

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

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

[25] K. Kawarabayashi and B. Reed, Computing crossing number in linear time, *Proc. 39th ACM Symposium on Theory of Computing (STOC'07)*, 382–390, (2007).

[26] K. Kawarabayashi and B. Mohar, Graph and Map Isomorphism and all polyhedral embeddings in linear time, *Proc. 40th ACM Symposium on Theory of Computing (STOC'08)*, 471–480, (2008).

[27] K. Kawarabayashi, B. Mohar and B. Reed, A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of bounded tree-width graphs, *Proc. 49th Annual Symposium on Foundations of Computer Science (FOCS'08)*, 771–780, (2008).

[28] K. Kawarabayashi, S. Norine, R. Thomas and P. Wollan, K_6 minors in large 6-connected graphs, *submitted*.

[29] P. N. Klein, A linear time approximation scheme for TSP for planar wieghted graphs, *Proc. 46th Symposium on Foundations of Computer Science (FOCS'05)*, 146–155, (2005).

[30] J. M. Lewis and M. Yannakakis, The vertex-deletion problem for hereditary propeties is NP-complete, *J. Comput. System Sci.*, **20** (1980), 219–230.

[31] R.J. Lipton and R. E. Tarjan, Applications of a planar separator theorem, *Siam J. Comput.*, **9** (1980), 615–627.

[32] D. Marx and I. Schlotter, Obtaining a planar graph by vertex deletion, *Proc. the 33rd Workshop on Graph-Theoretic Concepts in Computer Scienece*, (2007), 292–303.

[33] B. Mohar, Embedding graphs in an arbitrary surface in linear time, *Proc. 28th ACM Symposium on Theory of Computing (STOC'96)*, 392–397, (1996).

[34] B. Mohar, A linear time algorithm for embedding graphs in an arbitrary surface, *SIAM J. Discrete Math.*, **12** (1999), 6–26.

[35] B. Mohar, Face covers and the genus of apex graphs, *J. Combin. Theory Ser. B*, **82** (2001), 102–117.

[36] B. Mohar, A. Schrijver, Blocking nonorientability of a surface, *J. Combin. Theory Ser. B* **87** (2003), 2–16.

[37] B. Mohar and C. Thomassen, Graphs on Surfaces, Johns Hopkins University Press, Baltimore, MD, 2001.

[38] L. Perković and B. Reed, An improved algorithm for finding tree decompositions of small width, *International Journal on the Foundations of Computing Science*, **11** (2000), 81–85.

[39] B. Reed, Tree width and tangles: a new connectivity measure and some applications, in *"Surveys in Combinatorics, 1997 (London)"*, London Math. Soc. Lecture Note Ser. **241**, Cambridge Univ. Press, Cambridge, 87–162, (1997).

[40] B. Reed, Rooted Routing in the Plane, *Discrete Appl. Math.*, **57** (1995), 213–227.

[41] B. Reed, N. Robertson, A. Schrijver and P. D. Seymour, Finding disjoint trees in planar graphs in linear time. Graph structure theory (Seattle, WA, 1991), 295–301, Contemp. Math. 147, Amer. Math. Soc., Providence, RI, 1993.

[42] B. Reed and D. Wood, A linear time algorithm to find a separator in a graph with an excluded minor, *submitted*.

[43] N. Robertson and P. D. Seymour, Graph minors. V. Excluding a planar graph, *J. Combin. Theory Ser. B*, **41** (1986), 92–114.

[44] N. Robertson and P. D. Seymour, Graph minors. XIII. The disjoint paths problem, *J. Combin. Theory Ser. B*, **63** (1995), 65–110.

[45] N. Robertson and P. D. Seymour, Graph minors. XVI. Excluding a non-planar graph, *J. Combin. Theory Ser. B*, **89** (2003), 43–76.

[46] N. Robertson, P. D. Seymour and R. Thomas, Quickly excluding a planar graph, *J. Combin. Theory Ser. B*, **62** (1994), 323–348.

[47] C. Thomassen, The graph genus problem is NP-complete, *J. Algorithms* **10** 458–476, (1988).

[48] S. G. Williamson, Depth-first search and Kuratowski subgraphs, *J. ACM* **31** (1984), 681–693.