Graph and Map Isomorphism and All Polyhedral Embeddings 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

Bojan Mohar†
Department of Mathematics
Simon Fraser University
Burnaby, B.C. V5A 1S6
mohar@sfu.ca

ABSTRACT

For every surface S (orientable or non-orientable), we give a linear time algorithm to test the graph isomorphism of two graphs, one of which admits an embedding of face-width at least 3 into S. This improves a previously known algorithm whose time complexity is $n^{O(g)}$, where g is the genus of S. This is the first algorithm for which the degree of polynomial in the time complexity does not depend on g.

The above result is based on two linear time algorithms, each of which solves a problem that is of independent interest. The first of these problems is the following one. Let S be a fixed surface. Given a graph G and an integer $k \geq 3$, we want to find an embedding of G in S of face-width at least k, or conclude that such an embedding does not exist. It is known that this problem is NP-hard when the surface is not fixed. Moreover, if there is an embedding, the algorithm can give all embeddings of face-width at least k, up to Whitney equivalence. Here, the face-width of an embedded graph G is the minimum number of points of G in which some non-contractible closed curve in the surface intersects the graph. In the proof of the above algorithm, we give a simpler proof and a better bound for the theorem by Mohar and Robertson concerning the number of polyhedral embeddings of 3-connected graphs.

The second ingredient is a linear time algorithm for map isomorphism and Whitney equivalence. This part generalizes the seminal result of Hopcroft and Wong that graph isomorphism can be decided in linear time for planar graphs.

Received November 17, 2007, revised March 7, 2008.

*Research partly supported by JSPS Postdoctoral Fellowship for Research Abroad. Visiting Department of Mathematics, Simon Fraser University, Burnaby, B.C.

†Supported in part by the ARRS, Research Program P1-0297, by an NSERC Discovery Grant, and by the Canada Research Chair Program. On leave from Department of Mathematics, University of Ljubljana, Slovenia.

STOC'08, May 17–20, 2008, Victoria, British Columbia, Canada.

Categories and Subject Descriptors

G.2 [**Discrete Math**]: Combinatorics; G.2.2 [**Graph Theory**]: Combinatorics—*Graph algorithms, Computations on discrete structures*

General Terms

Algorithms, Theory

Keywords

Graph isomorphism, Map isomorphism, Linear time algorithm, Surface, Face-width, Polyhedral embedding

1 The Graph Isomorphism Problem

The graph isomorphism problem asks whether or not two given graphs are isomorphic. It is one of the most fundamental problems in the theory of algorithms and in complexity theory. It is probably the most notorious problem whose algorithmic complexity is still largely undecided. While some complexity theoretic results indicate that this problem is not NP-complete (if it were, the polynomial hierarchy would collapse to its second level, see [13, 8, 26, 27, 65]), no polynomial time algorithm is known for it, even with extended resources like randomization or quantum computing.

On the other hand, there is a number of important classes of graphs on which the graph isomorphism problem is known to be solvable in polynomial time. For example, in 1990, Bodlaender [9] gave a polynomial time algorithm for graph isomorphism of graphs of bounded tree-width. Many NP-hard problems can be solved in polynomial time, even linear time, when input is restricted to graphs of tree-width at most k [3, 10]. So, Bodlaender's result may not be surprising, but the time complexity in [9] is $O(n^{k+2})$, and no one could improve the time complexity to $O(n^{O(1)})$ so far. This indicates that even for graphs of bounded tree-width, the graph isomorphism problem is not trivial at all.

In this paper, we are interested in planar graphs and, more generally, graphs of bounded genus. In 1966, Weinberg [71] gave a very simple $O(n^2)$ algorithm for the graph isomorphism problem of planar graphs. This was improved by Hopcroft and Tarjan [32, 33] to $O(n \log n)$. Building on this earlier work, Hopcroft and Wong [30] published in 1974 a seminal paper, where they presented a linear time algorithm for isomorphism testing of planar graphs.

Leaving the plane to consider graphs on surfaces of higher genus, the graph isomorphism problem seems much harder.

In 1980, Filotti, Mayer [24] and Miller [44] showed that for every orientable surface S, there is a polynomial time algorithm for testing the isomorphism of graphs that can be embedded in S, but the time complexity is $n^{O(g)}$, where g is the genus of S. Lichtenstein [40] gave an $O(n^3)$ algorithm for testing graph isomorphism on projective planar graphs. These works came out in the early 1980's. These classes of graphs were extensively studied from other perspectives. For example, Grohe and Verbitsky [28, 29], who studied this problem from a logic point of view, made some interesting progress. However, no one could improve the time complexity in the last 25 years. This can be perhaps explained in the following way. We can rather easily reduce the problem to 3-connected graphs. For planar graphs, the famous result of Whitney tells that us that embeddings of 3-connected graphs in the plane are (combinatorially) unique. But for every non-simply connected surface S, there exist 3-connected graphs with exponentially many embeddings. This makes an essential difference between planar graphs and graphs of higher genus.

There are some other classes of graphs on which the graph isomorphism problem is solvable in polynomial time. This includes general minor-closed families of graphs [45, 53, 54]. A powerful approach based on group theory was introduced by Babai [4]. Based on this approach, Babai et al. [6] proved that isomorphism problem is polynomially solvable for graphs of bounded eigenvalue multiplicity, and Luks [43] described his well-known group theoretic algorithm for isomorphism of graphs of bounded degree. Babai and others [5, 7] investigated the isomorphism problem for random graphs. Chen [16, 17] found a linear time algorithm for graphs of bounded average genus. However, as proved by Chen, these graphs have a very special and restricted structure. Time complexity in these cases usually depends on the maximum degree of graphs and does not apply to the bounded genus case treated in this paper.

2 Polyhedral Maps

A graph G embedded in a surface S has *face-width* or *representativity* at least k, $\mathtt{fw}(G) \geq k$, if every non-contractible closed curve in the surface intersects the graph in at least k points. This notion turns out to be of fundamental importance in the graph minor theory of Robertson and Seymour, cf. [36], and in topological graph theory, cf. [52]. If G is 3-connected and $\mathtt{fw}(G) \geq 3$, then the embedding has properties that are characteristic for 3-connected planar graphs. The main property is that the faces are all simple polygons and that they intersect nicely – if two distinct faces are not disjoint, their intersection is either a single vertex or a single edge. Therefore such embeddings are sometimes called *polyhedral embeddings*.

Whitney proved that any embedding of a graph G in the sphere can be obtained from any other embedding of G into the sphere by performing a sequence of simple local re-embeddings, called *Whitney flippings*. See Section 4 (and Figure 1) for a precise definition; check also [52, Sections 2.6 and 5.2] for more details. Whitney flippings are defined for embeddings in arbitrary surfaces and can be made only when the graph is not 3-connected. We say that two embeddings of the same graph G are *Whitney equivalent* if one embedding can be obtained from the other by a sequence of Whitney flippings.

We say that an embedding of a graph G is (*weakly*) *polyhedral* if $\mathtt{fw}(G) \geq 3$. In the sequel we shall omit the adjective "weakly". Note that embeddings of graphs in the plane are always polyhedral under this definition. As proved by Robertson and Vitray [64], a graph that is polyhedrally embedded in a non-planar surface contains unique non-planar 3-connected component whose induced embedding is in the same surface and whose face-width is the same as the face-width of G. Since this 3-connected component can be discovered in linear time by the algorithm of Hopcroft and Tarjan [31], it may usually be assumed that the graph with a polyhedral embedding is 3-connected.

Thomassen [68] proved that it is NP-complete to decide if a given graph triangulates a surface, and Mohar [50] proved that deciding if a graph admits a polyhedral embedding in some surface is also NP-complete.

Mohar and Robertson [51] proved that for every integer g there is a constant $\xi = \xi(g)$ such that every graph G admits at most $\xi(g)$ polyhedral embeddings, up to Whitney equivalence, in surfaces of genus at most g.

Importance of embeddings of large face-width is highlighted in the book [52]. Let us point out that it is one of the fundamental tools in the seminal Graph Minor Theory by Robertson and Seymour. In fact, they have introduced the representativity (or face-width) in [59], and it is extensively used in the proof of their structure theorem [61] and their proof of Wagner's conjecture [62].

3 Our Main Results

Although existence of polyhedral embeddings is NP-hard [50], we show that for every fixed surface, one can decide this problem in linear time. Moreover, we can find not only one but all such embeddings (up to Whitney equivalence) at the same time.

THEOREM 1. *For each surface S, there is a linear time algorithm for the following problem: Given an integer $k \geq 3$ and a graph G, either find an embedding of G in S with face-width at least k, or conclude that G does not have such an embedding. Moreover, if there is an embedding in S of face-width at least k, the algorithm gives all embeddings with this property, up to Whitney equivalence.*

The importance of Theorem 1 lies in the final conclusion. The reader may wonder why this can be done in linear time, because there could be exponentially many (non-Whitney-equivalent) embeddings on any non-planar surface. But this cannot happen for polyhedral embeddings in a fixed surface, as proved in [51]. The proof in [51] is hard and complicated. Let us observe that our proof is constructive, simpler, and gives a better bound on the number of embeddings. This fact tells us why we can output all polyhedral embeddings in linear time.

If the surface S is not fixed, the problem is NP-hard [50], and examples with exponentially many non-Whitney-equivalent polyhedral embeddings are known. These can be found in [51]. See also [11], where it is shown that the complete graphs K_{36n+7} and K_{36n+19} admit at least 2^{cn^2} distinct polyhedral embeddings, for some constant $c > 0$ and every $n \geq 1$. We have to require the face-width of the embedding to be at least 3 in Theorem 1, since there are 3-connected graphs with exponentially many non-polyhedral embeddings in any surface (other than the sphere). If we want to have unique embedding in the surface of the Euler genus g (which is an analogue of Whitney's theorem on the uniqueness of an embedding in a plane), then the face-width must be at least $\Theta(\log g/\log\log g)$. Sufficiency of this was proved in [47, 66], necessity in [2].

There is a "near-linear" time algorithm to determine the face-width, see [15], and it is believed that the correct order would be $O(n \log n)$. But if we only need to decide whether or not a given graph has face-width at least k, we can do it in linear time. Specifically, Theorem 1 implies the following result.

Corollary 1. *For each surface S, there is a linear time algorithm to decide, for a given integer $k \geq 3$ and a graph G, if G has an embedding on S with face-width at least k.*

Our second main result is about map isomorphism. Let us recall that a *map* is a graph together with a (2-cell) embedding into some surface, and that a *map isomorphism* between two maps is an isomorphism of underlying graphs which preserves the facial walks of the maps.

Theorem 2. *For every surface S (orientable or not), there is a linear time algorithm for to decide whether or not two embedded graphs in S represent isomorphic maps.*

Together with Theorem 1, this implies the following.

Theorem 3. *For every surface S (orientable or not), there is a linear time algorithm for testing graph isomorphism of two graphs, one of which admits a (weakly) polyhedral embedding in S.*

Every planar graph is polyhedrally embeddable into the sphere, so Theorem 3 is an appropriate generalization of the seminal result of Hopcroft and Wong [30]. As remarked above, the time complexity of previously known results for isomorphism of graphs of genus g is $n^{O(g)}$, as proved in the early 1980's. Theorem 3 is the first essential improvement after that, reducing the degree of the polynomial in the time complexity not only to a constant independent of g, but even reducing algorithm complexity to linear time. The only drawback is that it applies only to "polyhedrally embeddable" graphs.

4 Basic Definitions

Before proceeding, we review basic definitions. For basic graph theory notions, we refer the reader to the book by Diestel [20], for topological graph theory we refer to the monograph by Mohar and Thomassen [52]. By an *embedding* of a graph in a surface S we mean a 2-*cell embedding* in S, i.e., we always assume that every face is homeomorphic to an open disk in the plane. Such embeddings can be represented combinatorially by means of *local rotation* and *signature*. See [52] for details. The local rotation and signature determine the *facial walks*, which represent face boundaries. We define the *Euler genus* of a surface S as $2 - \chi(S)$, where $\chi(S)$ is the Euler characteristic of S. This parameter coincides with the usual notion of the genus, except that it is twice as large if the surface is orientable.

Let G be a connected graph that is embedded in a surface S. Suppose that C is a cycle of G that is contractible in S. Let $D \subset S$ be the disk bounded by C. Suppose, moreover, that only (one or) two vertices of C, say v and w, have incident edges that are embedded in $S \setminus D$. Then we define a *flipping* of G (with respect to C) as a re-embedding of G such that the embedding in $S \setminus D$ is unchanged and the embedding of $H := G \cap D$ is changed so that the new embedding of H is equivalent with the original one but the clockwise orientations of all the facial cycles are reversed. Moreover, the outer face boundary of H is the same as the outer face boundary of H in the flipped graph. In other words, we change the embedding of G only inside the disk D bounded by C, where we replace the embedding with its "mirror image". An example of a flipping is shown in Figure 1.

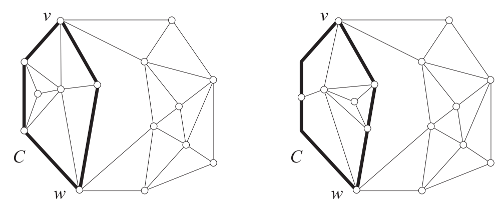

Figure 1: A Whitney flipping

We say that two embeddings of the same graph G are *Whitney equivalent* if one embedding can be obtained from the other by a sequence of Whitney flippings. Note that Whitney flippings and Whitney equivalence do not change the underlying surface. Whitney proved that all embedding of a graph G in the sphere are Whitney equivalent to each other. See [52, Section 2.6] for more details, and see [52, Section 5.2] for treatment on general surfaces.

A graph G embedded in a surface Σ has *face-width* (or *representativity*) at least θ if every closed curve in S, which intersects G in fewer than θ vertices is contractible (null-homotopic) in Σ. Alternatively, the *face-width* of G is equal to the minimum number of facial walks whose union contains a cycle which is non-contractible in Σ. See [52] for further details.

Let W be an embedding of G in a surface S (given by means of a rotation system and a signature). Recall that a surface minor is defined as follows. For each edge e of G, W induces an embedding of both $G - e$ and G/e. The induced embedding of G/e is always in the same surface, but the removal of e may give rise to a face which is not homeomorphic to a disk, in which case the induced embedding of $G - e$ may be in another surface (of smaller genus). A sequence of contractions and deletions of edges results in a W'-embedded minor G' of G, and we say that the W'-embedded minor G' is a *surface minor* of W-embedded graph G.

Let K be a subgraph of G. A K-*bridge* in G (or a *bridge* of K in G) is a subgraph of G which is either an edge $e \in E(G) \setminus E(K)$ with both endpoints in K, or it is a connected component of $G - K$ together with all edges (and their endpoints) between the component and K. The vertices of $B \cap K$ are the *vertices of attachment* of B, A vertex of K of degree different from 2 is called a *branch vertex* of K. A *branch* of K is any path in K (possibly closed) whose endpoints are branch vertices but no internal vertex on this path is a branch vertex of K. Every subpath of a branch e is a *segment* of e. If a K-bridge is attached to a single branch e of K, it is said to be *local*. The number of branch vertices of K is denoted by $\texttt{bsize}(K)$.

A *tree decomposition* of a graph G is a pair (T, Y), where T is a tree and Y is a family $\{Y_t \mid t \in V(T)\}$ of vertex sets $Y_t \subseteq V(G)$, such that the following two properties hold:

(W1) $\bigcup_{t \in V(T)} Y_t = V(G)$, and every edge of G has both ends in some Y_t.

(W2) If $t, t', t'' \in V(T)$ and t' lies on the path in T between t and t'', then $Y_t \cap Y_{t''} \subseteq Y_{t'}$.

The *tree-width* of G is defined as the minimum width taken over all tree decompositions of G.

One of the most important results about graphs, whose tree-width is large, is existence of a large grid minor or, equivalently, a large wall. Let us recall that an r-*wall* is a graph which is isomorphic to a subdivision of the graph W_r with vertex set $V(W_r) = \{(i,j) \mid 1 \leq i \leq r,\ 1 \leq j \leq r\}$ in which two vertices (i,j) and (i',j') are adjacent if and only if one of the following possibilities holds:

(1) $i' = i$ and $j' \in \{j-1, j+1\}$.

(2) $j' = j$ and $i' = i + (-1)^{i+j}$.

We can also define an $(a \times b)$-wall in a natural way, so that the r-wall is the same as the $(r \times r)$-wall. It is easy to see that if G has an $(a \times b)$-wall, then it has an $(\lfloor \frac{1}{2}a \rfloor \times b)$-grid minor, and conversely, if G has an $(a \times b)$-grid minor, then it has an $(a \times b)$-wall. Let us recall that the $(a \times b)$-grid is the Cartesian product of paths $P_a \times P_b$. An (8×5)-wall is shown in Figure 2.

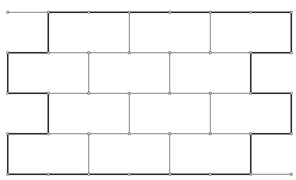

Figure 2: The (8×5)-wall and its outer cycle

The main result of Graph Minors V [58] says that a graph has large tree-width if and only if it contains a large wall as a (topological) minor. See also [21, 55, 63]. For planar graphs, Robertson, Seymour and Thomas [63] proved the following theorem.

THEOREM 4. *For every positive integer r, if a graph G is planar and has tree-width at least $6r$, then G contains an r-wall as a (topological) minor.*

The bound $6r$ in Theorem 4 is best possible.

Let H be an r-wall in G. If G is embedded in a surface S, then we say that the wall H is *flat* if the outer cycle of H bounds a disk in S and the degree-3 vertices of H are contained in this disk. The following theorem was proved by Thomassen [69].

THEOREM 5. *Let S be a surface of Euler genus g. For every r, there is a value $f(g,r)$ satisfying the following. If a graph G embedded in S has tree-width at least $f(g,r)$, then G contains a flat r-wall. Hence, if there is no flat r-wall, then the tree-width of G is at most $f(g,r)$.*

5 Our Algorithms

We now give an overview of our algorithm for Theorem 3. The main new contribution, the core of the algorithm and the hardest part is the proof of Theorem 1. We also apply several existing nontrivial algorithms, which are used to perform intermediate tasks. Our algorithm of Theorem 3 has seven steps that are outlined below. The first six steps are devoted to prove Theorem 1, while the last step is needed for Theorem 3.

The algorithm uses a known result from topological graph theory that there are only finitely many minor-minimal graphs that have embedding of face-width k into S. Then we proceed as follows. First, we embed the graph G into S in an arbitrary way. Having done that, we can reduce the tree-width to a constant by using techniques from graph minors theory. In the resulting graph of bounded tree-width we can find all minors that are minor-minimal graphs for embeddings of face-width k in S. For each of these we look for extensions to the whole graph, and by doing this, we find all embeddings of G of face-width at least k in S.

Henceforth we assume that S is a fixed surface (orientable or not) of Euler genus g, and that G and H are given input graphs. We want to test if G (and this can be done also for H) admits a polyhedral embedding in S. If one exists, we find all of them, up to Whitney equivalence. Finally, we verify if G and H are isomorphic graphs by comparing all produced embeddings.

Step 1. *Find an embedding of G into a surface S' of smallest possible Euler genus $g' \leq g$.*

If such an embedding does not exist, we stop. This task can be achieved by using, for example, the linear time algorithm of Theorem 6 (for each surface of Euler genus at most g), see Section 6.

At this moment, we may assume that we have an embedding of G in S'. We do not require G to have an embedding with face-width at least k. Existence of such embeddings will be addressed later.

Step 2. *Cut the graph on the surface S' into simply connected regions (disks).*

For this task we use some strong results of computational surface topology [22, 39].

Cutting an embedded graph into planar pieces can be done in different ways. One is to break this embedded graph into a bounded number of pieces by using the algorithm from [39]. Another one is to cut the embedded graph into planar pieces, after adding some vertices. This step was previously used in [37]. Details are provided in Section 7.

Now we are given a bounded number of planar graphs. This allows us to find many vertices to be thrown away at once, and we can apply the technique developed in [56, 57] for reducing the tree-width of planar graphs in linear time as explained in the next step.

Step 3. *Bounding the tree-width.*

We remove some "irrelevant" parts of G and get its subgraph G' which has bounded tree-width, and has essentially the same polyhedral embeddings (and essentially the same embeddings of face-width at least k) in S as the graph G. This one and the next part are the heart of our algorithm.

For this task, we adapt the technique from [56, 57], where it is shown that there is a linear time algorithm for the k disjoint paths problem for fixed k when an input graph is planar. This algorithm handles planar graphs more quickly than the classical algorithm of Robertson and Seymour in [60] which solves the same problem for arbitrary graphs in cubic time. The proof in [56, 57] uses several ideas underlying Robertson and Seymour's algorithm.

Let us first sketch the proof of Theorem 1. Recall that an embedding of a given graph is *minimal of face-width k*, if it has face-width k, but for each edge e of G, the face-width of $G - e$ and of G/e are both less than k. It can be shown that a minimal embedding of face-width k cannot contain a flat $4k$-wall, since a vertex in the "middle" of a large flat wall can be deleted, and the resulting subgraph of G would still have face-width at least k.

Consequently, any minimal embedding of face-width k has tree-width at most $f(g, 4k)$ by Theorem 5. Also, by Theorems 5.6.1 and 5.4.1 in [52], any minimal embedding of face-width k has at most $N = N(g, k)$ vertices.

Most importantly, a given graph G has an embedding in the surface S with face-width at least k if and only if G contains one of minimal embeddings of face-width k as a (surface) minor.

Therefore, our task is to find all minimal embeddings of face-width k for the surface S and check for their presence in G. Note that it is possible that several minimal embeddings of face-width k have the same underlying graph with different embeddings. But since each surface minor has at most $N(g, k)$ vertices, the number of all different embeddings of these graphs with face-width k is also bounded by a number $N'(g, k)$ depending on g and k only.

Define a vertex of G to be an *irrelevant vertex* if for every embedding of G of face-width at least k, the induced embedding of $G-v$ also has face-width at least k. Therefore, if a given graph has large tree-width, we can find an irrelevant vertex in a flat grid minor. We delete irrelevant vertices as long as to obtain a subgraph G' of G of bounded tree-width. Let us observe that G' contains as a minor some fixed minimal embedding of face-width k if and only if the original input graph G does. Since G' has bounded tree-width, we can find all surface minors of minimal embeddings of face-width k contained in G' in linear time by the standard dynamic programming approach.

In order to get a linear time algorithm, we have to find and remove many irrelevant vertices at once. Therefore, we need to modify the reduction step which results in a bounded tree-width graph. Roughly speaking, we need to find many irrelevant vertices to be thrown away at the same time. Such an idea was demonstrated in [56, 57] when an input graph is planar. We upgrade on this idea to work in our case. More details are provided in Section 8.

Step 4. *Finding minors in graphs of bounded tree-width.*
At this step, we need to detect, not only one, but all surface minors of minor-minimal embedding of face-width k. This is because we want to find all embeddings of face-width at least k. Note that the number of these surface minors is at most $N'' = N''(g, k)$, where N'' is an integer depending only on g, k. See [52, Theorem 5.6.1]. In particular, each of these maps has bounded order. At this moment, the current graph G' has bounded tree-width by Step 3. Therefore, we can use the standard dynamic programming approach to find all surface minors in G' that are minor minimal embeddings of face-width k. If G' has none of these surface minors, we conclude that G is not embeddable in S with face-width at least k. Hereafter, we assume that we have found all such surface minors.

Alternatively, this can be done by the recent result of Adler, Grohe and Kreutzer [1].

Step 5. *Expanding each excluded minor.*
We expand each vertex of every surface minor so that each vertex becomes a subgraph of the given graph G. This is actually easy, and the size of this subgraph is still bounded in terms of Euler genus g and the face-width k. We also need to eliminate all local bridges for each of the subgraphs, i.e., bridges attached to only one subdivided edge of the abstract graph.

Step 6. *Finding all polyhedral embeddings of each subgraph from Step 5, and extending the embedding to the whole graph.*
Let us first observe that we may assume that our input graph is 3-connected. To see this, we first perform the algorithm by Hopcroft and Tarjan [31] to make the input graph 3-connected. Each 2-connected component has to be planar, since otherwise, the input graph cannot have an embedding in the surface with face-width at least 3 (the 2-separation gives rise to a non-contractible curve of size 2), see [52]. Hence we can replace this 2-connected component by an edge. Any embedding of the current graph can be extended to each of 2-connected components, because they are all planar. Therefore, we can assume that a current graph is now 3-connected.

Step 6 is actually easy, since the size of this subgraph is bounded. So we can use the dynamic programming approach, and we can do it in linear time. Find the embedding extension of each surface minor to the whole graph, if one exists. At the moment, all the bridges of this subgraphs are in the face of the embedding, if one exists. In this case, we just embed all the bridges in the disk bounded by the face of the embedding of the subgraph. This can be done in linear time by the result of Juvan and Mohar [35], if the input graph is 3-connected.

Step 7. *Isomorphism of embedded graphs.*
We start by describing an easy $O(n^2)$ algorithm based on the algorithm of Hopcroft and Tarjan [32], and Weinberg [71]. Then we expose a rather straightforward $O(n \log n)$ algorithm, which also modifies the algorithm by Hopcroft and Tarjan [33]. Finally, we give a linear time algorithm, which generalizes the result by Hopcroft and Wong [30] and uses ideas similar to those in [30].

We shall look at each step in the next sections, except for Step 3, which was given in [37]. Let us observe that our algorithm of Theorem 1 gives rise to all drawings in the surface S with face-width at least k, up to Whitney equivalence.

6 Embedding Graphs into a Fixed Surface

Our algorithms make use of planarity and need testing for embeddability of graphs on a fixed surface. A seminal result of Hopcroft and Tarjan [34] from 1974 gives a linear time algorithm for testing planarity of graphs. Going from the plane to general surfaces, embedding problems become notoriously hard. Thomassen [67] proved that computing the genus of graphs is NP-hard. On the other hand, if the genus is bounded, one can say more. Filotti, Miller, and Reif [23] were the first to give an $O(n^{O(g)})$ polynomial time algorithm for testing embeddability of graphs into an orientable surface of genus g. Djidjev and Reif [19] improved the algorithm of [23] by presenting a polynomial time algorithm for each fixed orientable surface, where the degree of the polynomial is fixed.

Robertson and Seymour [60] proved that every class of graphs that is closed under taking minors is recognizable in cubic time. Their results give rise to an $O(n^3)$ algorithm for deciding whether or not G can be embedded into the surface of the Euler genus g, for any fixed g, but it does not give an embedding, if one exists. In 1996, Mohar [48, 49] gave a linear time algorithm for testing embeddability of graphs in surfaces and constructing an embedding, if one exists.

THEOREM 6 (MOHAR [48, 49]). *For every fixed surface S, there is a linear time algorithm which either finds an embedding of a given graph G into S or returns a minimal forbidden minor for S contained in G.*

This is one of the hardest results in this area. It clearly generalizes linear time algorithms for testing planarity and constructing a planar embedding if one exists [12, 18, 34, 72]. A new, simpler linear time algorithm was found recently by Kawarabayashi, Mohar, and Reed [38].

7 Cutting Embedded Graph into Planar Pieces

Let G be a graph that is embedded into the surface S of Euler genus at most g.

The purpose of this section is to explain how to cut the embedded graph into a bounded number of planar pieces (disks). As far as we see, there are two methods. One is to get at most $O(g^2)$ planar subgraphs in G such that intersection of any two planar subgraphs are on the boundary. The other is to cut the embedded graph into a planar one after adding some vertices as described in [37].

We shall get $O(g^2)$ planar disks in G such that intersection of any two of them is part of the boundary. This can be done if we can detect a shortest non-contractible curve in linear time, since we know that the Euler genus is at most g, so we could repeatedly apply the algorithm until, after cutting along these curves, the resulting graph is planar. But unfortunately, there is only a "near" linear time algorithm for this problem [15], and it is believed that the correct order would be $O(n \log n)$, see [15]. So we cannot apply this method in our algorithm. Instead, we will adapt the method of detecting so-called "canonical polygonal schema", which we shall define here.

Suppose that G is embedded into the surface S. For simplicity we assume here that S is orientable and that g is the genus of S. The non-orientable case is similar. We first define a *cut graph* C of G. This is a subgraph of G such that after slicing at C (i.e., cutting along edges of C), the resulting graph G' is embedded into a disk. This disk is sometimes called a "polygonal scheme" of G. Each edge of C appears twice on the boundary of polygonal schema of G', and we can obtain G by gluing together these corresponding boundary edges. Let us look at C more closely. We would like to get such a set C so that C consists of $2g$ non-contractible curves $a_1, \dots, a_g, b_1, \dots, b_g$ in such a way that each of these curves is a cycle. (Here, a cycle means an alternating sequence of edges and vertices, where edges can connect two successive vertices that lie in the same face, either in its interior or on the interior of one of its boundary edges. So if a graph embedded into this surface is a triangulation, then this cycle must be a cycle contained in G.) After slicing each a_i and b_i, we would get $4g$ curves $a_1, \overline{a_1}, b_1, \overline{b_1}, \dots, a_g, \overline{a_g}, b_g, \overline{b_g}$ in such a way that each a_i and b_i are directed counterclockwise, and each $\overline{a_i}$ and $\overline{b_i}$ are directed clockwise. If we identify curves a_i and $\overline{a_i}$, and b_i and $\overline{b_i}$ for $i = 1, \dots, g$, then we would get an embedding of G into the orientable surface of Euler genus g. Similarly, we can do it for the non-orientable case. We call these cycles $a_1, \dots, a_g, b_1, \dots, b_g$ *canonical polygonal scheme*. It is easy to see that these $2g$ curves consist of generators of the fundamental group of the surface of the Euler genus g.

The main result in [39] is the following. See also [22].

THEOREM 7. *For any graph G on the surface of Euler genus g, there is an $O(gn)$-time algorithm to detect a canonical polygonal scheme. Actually, the algorithm detects non-contractible curves $a_1, \dots, a_g, b_1, \dots, b_g$ such that each of these curves is a cycle (for the definition of the cycle, see above), and after slicing each a_i and b_i, we would get $4g$ curves $a_1, \overline{a_1}, b_1, \overline{b_1}, \dots, a_g, \overline{a_g}, b_g, \overline{b_g}$ in such a way that each a_i and b_i are directed counterclockwise, and each $\overline{a_i}$ and $\overline{b_i}$ are directed clockwise. Furthermore, if we identify curves a_i and $\overline{a_i}$, and b_i and $\overline{b_i}$ for $i = 1, \dots, g$, then we would get an embedding of G into the surface of the Euler genus g.*

We can modify these non-contractible curves. Since they generate the fundamental group of the surface, we can take these non-contractible curves so that the intersection of any two curves C_1 and C_2 is a path. (Here, again, a path means an alternating sequence of edges and vertices, where edges can connect two successive vertices that lie in the same face, either in its interior or on the interior of one of its boundary edges. So if a graph embedded into this surface is a triangulation, then this path is contained in the graph.) This argument also follows from the algorithm of [39], where the authors first construct a breadth-first search spanning tree T in a triangulation of a given graph on the surface, and contract T into a single point. Then they find $2g$ loops which consist of generators of the fundamental group of the surface. The corresponding cycles in T are determined by these loops, and the intersection of any two curves C_1 and C_2 is a path. Our graph G may not be a triangulation, but we can extend it to a triangulation by triangulating each face, and then apply this argument to the resulting graph. This idea is also demonstrated in [14, 70].

Now these non-contractible curves divide the graph G into $r \leq 4g^2$ planar disks $P_1, \dots, P_r$ such that the intersection of any two of them is part of the boundary.

Alternatively, we can adapt the method of Reed, Robertson, Schrijver and Seymour [56, 57]. Their method shows how to bound the tree-width of graphs on a fixed surface in linear time. We have even simpler way of doing this.

We want to find a short non-contractible simple closed curve γ in S that intersects G in vertices only. Then we cut the surface along γ and henceforth reduce the genus. By doing this g times, we end up with a planar embedding. The argument in [22] implies that there is a linear time 2-approximation algorithm for computing the face-width of embedded graphs. It was shown in [25] that the face-width is at most $O(\sqrt{gn})$. So, if we perform this procedure $O(g)$ times, then we get a plane embedding, after duplicating vertices of the embedded graph, where the cutting occurred. The number of duplicated vertices is $O(g\sqrt{gn})$. Hence we get a desired planar graph from the embedded graph in linear time.

8 Bounding the tree-width

Let $P_1, \dots, P_r$, $r \leq 4g^2$, be planar disks obtained as explained in the previous section.

What we shall do now is that, for each P_i, we are going to bound the tree-width by deleting many vertices, and we will do this in linear time. We will need the following result. For the proof see [52].

THEOREM 8. *Suppose that G 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 are contained in Q. If Q contains a $4k$-wall W, then every vertex of W, which has face-distance in the wall W at least k from all the vertices of the outer cycle of W, is irrelevant.*

Theorem 8 says that for any planar subgraph, if there is a $4k$-wall, then we can delete the middle vertex, and we can keep deleting irrelevant vertices until there is no flat $4k$-wall in the resulting graph. 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 used by Reed, Robertson, Seymour and Schrijver [56, 57], who proved that there is a linear time algorithm for the k disjoint paths problem for planar graphs. So we shall use this method to delete vertices of each planar graph P_i until the resulting graph has no flat $4k$-wall. Let us state this as a lemma.

LEMMA 1. *Suppose that Q is a planar subgraph of G such that for every vertex of Q that is not on the outer cycle of Q, all its neighbors in the graph G are contained in Q. There is a linear time algorithm to find a vertex set $X \subseteq V(Q)$ such that deleting the vertices of X does not change the problem of finding minimal embeddings of face-width at least k as a minor, by a sequence of applications of Theorem 8. Furthermore, this algorithm can output the graph $Q - X$ such that it does not contain a $4k$-wall.*

The proof of Lemma 1 is exactly the same as what Reed, Robertson, Schrijver and Seymour did. We omit the details, and refer to [56, 57]. In the full version of this paper, these details will be included.

After performing Lemma 1 for each planar graph, the tree-width of each of these planar subgraphs is at most $24k$ by Theorem 4.

One can show (and the proof will be outlined in the full paper) that this implies (using Theorem 5) that the face-width of the subgraph of G obtained by pasting these planar pieces each together, is at most $f(g, 48k)$.

9 Finding minors in graphs of bounded tree-width

From the previous section, the input graph is reduced to a subgraph of bounded tree-width. Now we need to find some excluded surface minors in this bounded tree-width graph. Theorems 5.6.1 and 5.4.1 in [52] show the following:

THEOREM 9. *A minimal embedding of face-width k in the surface S does not contain a flat $4k$-wall, and consequently has tree-width at most $f(g, 4k)$. Moreover, it has at most $N = N(k, g)$ vertices, where the integer N depends on g and k only. Therefore, there are at most $N'' = N''(g, k)$ minor-minimal embeddings of face-width k.*

Our task is to detect the presence of all surface minors from Theorem 9. This is because, in order to prove Theorem 1, and apply it to prove Theorem 3, we need to find all polyhedral embeddings in S.

By Theorem 9 and the fact that our current graph has bounded tree-width, we can find all the surface minors of minimal embeddings of face-width k in linear time, by the standard method dynamic programming approach, see [10] (as we can find each of the surface minors in linear time). If G has none of these minors, we conclude that G is not embeddable in S with face-width at least k. Hereafter, we assume that we have found all the surface minors. Alternatively, a recent result of Adler, Grohe and Kreutzer [1] gives all required surface minors.

Therefore, at this stage, we can detect all surface minors of minimal embeddings of face-width k in linear time.

10 Expanding each excluded minor to excluded subgraphs

Hereafter, we consider the original input graph G, which may have large tree-width.

Furthermore, we are given the family of surface minors $\mathcal{F} = \{F_1, \dots, F_l\}$ of G such that the following holds:

1. For all i, F_i is a minor-minimal embedding of face-width k, and F_i is a surface minor for an embedding of G in S.
2. $l \leq N''(g, k)$ and $|F_i| \leq N(g, k)$ for $i = 1, \dots, l$.
3. For every embedding of G in S of face-width $\geq k$, there exists an i such that F_i is a surface minor of this embedding of G.

In order to get all embeddings of G in S whose face-width is at least k, we need to figure out how the embedding of its surface minor F_i can be extended to G. But there is one problem here.

Suppose we find an embedding of F_i of face-width k. Then each face is homeomorphic to a disk. Ideally, we would like to prove that the rest of the graph (each F_i-bridge) will lie in a unique face of F_i. Since each vertex in F_i can be obtained by the minor operation, it is not easy to figure out how the rest of the graph can be attached to F_i. It would be much easier if F_i is a subgraph. So we need to expand each vertex of F_i in order to get possible subgraphs F_i' of G by reversing the minor operation. This is actually easy. Note that each obtained subgraph F_i' may have many vertices of degree 2, but $\texttt{bsize}(F_i')$ is still bounded, since the expansion from F_i only involves the vertices of F_i. So $\texttt{bsize}(F_i')$ is at most $|E(F_i)|/2 \times |F_i|$, since each vertex of degree d may be expanded to at most $d-2$ vertices of degree at least three.

Since $|\mathcal{F}| = l \leq N''(g, k)$, we can do the expansion of all the surface minors in linear time.

11 Finding all polyhedral embeddings of subgraphs

Let G be a given input graph. Furthermore, we are given a family of embedded subgraphs $\mathcal{F}' = \{F_1', \dots, F_l'\}$ of G such that the following holds:

1. For all i, F_i' is embedded with face-width at least k in S.
2. $l \leq N''(g, k)$ and $\texttt{bsize}(F_i') \leq N'(g, k)$ for all i.

In order to get all embeddings of G of face-width at least k in S, we first determine all embeddings in S of face-width at least k for each F_i'. Since the branch size of F_i' is bounded, we can do this in constant time.

Let us observe that we may assume that our input graph is 3-connected. To see this, we first perform the algorithm by Hopcroft and Tarjan [31] to make the input graph 3-connected. If $G = G_1 \cup G_2$, where $|G_1 \cup G_2| = 2$, we let G'_i be the graph obtained from G_i by adding the edge connecting the verices of $G_1 \cup G_2$. Now, precisely one of G'_i $(i = 1, 2)$ has to be planar. Otherwise, G cannot have an embedding in a non-planar surface with face-width at least 3 ([52]). Hence we can replace this planar 2-connected component, say G_2 by an edge, i.e. we replace G by G'_1. Any embedding of G'_1 of face-width $k \geq 3$ can be extended (uniquely up to Whitney equivalence) to an embedding of G of the same face-width by using a plane embedding of G'_2.

Therefore, we can assume that our current graph is now 3-connected.

Suppose there are no local F'_i-bridges for the subgraph F'_i. Once we fix an embedding of F'_i, we can figure out whether or not this embedding extends to the whole graph G. Since F'_i has a polyhedral embedding and F'_i is a subgraph of G, so if each F'_i-bridge lies in the face of the embedding of F'_i (the face is uniquely determined, since each F'_i-bridge is not local), then we can extend the embedding of F'_i to the whole graph.

The embedding extension can be done in linear time by using known planarity testing algorithms [12, 18, 34, 72].

Theorem 10. *Suppose C is a cycle of a given graph G. Then there is a linear algorithm to decide whether or not G can be embedded into a plane with the outer face boundary C. Moreover, if it can, the algorithm gives rise to a desired embedding. The embedding is unique, up to Whitney equivalence.*

For the proof, we refer the reader to [52].

As pointed out above, in order to apply Theorem 10, we need to eliminate all local bridges. This can be done by the algorithm in [35]. It can be done in linear time. So, we now get the following.

Theorem 11. *In linear time, we can modify the subgraphs $F'_1, \dots, F'_l$ of G so that no F'_i-bridge is local for $i = 1, \dots, l$.*

In conclusion, if we fix one of the embeddings of F'_i and no F'_i-bridge is local, either the rest of bridges can be embedded into faces bounded by the embedding, or else, there are no embedding extensions. By Theorem 10, if the first case occurs, we can embed the rest of the bridges in linear time. Since $\texttt{bsize}(F'_l) \leq N'(g, k))$ and $l \leq N''(g, k)$, we can get all embeddings of face-width at least k, up to Whitney equivalence, in linear time.

It is worth mentioning that our proof also yields the result of Mohar and Robertson [51], but our proof is constructive, simpler and gives a better bound on $f(g)$.

This completes the description of our algorithm for Theorem 1.

12 Map isomorphism in linear time

It remains to prove Theorem 3. By Theorem 1, for both input graphs G and H, we have all polyhedral embeddings, up to Whitney equivalence. The number of embeddings is at most $f(g)$ for some function f of Euler genus g. Our idea is to compare each of all the embeddings of G to each of all the embeddings of H. If we can figure out each of them in linear time, we would get a linear time algorithm for the graph isomorphism problem of polyhedral embeddable graphs, since there are at most $f(g)$ embeddings of G and H, respectively.

It remains to figure out the graph isomorphism of two embedded graphs, in terms of embeddings, i.e., whether or not two embeddings are same. As pointed out by Weinberg [71], there is an easy algorithm for this problem when the surface is planar. In fact, one can mimic this algorithm for an arbitrary surface, so we can easily get an $O(n^2)$ time algorithm for this problem.

Hopcroft and Tarjan [33] gave an $O(n \log n)$ time algorithm for this problem when the surface is planar. The idea is to use the famous planar separator theorem (a *separator* is a vertex set X of order at most $O(\sqrt{n})$ such that $G - X$ can be partitioned into two vertex sets A, B in such a way that there are no edges between A and B, and $|A|, |B| \leq 2n/3$) by Lipton and Tarjan [41]. The applications of this separator theorem were demonstrated in [42]. Specifically, the separator theorem was applied $O(\log n)$ times to obtain $O(\log n)$ subgraphs of constant size. Then Hopcroft and Tarjan [33] compare two graphs by placing vertices to these small components which can be done in linear time. As a planar separator can be found in linear time, Hopcroft and Tarjan [33] can get an $O(n \log n)$ time algorithm for this problem.

The same method easily works for bounded genus graphs. There is a linear time algorithm to find a separator in bounded genus graphs by Gilbert, Hutchinson and Tarjan [25]. We can use this theorem to follow the Hopcroft and Tarjan's approach [33] to obtain an $O(n \log n)$ algorithm for this problem. Therefore, using Theorem 1, there is an $O(n \log n)$ algorithm for the graph isomorphism problem of polyhedrally embeddable graphs.

At the moment, we are able to mimic the proof of Hopcroft and Wong [30] to obtain a linear time algorithm. Specifically, we can prove the following.

Theorem 12. *Let S be a fixed surface. There is a linear time algorithm to decide if, for two graphs G, H embedded in S, the embeddings of G and H are combinatorially the same.*

Sketch of the proof. The basic idea of the algorithm is the same as that in Hopcroft and Wong [30]. Roughly speaking, they assign labels to each vertex and each edge. Then they made a reduction. This reduction takes place when (i) there is a vertex of degree at most 2 or (ii) there is a face F of size d such that some face adjacent to F has size other than d or (iii) there is a vertex v of degree d such that some vertex adjacent to v has degree other than d. These reductions are performed in order determined by their priority, and labels of vertices and edges are changed accordingly. The priority ordering insures a canonical form for the graph at each stage. This allows to prove that the resulting graphs are isomorphic if an only if the original graphs are isomorphic. When no further reduction is possible, the graphs lie in five families of graphs, which are easy to recognize.

We can do the same reduction process for maps on the surface S. After performing all reductions, we are left either with very small graph or with a regular map on S. With the exception of the torus and the Klein bottle, which admit arbitrarily large regular maps, all other surfaces only have finitely many regular maps. Therefore, we can check their isomorphism (even with labels on vertices and edges) in time proportional to the time needed to compare the labels. This completes the proof if S is not the torus or the Klein bottle.

Finally, for the two exceptional surfaces, all regular maps are classified. They fall into three categories: (3,6)-regular maps (honeycomb lattices), (6,3)-regular (their duals), and (4,4)-regular ones. This case needs a special touch but the map isomorphism can nevertheless be detected in linear time. □

By Theorems 1 and 12, we can prove Theorem 3, since there are at most $f(g)$ polyhedral embeddings in the fixed surface.

13. REFERENCES

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

[2] D. Archdeacon, Densely embedded graphs, *J. Combin. Theory Ser. B* **54** (1992), 13–36.

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

[4] L. Babai, Monte Carlo algorithms in graph isomorphism testing, Univ. Montreal Tech. Rep. DMS 79-10, 1979. http://people.cs.uchicago.edu/ laci/lasvegas79.pdf

[5] L. Babai, P. Erdős, and S. Selkow, Random graph i-isomorphism, *SIAM J. Computing* **9**, 628–635, (1980).

[6] L. Babai, D.Y. Grigoryev and D. Mount, Isomorphism of graphs with bounded eigenvalue multiplicity, In *Proc. 14th ACM Symposium on Theory of Computing (STOC'82)*, 310–324, (1982).

[7] L. Babai and L. Kucera, Canonical labelling of graphs in linear average time, In *Proc. 20th Annual IEEE Symposium on Foundation of Computing (FOCS'79)*, 39–46, (1979).

[8] L. Babai, S. Moran, Arthur - Merlin games: a randomized proof system and a hierarchy of complexity classes, *J. Comp. Syst. Sci.* **36** (1988), 254–276.

[9] H. L. Bodlaender, Polynomial time algorithm for graph isomorphism and chromatic index on partial k-trees, *J. Algorithm*, **11**, 631–643, (1990).

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

[11] P. Bonnington, M. Grannell, T. Griggs, and J. Širáň, Exponential families of non-isomorphic triangulations of complete graphs, *J. Combin. Theory Ser. B* **78** (2000), 169–184.

[12] 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.

[13] R. Boppana, J. Hastad and S. Zachos, Does co-NP have short interactive proofs? *Inform. Proc. Lett.* **25** (1987), 127–132.

[14] T. Brahana, Systems of circuits on 2-dimensional manifolds, *Ann. Math.* **23** (1921), 144–168.

[15] S. Cabello and B. Mohar, Finding shortest non-separating and non-contractible cycles for topologically embedded graphs, *Discrete Comput. Geom.* **37** (2007), 213–235.

[16] J. Chen, A linear-time algorithm for isomorphism of graphs of bounded average genus, *SIAM J. Discrete Math.* **7** (1994), 614–631.

[17] J. Chen, Algorithmic graph embeddings, *Theoret. Comput. Sci.* **181** (1997), 247–266.

[18] 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.

[19] H. Djidjev and J. H. Reif, An efficient algorithm for the genus problem with explicit construction of forbidden subgraphs, In *Proc. 23rd ACM Symposium on Theory of Computing (STOC'91)*, pp. 337–347, (1991).

[20] R. Diestel, Graph Theory, Graduate Text 3rd Edition, Springer, 2005.

[21] 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.

[22] J. Erickson and S. Har-Peled, Optimal cutting a surface into a disk, *Discrete Comput. Geom.* **31** (2004), 37–59. Conference version in *Proc. 18th Annual ACM Symposium on Computational Geometry* (2002), 244–253.

[23] L. S. Filotti, G. L. Miller, and J. Reif, On determining the genus of a graph in $O(n^{O(g)})$ steps, in *Proc. 11th ACM Symposium on Theory of Computing (STOC'79)*, pp. 27–37, (1979).

[24] L.S. Filotti and J. N. Mayer, A polynomial time algorithm for determining the isomorphism of graphs of fixed genus, In *Proc. 12th ACM Symposium on Theory of Computing (STOC'80)*, pp. 236–243, (1980).

[25] J. R. Gilbert, J.P. Hutchinson and R. E. Tarjan, A separator theorem for graphs of bounded genus, *J. Algorithms* **5** (1984), 391–407.

[26] O. Goldreich, S. Micali, A. Wigderson, Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems, *J. ACM* **38** (1991), 691–729.

[27] S. Goldwasser, M. Sipser, Private coins versus public coins in interactive proof systems, *Proc. 18th ACM Symposium on Theory of Computing (STOC'86)* (1986), 59–68.

[28] M. Grohe, Isomorphism testing for embeddable graphs through definability, In *Proc. 32th ACM Symposium on Theory of Computing (STOC'00)*, 2000, 63–72.

[29] M. Grohe and O. Verbitsky, Testing graph isomorphism in parallel by playing a game, In *33rd International Colloquium on Automata, Languages and Programming (ICALP)*, 2006, 3–14.

[30] J. E. Hopcroft and J. Wong, Linear time algorithm for isomorphism of planar graphs, In *Proc. 6th ACM Symposium on Theory of Computing (STOC'74)*, 27–37, (1974).

[31] J. E. Hopcroft and R. Tarjan, Dividing a graph into triconnected components, *Siam J. Comput.* **2** (1973), 135–158.

[32] J. E. Hopcroft and R. Tarjan, Isomorphism of planar graphs (working paper), In R. E. Miller and J. W. Thathcer, editors, *Complexity of Computer Computations.* Plenum Press, 1972.

[33] J. E. Hopcroft and R. Tarjan, A $v \log v$ algorithm for isomorphism of triconnected planar graphs, *J. Comp. Syst. Sci.* **7** (1973), 323–331.

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

[35] M. Juvan, J. Marinček, and B. Mohar, Elimination of local bridges, *Math. Slovaca* **47** (1997), 85–92.

[36] K. Kawarabayashi, B. Mohar, Some recent progress and applications in graph minor theory, *Graphs Combin.* **23** (2007), 1–46.

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

[38] K. Kawarabayashi, B. Mohar, and B. Reed, A simpler linear time algorithm for embedding graphs into an arbitrary surface, *in preparation.*

[39] F. Lazarus, M. Pocchiola, G. Vegter and A. Verroust, Computing a canonical polygonal schema of an orientable triangulated surface, in *Proc. 17th Annual ACM Symposium on Computational Geometry* (2001), 80–89.

[40] D. Lichtenstein, Isomorphism of graphs embeddable on the projective plane, In *Proc. 12th ACM Symposium on Theory of Computing (STOC'80)* (1980), 218–224.

[41] R.J. Lipton and R. E. Tarjan, A separator theorem for planar graphs, *Siam J. Appl. Math.* **36** (1979), 177–189.

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

[43] E. Luks, Isomorphism of graphs of bounded valance can be tested in polynomial time, *J. Computer and System Sciences* **25**, 42–65, (1982).

[44] G. Miller, Isomorphism of graphs which are pairwise k-separable, *Information and Control* **56**, 21–33, (1983).

[45] G. Miller, Isomorphism of k-contractible graphs. A generalization of bounded valence and bounded genus graphs, *Information and Control* **56**, 1–20, (1983).

[46] G. Miller, Isomorphism testing for graphs of bounded genus, In *Proc. 12th ACM Symposium on Theory of Computing (STOC'80)*, 225–235, (1980).

[47] B. Mohar, Uniqueness and minimality of large face-width embeddings of graphs, *Combinatorica* **15** (1995), 541–556.

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

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

[50] B. Mohar, Existence of polyhedral embeddings of graphs, *Combinatorica* **21** (2001), 395–401.

[51] B. Mohar and N. Robertson, Flexibility of polyhedral embeddings of graphs in surfaces, *J. Combin. Theory Ser. B* 83 (2001), 38–57.

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

[53] I. N. Ponomarenko, The isomorphism problem for classes of graphs that are invariant with respect to contraction, *Zap. Nauchen. Sem. Leningrad. Osdel. Mat. Inst. Steklov.* (LOMI) **174**, 147–177, (1988), in Russian.

[54] I. N. Ponomarenko, The isomorphism problem for classes of graphs, *Dokl. Akad. Nauk SSSR* **304** (1989) 552–556; Engl. transl. in *Soviet Math. Dokl.* **39** (1989) 119–122.

[55] B. Reed, Tree width and tangles: a new connectivity measure and some applications, in *"Surveys in Combinatorics, 1997 (London)"*, Cambridge Univ. Press, Cambridge, 1997, pp. 87–162.

[56] B. Reed, N. Robertson, A. Schrijver and P. D. Seymour, Finding disjoint trees in planar graphs in linear time, in *Graph Structure Theory (Seattle, WA, 1991)*, 295–301, Amer. Math. Soc., 1993.

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

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

[59] N. Robertson and P.D. Seymour, Graph minors VII. Disjoint paths on a surface, *J. Combin. Theory Ser. B* **45** (1988) 212–254.

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

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

[62] N. Robertson and P. D. Seymour, Graph minors. XX. Wagner's Conjecture, *J. Combin. Theory Ser. B* **92** (2004), 325–357.

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

[64] N. Robertson, R. P. Vitray, Representativity of surface embeddings, in: "Paths, Flows, and VLSI-Layout," B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver Eds., Springer-Verlag, Berlin, 1990, pp. 293–328.

[65] U. Schöning, Graph isomorphism is in the low hierarchy, *J. Comp. Syst. Sci.* **37** (1988), 312–323.

[66] P. D. Seymour and R. Thomas, Uniqueness of highly representative surface embeddings, *J. Graph Theory* **23** (1996), 337–349.

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

[68] C. Thomassen, Triangulating a surface with a prescribed graph, *J. Combin. Theory Ser. B* **57** (1993), 196–206.

[69] C. Thomassen, A simpler proof of the excluded minor theorem for higher surfaces. *J. Combin. Theory Ser. B* **70** (1997), 306–311.

[70] G. Vegter and C. K. Yap, Computational complexity of combinatorial surfaces, in *Proc. 6th Annual ACM Symposium on Computational Geometry* (1990), 102–111.

[71] H. Weinberg, A simple and efficient algorithm for determining isomorphism of planar triply connected graphs, *Circuit Theory* **13** (1966), 142-148.

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