University of Ljubljana
Institute of Mathematics, Physics and Mechanics
Department of Mathematics
Jadranska 19, 1000 Ljubljana, Slovenia

Preprint series, Vol. 47 (2009), 1078

A SIMPLER LINEAR TIME ALGORITHM FOR EMBEDDING GRAPHS INTO AN ARBITRARY SURFACE AND THE GENUS OF GRAPHS OF BOUNDED TREE-WIDTH

Ken-ichi Kawarabayashi   Bojan Mohar
Bruce Reed Ken-ichi Kawarabayashi

ISSN 1318-4865

Ljubljana, February 2, 2009

A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width

Ken-ichi Kawarabayashi*
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo 101-8430, Japan

Bojan Mohar†
CRC in Graph Theory
Department of Mathematics
Simon Fraser University, Burnaby, B.C.

Bruce Reed‡
Canada Research Chair in Graph Theory
McGill University, Montreal, Canada

Abstract

For every fixed surface S, orientable or non-orientable, and a given graph G, Mohar (STOC'96 and Siam J. Discrete Math. (1999)) described a linear time algorithm which yields either an embedding of G in S or a minor of G which is not embeddable in S and is minimal with this property. That algorithm, however, needs a lot of lemmas which spanned six additional papers. In this paper, we give a new linear time algorithm for the same problem. The advantages of our algorithm are the following:

1. The proof is considerably simpler: it needs only about 10 pages, and some results (with rather accessible proofs) from graph minors theory, while Mohar's original algorithm and its proof occupy more than 100 pages in total.
2. The hidden constant (depending on the genus g of the surface S) is much smaller. It is single exponential in g, while it is doubly exponential in Mohar's algorithm.

As a spinoff of our main result, we give another linear time algorithm, which is of independent interest. This algorithm computes the genus and constructs minimum genus embeddings of graphs of bounded tree-width. This resolves a conjecture by Neil Robertson and solves one of the most annoying long standing open question about complexity of algorithms on graphs of bounded tree-width.

10 April, 2008, revised April 11, 2008.

Keywords: Linear time algorithm, Embedding, Surface, Genus of a graph, Tree-width.

*Research partly supported by Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research, by C & C Foundation, by Kayamori Foundation and by Inoue Research Award for Young Scientists. Visiting Department of Mathematics, Simon Fraser University, Burnaby, B.C. V5A 1S6. Email address: `k_keniti@nii.ac.jp`

†Supported in part by 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, Ljubljana, Slovenia. Email address: `mohar@sfu.ca`

‡On leave from Project Mascotte INRIA, Laboratoire I3S, CNRS, Sophia-Antipolis, France. Email address: `breed@cs.mcgill.ca`

1 Introduction

1.1 Planarity Testing and Surface Embeddings

A seminal result of Hopcroft and Tarjan [11] 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 importance of the genus parameter of graphs and from the fact that graphs of bounded genus naturally generalize the family of planar graphs and share many important properties with them.

Let us first observe that deciding the genus of a given graph is NP-complete as Thomassen showed in [39]. Our main interest here is whether or not a given graph can be embedded into the surface of Euler genus g (orientable or non-orientable), where g is a fixed integer. Filotti, Miller and Reif [9] were the first to give a polynomial time algorithm for this problem. In their solution, the degree of the polynomial bound on the time complexity depends on g. Djidjev and Reif [8] improved the algorithm of [9] by presenting a polynomial time algorithm for each fixed orientable surface, where the degree of the polynomial is fixed. In addition, linear time algorithms have been devised for embedding graphs in the projective plane [19] and in the torus [14].

Robertson and Seymour [35] have already proved that all classes of graphs that are closed under taking minors are recognizable in cubic time. This implies that there is an $O(n^3)$ algorithm for deciding whether G can be embedded into a surface of Euler genus g, for every fixed g. In 1996, Mohar [21, 23] gave a linear time algorithm for testing embeddability of graphs in surfaces and constructing an embedding, if one exists. This is one of the deepest results in this area, since it clearly generalizes linear time algorithms for testing planarity and constructing a planar embedding if one exists [4, 6, 11, 41].

Recently, some apparently new and nontrivial linear time algorithms concerning graph embeddings appear: One [16] is concerning drawing a given graph into the plane with at most k crossings (for fixed k), and the other one [17] is concerning embedding a given graph into a surface with face-width at least k with an application to the graph isomorphism problem. Both algorithms depend on the linear time algorithm for embedding a given graph into a given surface [21, 23].

1.2 Our main result

While Mohar's algorithm [21, 23] is an ultimate result in this area, it needs a lot of preparation which spanned six other papers [12, 13, 15, 20, 22, 26]. Our purpose of this paper is to give a new, simpler linear time algorithm for testing embeddability of graphs in surfaces and constructing an embedding, if one exists.

Specifically, we provide the following theorem.

Theorem 1.1 *For every fixed surface S (orientable or non-orientable), and a given graph G, there is a linear time algorithm which returns one of the following:*

(1) *an embedding of G in S, or*

(2) *a minor of G which is not embeddable in S and is minimal with this property.*

The advantages of our algorithm are the following:

1. The proof is considerably simpler: it needs only about 10 pages, and some results (with rather accessible proofs) from graph minors theory, while Mohar's original algorithm and its proof occupy more than 100 pages in total.
2. The hidden constant (depending on the genus g of the surface S) is much smaller. It is single exponential in g, while it is doubly exponential in Mohar's algorithm.

Our algorithm has several appealing features. It applies techniques which were used in [31, 32] 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 [35]).

Computing the genus of graphs in certain minor closed families has mainly theoretical (but also some practical) importance. We refer to Mohar and Schrijver [27] 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 apex graphs [24]. (Recall that a graph G is an *apex graph* if it contains a vertex v such that $G - v$ is planar.)

It was most annoying that the status of the genus problem was open even for such a simple case as graphs of bounded tree-width. As a spinoff of our main result, we give another linear time algorithm.

Theorem 1.2 *There is a linear time algorithm to compute the genus, the Euler genus, the nonorientable genus and to construct corresponding minimum genus embeddings of graphs of bounded tree-width.*

Theorem 1.2 proves a conjecture by Robertson (private communication), see [28], and solves one of the most annoying long standing open question about complexity of algorithms on graphs of bounded tree-width.

1.3 Overview of the algorithm of Theorem 1.1

We now turn to our algorithm for constructing an embedding of a given graph G into the surface S of Euler genus g, if one exists. Otherwise, we discover a minimal forbidden minor for embeddability in S contained in G. The crux of the matter is to understand which vertices are irrelevant with respect to *any* embedding of G into the surface S of Euler genus g. We prove that the removal of these vertices yields a graph of bounded tree-width if G has an embedding in S.

We may assume that the minimum degree of G is at least 2. 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 contracting a matching M_i with at least $\epsilon|G_{i-1}|$ vertices for some small but constant $\epsilon > 0$, or by deleting a stable set I of $\epsilon|G_{i-1}|$ vertices, each of degree 2. In the latter case, every vertex in I has the same pair of neighbors as several other vertices in I. In this case, we also add edges xy to G_{i-1} for every x, y for which there is a vertex $z \in I$ whose neighbors are x and y. 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 into some surface of Euler genus at most g (and of the same orientability type as S), or give a minimal forbidden minor for the surface S contained in 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} to help in constructing an embedding for G_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 G_i has an embedding in some surface of Euler genus at most g if and only if the subgraph G''_i obtained from G'_{i+1} by uncontracting the matching M_i restricted on the vertices of G'_{i+1} has such an embedding. Note that G_{i+1} may be obtained from G_i by deleting a stable set I of $\epsilon|G_i|$ vertices of degree 2. In this case, it is actually easy to construct an embedding of G_i from G_{i+1}. Therefore, we shall only concentrate on the case when G_{i+1} is obtained from G_i by contracting the matching M_i.

Suppose that we have an embedding of G'_{i+1}, which has tree-width at most $h(g)$ for some function h of g and satisfies the property mentioned above. We then uncontract the matching M_i of G_i restricted on G'_{i+1} to obtain the graph G''_i. It is easy to see that the tree-width of G''_i is at most $2h(g)$. Therefore, we can find an embedding of G''_i using the standard dynamic programming approach; otherwise, we can detect a minimal forbidden minor for the surface S in G'_i. By the property of G'_{i+1}, we can extend the embedding of $G''_i \subseteq G'_i$ to G_i that is obtained from G_{i+1} by uncontracting the 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 $h(g)$, and G_{i-1} has an embedding in some surface of Euler genus at most g if and only if the subgraph G''_{i-1} obtained from G'_i by uncontracting the matching M_{i-1} of G_{i-1} restricted on the vertices of G'_i has. This is our main challenge in this algorithm, and we shall describe more details in Section 5.

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 [5].
2. Finding an embedding of a bounded tree-width graph.
3. Reducing the tree-width by deleting irrelevant vertices.

For the third ingredient, we shall use the technique in [31, 32]. The results in [31, 32] show 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 seminal algorithm of Robertson and Seymour in [35] which solves the same problem for arbitrary graphs in cubic time. The proof in [31, 32] uses several ideas underlying Robertson and Seymour's algorithm.

For the second ingredient, we actually get a much stronger result, as mentioned above. Namely, if a given graph G has bounded tree-width, then the algorithm can determine the minimum Euler genus of G, and furthermore, returns an embedding of G into the surface of minimum (Euler) genus.

These tasks are presented in more detail in Sections 2, 3, and 4, respectively.

1.4 Overview of the genus algorithm for graphs of bounded tree-width

Our algorithm for genus of graphs of bounded tree-width follows the standard dynamic programming approach of Arnborg and Proskurowski [1] and uses some lemmas that establish that there is only a bounded number of "boundary schemes" which need to be memorized when computing the genus of the union of two graphs, whose intersection contains a bounded number of vertices.

Let G be a connected graph, and let $T \subseteq V(G)$ be a separating set of G. Suppose that G is embedded in a surface S, and let $G = G_1 \cup G_2$ be a separation of G with $V(G_1 \cap G_2) = T$. Let $W = v_1 e_1 v_2 e_2 \dots v_k e_k v_1$ be a facial walk. A triple $e_{i-1} v_i e_i$ in W (including the triple $e_k v_1 e_1$) is called a *mixed angle* if one of the edges e_{i-1} or e_i belongs to $E(G_1)$, the other one to $E(G_2)$. In particular, the vertex v_i of a mixed angle is in T.

Let M be the multigraph embedded in S obtained by joining vertices of consecutive mixed angles in the facial walks such that for every facial walk W with $r > 0$ mixed angles, there is a closed walk M_W of length r in M. Clearly, $V(M) \subseteq T$, and the number of edges of M is equal to the number of mixed angles. We assume that $E(M) \cap E(G) = \emptyset$.

Two embeddings Π_1 and Π_2 of G are said to be of the *same type* with respect to the separation $G = G_1 \cup G_2$ if they have the same (isomorphic) graph M of mixed angles, the extended embeddings $\tilde{\Pi}_1$ and $\tilde{\Pi}_2$ of $\tilde{G} = G \cup M$ induce the same rotation system on M and the same corresponding partition $\{F_i \mid 1 \leq i \leq r\}$ of faces of M.

We say that the embedding Π is *T-homogeneous* if every embedding of smaller or equal genus has at least as many mixed angles as Π.

Lemma 1.3 *Every S-homogeneous embedding Π of G has the following properties:*

(a) If F is a mixed face and $v \in T$, then F and v have at most two mixed incidences.

(b) For every $v \in S$ there is at most one mixed face with two mixed incidences with v.

(c) For every pair $u, v \in T$, there are at most $c(|T|)$ mixed faces which have mixed incidence with both, u and v, where the value $c(|T|)$ depends only on the size of T.

The main fact needed for the algorithm of Theorem 1.2 to work is the following result whose proof is deferred for the full version of the paper.

Lemma 1.4 *The minimum genus taken over all T-homogeneous embeddings of G is equal to the genus of G.*

2 Finding a large induced matching

In the Appendix we review the greedy algorithm for finding a maximal matching in the subgraph induced on some vertex set T. We now give an algorithmic proof of the following result.

Theorem 2.1 *Let g be a fixed positive integer, and let G be a graph of order n with minimum degree at least 2 and with at most $4n$ edges. Let $d > 8g \cdot 2^{16\sqrt{g}}$ and $\epsilon = d^{-6}$. Then we can find in linear time one of the following:*

(1) *A vertex set Z of at least $5\epsilon|G|$ vertices of degree 2, each of which has the same pair of neighbors as at least one other vertex in Z.*

(2) *A matching M in the subgraph of G induced by the vertices of degree at most d that has size $|M| > 2d\epsilon|G|$.*

(3) *A minor G' of G with at least $2^{8\sqrt{g}}|G'|$ edges.*

(4) *A minor G'' of G which is isomorphic to $K_{3,4g+4}$.*

Proof. The proof outlined here is similar to a proof given in [5] and in [33]. Let S be the set of vertices of degree at most d in G. The lower bound on d and the imposed conditions on the degrees and on the number of edges in G imply that $|S| \geq (1 - 6/(d-1))|G|$. So, S contains almost all vertices of G. We then choose a maximal matching M within $G(S)$. This can be done by Algorithm A.5. We can assume that M has size at most $2d\epsilon|G|$ or we are done. Then, since the vertices in M have degree $\leq d$ and $|M|$ is small, $M \cup N(M)$ is also small. More precisely, $|M \cup N(M)| \leq |M| + (d-1)|M| \leq 4d^2\epsilon|G|$. Therefore, $S^\circ = S - (M \cup N(M))$ is large. Since M is maximal, edges with one endpoint in S° have their other endpoint in $V - S$. So each vertex in S° has at least 2 and at most d neighbors in $V - S$. For each of these vertices in turn, we choose a pair of its neighbors in $V - S$. By contracting (for each $v \in S^\circ$) one of the edges towards such a pair, we get a minor of G with vertex set $V - S$ and with many edges (possibly some of them parallel to each other). If we choose $2^{8\sqrt{g}}|V - S|$ such edges, where no two of them are parallel to each other, then they give rise to a dense minor on the vertex set $V - S$, yielding outcome (3). Otherwise we construct as a minor a simple graph F on vertex set $V - S$ with fewer than $2^{8\sqrt{g}}|V - S|$ edges. Let S^* be the subset of vertices of S° used to get the edges of F, and let $S' = S^\circ - S^*$. Note that $|S^*| < 2^{8\sqrt{g}}|V - S|$.

If S' contains more than $2^{8\sqrt{g}}|V - S| + 5d\epsilon|G|$ vertices of degree 2, then there is a subset Z of these vertices with $|Z| \geq 5\epsilon|G|$, giving outcome (1) of the theorem. Note that such a subset can be obtained by a greedy selection. So we may assume that this is not the case, and hence S' contains a subset S'' of vertices of degree at least three whose cardinality is at least

$$|S''| \geq (1 - 6/(d-1) - 4d^2\epsilon - 5d\epsilon)|G| - 2 \cdot 2^{8\sqrt{g}}|V - S| \geq 4(g+1) \cdot 2^{16\sqrt{g}}|V - S|.$$

For each vertex $s \in S''$ choose three of its neighbors in $V - S$, and denote them by $T(s) = \{s_1, s_2, s_3\}$. If $4g+4$ vertices s have the same triple $T(s)$, then we get a $K_{3,4g+4}$ minor which is outcome (4). Otherwise, a subset Z' of $2^{16\sqrt{g}}|V - S|$ vertices in S'' will have pairwise distinct triples.

Consider a vertex $v \in V - S$ and let Z'_v be the set of all vertices s in Z' such that $v \in T(s)$. If $|Z'_v| \geq 2^{8\sqrt{g}} \deg(v)$, then we get a minor of the graph G on the neighbors of v which satisfies (3). Finally, if this does not happen for any of the vertices $v \in V - S$, then the number of triples and hence the number of elements in Z' is bounded:

$$|Z'| \leq \frac{1}{3} \sum_{v \in V-S} |Z'_v| \leq \frac{1}{3} \cdot 2^{8\sqrt{g}} \sum_{v \in V-S} \deg(v) \leq \frac{2}{3} \cdot 2^{16\sqrt{g}} |V - S|,$$

where we have used the fact that the subgraph on $V - S$ has average degree bounded by $2^{8\sqrt{g}}$ (or we get outcome (3)). The derived inequality contradicts the fact that $Z' \geq 2^{16\sqrt{g}}|V - S|$. This completes the proof. $\Box$

Let us observe that if we find a minor G' of G with at least $2^{8\sqrt{g}}|G'|$ edges in Theorem 2.1, then we can find a minimal forbidden minor for embeddability in surfaces of Euler genus g in linear time. Specifically, this is a corollary of the following result in [33].

Theorem 2.2 *Suppose G has at least $2^k|G|$ edges. Then we can detect a K_k-minor in G in linear time.*

Since the maximum degree of the vertices in the matching M obtained as outcome (2) of Theorem 2.1 is d, this outcome can be used to get a large induced matching.

Theorem 2.3 *Suppose there is a matching M that comes from outcome (2) of Theorem 2.1. Then there is a submatching M' of M of size $M' \geq \frac{|M|}{2d} > \epsilon n$ which is induced.*

Having $K_{3,4g+4}$ or $K_{8\sqrt{g}}$ (extracted by applying Theorem 2.2) as a minor, we obtain from it (in constant time) a minimal forbidden minor for the surface S of Euler genus g. In summary:

Theorem 2.4 *Let G be a graph with minimum degree at least 2 with at most $4n$ edges. Let $d > 8g \cdot 2^{16\sqrt{g}}$ and $\epsilon = d^{-6}$. Then we can find in linear time one of the following:*

1. *A vertex set Z of at least $5\epsilon|G|$ vertices of degree 2, each of which has the same pair of neighbors as at least one other vertex in Z.*
2. *An induced matching M in G containing at least $\epsilon|G|$ edges.*
3. *A minor G' of G which is a minimal forbidden minor for the surface S of Euler genus g.*

3 Tree-width bounded case

Our algorithm needs to test whether or not a given graph G has bounded tree-width. This can be done in linear time by the algorithm of Bodlaender [5]. In fact, if the tree-width is bounded, say at most $f(k)$, we would like to have a tree-decomposition of tree-width at most $f(k)$.

Theorem 3.1 *For every fixed l, there is a linear time algorithm to determine whether or not a given graph G has tree-width at most l. Moreover, if this is the case, then the algorithm gives a tree-decomposition of tree-width at most l.*

The above theorem was slightly improved by Perković and Reed [29].

We now state the main theorems of this section.

Theorem 3.2 *Suppose G has tree-width at most k. Then there is a linear time algorithm to decide whether or not G can be embedded into a surface of (Euler) genus at most g. Moreover, if the answer is yes, then the algorithm gives an embedding of G, and if the answer is no, then the algorithm outputs a minimal forbidden minor.*

Proof. This can be achieved by using the standard dynamic programming approach. See [5]. We need an actual upper bound $a(g)$ on the order of the largest minimal forbidden minor for the surfaces of Euler genus g. Such a bound does not follow from the non-constructive graph minors proof of Robertson and Seymour. It follows from the constructive proof of Mohar [21]. The best known bound is given explicitly in Seymour's unpublished paper [37]. See also [25]. Therefore, by the standard dynamic programming [5], we can find each minimal forbidden minor in linear time, if one exists.

Alternatively, a recent result of Adler, Grohe and Kreutzer [2] gives rise to a minimal forbidden minor, if one exists. □

More generally, we can prove the following theorem, which answers a conjecture posed by Neil Robertson (private communication), see [28].

Theorem 3.3 *There is a linear time algorithm to compute the genus, the Euler genus, the nonorientable genus and to construct corresponding minimum genus embeddings of graphs of bounded tree-width.*

4 Bounding the tree-width of planar graphs

In this section, we are given a planar graph. What we shall do here is that, we are going to bound tree-width by deleting many vertices at once, and we have to do this in linear time. How do we find such vertices in every planar graph? Before we address the question, let us state a result which tells that the vertices deep inside the planar part are irrelevant. Specifically, the following theorem is proved in [18].

We say that a set U of vertices of G is *irrelevant* if the graph G embeds in some surface (of Euler genus at most g) if and only if $G - U$ does.

Theorem 4.1 *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 an h-wall W, where $h \geq 2g+1$, then the set of all vertices of W, that have face-distance in the wall W at least g from the outer cycle of W, is irrelevant. In particular, every vertex surrounded by a $2g$-wall in Q is irrelevant.*

Theorem 4.1 says that for any planar subgraph, if there is a "planar" $(2g+1)$-wall, then we can delete the middle vertex, and actually, we can keep deleting irrelevant vertices until there is no flat $(2g+1)$-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 adapted by Reed, Robertson, Seymour and Schrijver [31, 32], 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 until the resulting graph has no flat $(2g+1)$-wall. Let us state this as a lemma.

Lemma 4.2 *Suppose that Q is a planar subgraph of a graph 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. Then there is a linear time algorithm to find an irrelevant vertex set $X \subseteq V(Q)$. Furthermore, this algorithm can output the graph $Q - X$ such that it does not contain a $(2g+1)$-wall, and no matter how G is embedded into the surface S of Euler genus at most g, the embedding can be changed at vertices of Q such that all vertices in X are contained in a disk of the embedding of G in S.*

The proof of Lemma 4.2 is exactly same as what Reed, Robertson, Schrijver and Seymour did. We omit the detail, and refer to the papers [31, 32].

Actually, we shall need the following extension of Lemma 4.2.

Lemma 4.3 *Suppose Q is a planar subgraph of a graph 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. Let $C_1, \dots, C_l$ be disjoint faces of*

Q. Then there is a linear time algorithm to find a vertex set X such that each vertex in X is surrounded by a $2g$-wall in $Q - X$ that does not contain any face of $C_1, \dots, C_l$, and deleting the vertices of X can be shown not to change the problem of finding an embedding of G into the fixed surface S of Euler genus at most g, by a sequence of applications of Theorem 4.1, and $Q - X$ contains no $2gl$-wall. Furthermore, no matter how G is embedded into the surface S of Euler genus at most g, the deleted vertices are in the disk of the embedding of G.

The proof of Lemma 4.3 is also exactly same as what Reed, Robertson, Schrijver and Seymour did. We omit the detail, and refer to the papers [31, 32].

5 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 2 (in which case we shall write $M_i = \emptyset$), see Theorem 2.4. In this section, an integer i is fixed such that $M_i \neq \emptyset$ and the following hypothesis is assumed.

Hypothesis 5.1 *The graph G_{i+1} can be embedded into a surface S of Euler genus at most g (and cannot be embedded in any surface of smaller genus and the same orientability type). 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} satisfying the following conditions:*

1. *G'_{i+1} has tree-width at most $h(g)$ for some function h of g (to be determined later).*
2. *Let G''_i be the subgraph of G_i obtained from G'_{i+1} by uncontracting the matching M_i restricted to the vertex set of G'_{i+1}. Then G''_i can be obtained from G_i by deleting disjoint disks $D_1, D_2, \dots$ and every embedding of G''_i in S can be extended to an embedding of G_i into S by adding these disks D_j into faces of G''_i.*
3. *Every vertex of $G_i - G''_i$ is surrounded by a $2g$-wall that is flat in every embedding of G_i in S.*

The objective of this section is to start with G'_{i+1} satisfying the hypothesis and to either construct G'_i satisfying the hypothesis for $i-1$, or to find a minimal forbidden minor for the surface S in time $O(|G_{i-1}|)$.

From G'_{i+1} we construct G''_i simply by uncontracting M_i. It is easy to see that G''_i has tree-width at most $2h(g)$, since G'_i has tree-width at most $h(g)$. So, we can find an embedding of G''_i in S by using the algorithm for graphs of bounded tree-width. By Hypothesis 5.1, we can extend the embedding of G''_i to G_i. We now construct G'_i with the help of G''_i.

We assume that if the tree-width of a graph embedded in S is at most $4h(g)$, then this graph does not contain a flat $16g^2 \cdot s(g)$-wall for some function s of g, which will be determined later; see Theorem 5.2.

In order to construct G'_i or find a minimal forbidden minor for the surface S, we need several phases described below. We first uncontract the matching M_{i-1} restricted on G''_i. Note that G_i may have been obtained from G_{i-1} by deleting a stable set of $\epsilon|G_{i-1}|$ vertices, each of which has degree 2. In this case, it is actually easy to construct an embedding of G_{i-1} from G_i, and a desired subgraph G'_i, since the vertices of degree two are easy to embed into faces of G_i, and of G'_i. Therefore, we shall only consider the case when G_i is obtained from G_{i-1} by contracting the matching M_{i-1}.

Phase 1. Extend the embedding of G''_i to G_i by adding the disks $D_1, D_2, \dots$ and then delete some irrelevant vertices. If the embedding does not extend, then return a minimal forbidden minor for embeddability in S.

We need to consider vertices in the disjoint disks $D_1, D_2, \dots$ that have been temporarily deleted from G_i to construct the graph G''_i (see the last assumption of Hypothesis 5.1). In each disk D_j, we first add the "first neighborhood" of D_j (to make sure that we deal with all the matching edges of M_{i-1} that were contracted). By the first neighborhood we mean all facial cycles (in the corresponding embedding of G_i) that intersect D_j in at least one vertex. Let us denote this intermediate graph by $Q'_j \subseteq G_i$. Next, we uncontract the matching M_{i-1} restricted to the vertices in Q'_j. Let $Q_j \subseteq G_{i-1}$ be the resulting graph.

If Q_j is planar, then by Theorem 4.1, the vertices surrounded by a $2g$-wall in Q_j are irrelevant for G_{i-1}. Therefore we can delete all irrelevant vertices in time $O(|Q_j|)$, as shown by Lemma 4.2. In this way we destroy all planar subgraphs that contain flat walls of width at least $2g+1$. Therefore, by Theorem A.4, the tree-width of the resulting graph is at most $h(g)$.

We need to consider the case when Q_j is not planar. Note that Q'_j is planar. By the algorithm of Hopcroft and Tarjan [10], we can easily make Q'_j 3-connected in such a way that each 1- or 2-connected component can be dealt with separately.

Therefore, we may assume that each vertex v of degree at least 3 in Q'_j has the wheel property, i.e., the deletion of v results in a new face bounded by a cycle C, and each neighbor of v is on C. We call this cycle *the first neighbor cycle* of v.

The matching M_{i-1} is induced in G_{i-1} by Theorem 2.4. Therefore, if Q_j is not planar, then each of non-planar parts must involve a matching edge in Q_j. Let us look at each non-planar part more closely. Let e be a matching edge, and v be the vertex after the contraction of e. By the previous observation, if the uncontraction of v results in a non-planar part, the first neighbor cycle of v together with e includes a subdivision of $K_{3,3}$ (or K_5) such that e is an edge of this subdivision. We will say that e is a *genus boosting edge*.

We now check, for each matching edge $e \in M_{i-1} \cap Q_j$, if it is a genus boosting edge. Since Q_j has at most $3|Q_j| + 2g$ edges, it is easy to see that we can test this for all such edges in time $O(|Q_j|)$. Furthermore, we can adequatly treat the genus boosting edges by using the next result.

Theorem 5.2 *Let Q be a graph and M an induced matching in Q. Suppose that $Q' = Q - V(M)$ is a planar 3-connected graph and that each edge $e \in M$ has all its neighbors on a face F_e of the planar embedding of Q'. If e is a genus boosting edge, then we shall say that F_e is a* genus boosting face. *If the number of distinct genus boosting faces F_e is at least $s(g)$, then the Euler genus of Q is more than g.*

If there are more than $s(g)$ genus boosting faces, then we can find a minimal forbidden minor, by adding non-planar parts to Q'_j. This is simply because we can find one of the minimal forbidden minors by adding non-planar crosses to G''_i, in time $O(|V(G_{i-1})|)$.

Suppose now that there are at most $s(g)$ genus boosting faces. We now delete all vertices contained in the genus boosting edges in Q_j. Let Q''_j be the resulting graph, which is planar. In this case, we shall delete all irrelevant vertices X in time $O(|Q_j|)$ by using Lemma 4.3. When we apply the lemma, the genus boosting faces in Q''_j together with the outer face boundary of D_j play the role of the faces $C_1, \ldots, C_l$ $(l \leq s(g))$ in Lemma 4.3. Therefore, we can delete all irrelevant vertices with respect to the faces $C_1, \ldots, C_l$ $(l \leq s(g))$ in time $O(|Q''_j|)$ by Lemma 4.3. After deleting them, the resulting subgraph of Q''_j does not have a $2gs(g)$-wall. We then put the genus boosting edges of M_{i-1} back to each of the disks, if they exist, and we denote the resulting subgraph of Q_j by D'_j.

Since we destroy all flat $2gs(g)$-walls for each D_j, each disk D'_j has bounded tree-width by Theorem A.4. Let us observe that all deleted vertices are surrounded by a $2g$-wall in Q_j (and hence in G_{i-1}). This completes this phase.

Phase 2. Putting things together.

We first take the graph G'' which is obtained from G''_i by uncontracting all edges of M_{i-1} contained in it. From Phase 1, we have got disjoint disks $D'_1, D'_2, \ldots$, each of which has bounded tree-width, by deleting irrelevant vertices in G_{i-1}. We now put them together. More precisely, we add $D'_1, D'_2, \ldots$ to G''. Let $G' \subseteq G_{i-1}$ be the resulting graph. Let us recall that every vertex deleted during Phase 1 is surrounded by a $2g$-wall in G_{i-1}. Therefore, they are irrelevant by Theorem 4.2.

The tree-width of G' may be bigger than $h(g)$. In this case, we need to delete some irrelevant vertices in G_{i-1} again.

Let us show that G' has tree-width at most $4h(g)$. To see this, let us first observe that the graph G'' does not contain a flat $(8g+1)$-wall, and no disk $D'_1, D'_2, \dots$ (obtained in Phase 1) contains a flat $2gs(g)$-wall. It follows that G' does not contain a flat $16g^2s(g)$-wall and that the tree-width of G' is at most $16h(g)$. This conclusion has a technical proof (given in the full paper) and, roughly speaking, is based on the following fact: If we have a graph (in our case G'') of bounded tree-width embedded in a surface of bounded genus and we add planar graphs (in our case $D'_1, D'_2, \dots$) into some faces in such a way that all new vertices are close to the corresponding boundary in the "face-distance", then the resulting graph (in our case G') has bounded tree-width.

Since G' has bounded tree-width, we can either find an embedding of G' into the surface S or detect a minimal forbidden minor for S by using Theorem 3.2.

Suppose first that there is an embedding of G'. We need to reduce G' so that it has tree-width at most $h(g)$. Actually, we need to find all planar subgraphs and all flat walls of size at least $2g$, but this can be also done in time $O(|G'|)$, since, again, G' has tree-width at most $4h(g)$. By Theorem 4.1, the vertices surrounded by a $2g$-wall in each of the planar parts are irrelevant for G_{i-1}. In this case, we shall delete all irrelevant vertices in time $O(|G'|)$ by using Lemma 4.2.

But we need to make sure that after deleting irrelevant vertices in Phase 2, all the deleted vertices are still in the disk, and surrounded by a $2g$-wall in G_{i-1}. We now argue that this is, indeed, true. Suppose not. We pick up the "last" vertex v which was deleted in Phase 2, i.e, in the current graph W, all the deleted vertices are surrounded by $2g$-walls, but when we delete v from W, there is a vertex w which was deleted in either Phase 1 or 2, but w is not surrounded by any $2g$-wall in $W \setminus \{v\}$. Let $C_1, \dots, C_g$ be the g nested cycles of a $2g$-wall surrounding w in W, and let $C'_1, \dots, C'_g$ be the g nested cycles of a $2g$-wall surrounding v in W. Assume that v is in one of $C_1, \dots, C_g$, say C_l. If we cannot reroute C_l using C'_1, this means that C'_1 hits both C_{l-1} and C_{l+1}. Inductively, it can be shown that if we cannot reroute $C_{l-j}, \dots, C_l, \dots, C_{l+j}$ using $C'_1, \dots, C'_j$, this means that C'_j hits both C_{l-j-1} and C_{l+j+1}. However, clearly for some $j \le g/2$, we can reroute $C_{l-j}, \dots, C_l, \dots, C_{l+j}$ using $C'_1, \dots, C'_j$. So, there are $2g$ nested cycles for $w \in W \setminus \{v\}$. Since w is surrounded by a $2g$-wall, we may assume that there are g disjoint paths joining C_1 and C_g. The vertex v may be on one of these paths. By a similar argument, we can reroute these g disjoint paths using the cycles $C'_1, \dots, C'_g$. Therefore, w is surrounded by a $2g$-wall in $W \setminus \{v\}$, a contradiction.

Therefore, by Theorem A.4, the resulting graph (which is the graph G''_{i-1} in our notation) has tree-width at most $h(g)$. Let us repeat the important fact that every deleted vertex in Phases 1 and 2 is in a disk, and surrounded by a $2g$-wall in G_{i-1}. We now contract all the edges of M_{i-1}, and let G'_i be the resulting graph. Then we output the graph G'_i. By Theorem 4.2, G_{i-1} can be embedded into a surface S of Euler genus at most g if and only if G''_{i-1} can, where G''_{i-1} is a subgraph of G_{i-1} obtained from G'_i by uncontracting the matching M_{i-1} restricted to the vertex set on G'_i. This finishes the last phase.

Hence, in all phases, in time $O(|G_{i-1}|)$, either we can detect a minimal forbidden minor or we can bound the tree-width of G'_i, i.e., there are no flat walls of size $2g+1$ in G'_i.

Let us observe that the deleted vertices can be put back to the embedding of G''_{i-1} (if one exists), which is obtained from G'_i by uncontracting the submatching of the matching M_{i-1} restricted on the vertices of G'_i, since these vertices must be embedded into a disk of the embedding of G_{i-1}, and surrounded by a $2g$-wall.

In summary, either we can find a minimal forbidden minor for the surface S or we temporary delete some vertices in G_i to obtain the new graph G'_i, which is a subgraph of G_i and has tree-width at most $h(g)$. Moreover, as discussed above, G'_i satisfies Hypothesis 5.1 for $i-1$, instead of i. As observed above, this process can be done in time $O(|V(G_{i-1})|)$, which is linear.

This is one of the key parts of the whole algorithm.

6 Algorithm

Finally, we are ready to present the complete algorithm.

Algorithm for Theorem 1.1

We assume a positive integer g and a surface S of Euler genus g are fixed.

Input: A graph G of order n.

Output: Either an embedding of G in S or a minor of G which is not embeddable in S and is minimal with this property.

Running time: $O(f(g)n)$ 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 on a matching M_i with at least $\epsilon|G_{i-1}|$ edges for some small but constant $\epsilon > 0$, or deleting a stable set of $\epsilon|G_{i-1}|$ vertices each of degree 2. Every deleted vertex has the same pair of neighbors as another vertex of degree 2 that is not deleted.

This step can be done as discussed in Section 2. As pointed out there, if we find a minimal forbidden minor in Theorem 2.4, 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 matching or a desired stable set in time $O(|G_i|)$ as explained in Section 2. 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. Since $|G_b| < B$, this can be done in constant time.

We recursively apply Steps 3 and 4 for $i = b, b-1, \ldots$.

Step 3. For the ith iteration, either find a subgraph G'_i of G_i satisfying Hypothesis 5.1 or a minimal forbidden minor for the surface S. This can be done, as discussed in Section 5, in time $O(|G_{i-1}|)$.

Step 4. Extend the embedding of G'_i to G_i.

This can be done in time $O(V(|G_i|))$ by applying the planarity algorithm [4, 6, 11, 41]. Note that all vertices of $G_i - G'_i$ can be embedded into a disks $D_1, D_2, \ldots$. Therefore, we just need the planarity algorithm for this task.

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 $f(g)$. The most expensive part is dealing with $s(g)$ in Section 5. The other 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. Therefore, $f(g)$ only depends on $s(g)$ and $c(\epsilon)$, and the algorithm shows that $f(g) = Poly(s(g) \cdot c(\epsilon))$. Thus, $f(g)$ is single exponential as we claimed.

This completes the (sketch of the) proof of Theorem 1.1. $\square$

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] 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.

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

[4] 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.

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

[6] 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.

[7] 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.

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

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

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

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

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

[13] M. Juvan, J. Marincek and B. Mohar, Obstructions for simple embeddings, *Australas. J. Combin.* **38** (2006), 3–25.

[14] M. Juvan, J. Marincek and B. Mohar, Embedding a graph into a torus in linear time, Integer programming and combinatorial optimization (Copenhagen, 1995), 360–363, *Lecture Notes in Comput. Sci.* **920**, Springer, Berlin, 1995

[15] M. Juvan and B. Mohar, Extending 2-restricted partial embeddings of graphs, *Europ. J. Combin.* **26** (2005), 339-375.

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

[17] K. Kawarabayashi and B. Mohar, Graph and Map Isomorphism and all polyhedral embeddings in linear time, to appear in Proc. 40th ACM Symposium on Theory of Computing (STOC'08).

[18] B. Mohar, Combinatorial local planarity and the width of graph embeddings, *Canadian J. Mathematics* **44** (1992), 1272–1288.

[19] B. Mohar, Projective planarity in linear time, *J. Algrorithms* **15** (1993), 482–502.

[20] B. Mohar, Obstructions for the disk and the cylinder embedding extension problems, *Combin. Probab. Comput.* **3** (1994), 375–406.

[21] B. Mohar, Embedding graphs in an arbitrary surface in linear time, In *Proc. 28th Ann. ACM STOC*, Philadelphia, ACM Press, 1996, pp. 392–397.

[22] B. Mohar, Obstructions for 2-Möbius band embedding extensions, *Siam J. Discrete Math.* **10** (1997), 57–72.

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

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

[25] B. Mohar, Graph minors and graphs on surfaces. in *"Surveys in Combinatorics, 2001 (Sussex)"*, London Math. Soc. Lecture Note Ser. **288**, Cambridge Univ. Press, Cambridge, 2001, pp. 145–163.

[26] B. Mohar, Universal obstructions for embedding extension problems, *Australas. J. Combin.* **36** (2006), 39-68.

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

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

[29] 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.

[30] 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, 1997, pp. 87–162.

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

[32] 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.

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

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

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

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

[37] P. D. Seymour, A bound on the excluded minors for a surface, *preprint.*

[38] P. Seymour and R. Thomas, Graph searching and a min-max theorem for tree-width, *J. Combin. Theory Ser. B* **58** (1993), 22–33.

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

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

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

A Basic definitions

A.1 Graphs on a fixed surface

For basic graph theoretic notation, we refer the reader to the book of Bondy and Murty [3]. If G is a graph, we denote by $|G|$ the number of vertices and by $\|G\|$ the number of edges of G.

A *surface* is a compact connected 2-manifold without boundary. We assume familiarity with basic notions of surface topology, like genus and Euler's formula. We define the *Euler genus* of a surface S as $2 - \chi(S)$, where $\chi(S)$ is the Euler characteristic of S. An *I-arc* in a surface Σ is a subset of Σ homeomorphic to $[0, 1]$. An *O-arc* is a subset of Σ homeomorphic to a circle. Let G be a graph that is embedded in Σ. To simplify notation we do not distinguish between a vertex of G and the point of Σ used in the embedding to represent the vertex, and we do not distinguish between an edge and the arc on the surface representing it. We also consider G as the union of the points corresponding to its vertices and edges.

A graph G is *embedded* in a topological space X if the vertices of G are distinct elements of X and every edge of G is simple arc connecting in X the two vertices which it joins in G, such that its interior is disjoint from other edges and vertices. An *embedding* of a graph G in the topological space X is an isomorphism of G with a graph G' embedded in X. In this case, G' is said to be a representation of G in X. If there is an embedding of G into X, we say that G *can be embedded* into X.

If a graph G is embedded in Σ, a *region* or *face* of G in Σ is a connected component of $\Sigma \setminus (E(G) \cup V(G))$. Every region is an open set. We use the notation $R(G)$ for the set of regions of G. The embedding is said to be a *2-cell embedding* if every region is homeomorphic to a disc.

If $\Delta \subseteq \Sigma$, then $\overline{\Delta}$ denotes the *closure* of Δ, and the boundary of Δ is $\partial\Delta = \overline{\Delta} \cap \overline{\Sigma \setminus \Delta}$. An edge e (or a vertex v) is *incident* with a region r if $e \subseteq \partial r$ ($v \in \partial r$).

A subset of Σ meeting the embedded graph only in vertices of G is said to be *G-normal*. If an O-arc is G-normal then we call it a *noose*. We say that a disc $D \subset \Sigma$ is *bounded* by a noose N if $N = \partial D$. A graph G, which is 2-cell embedded in a surface Σ, has *face-width* or *representativity* at least θ if every noose, which intersects G in fewer than θ vertices is contractible (null-homotopic) in Σ. Alternatively, the *face-width* or *representativity* of G is equal to the minimum number of facial walks whose union contains a cycle which is non-contractible in Σ. See [28, Chapter 5] for further details.

A.2 Tree-width, walls and grid minors

A *bramble* in a graph G is a set of trees in G, every two of which intersect or are joined by an edge. A *hitting set* for a bramble β is a set of vertices of G intersecting the vertex set of each tree of β. The *order* of a bramble β, denoted $ord(\beta)$, is the minimum size of a hitting set. For any set W of vertices, the set β_W of trees of G containing more than half the vertices of W is a bramble since any two such trees intersect. We now characterize graphs which have no brambles of order l, using tree decompositions.

A *tree decomposition* of a graph G consists of a tree T and a subtree S_v of T for each vertex v of G such that if uv is an edge of G then S_u and S_v intersect. For each node t of the tree, we let Y_t be the set of vertices v of G such that $t \in S_v$. The *width* of a tree decomposition is the maximum of $|W_t|$ over the nodes t of T, and the *tree-width* of G is the minimum width of its tree decompositions.

It is not hard to see that for every bramble β in G and for every tree decomposition of G, there is a node t such that W_t is a hitting set for β. This implies that the tree width of G is at least the maximum order of a bramble. Seymour, and Thomas [38] showed that this bound is tight, proving:

Theorem A.1 *The maximum order of a bramble in G is equal to its tree-width.*

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

It is convenient to delete the vertices of degree 1 from W_r and refer to the resulting 2-connected planar graph as an r-wall as well.

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 $P_a \times P_b$. The (8×5)-wall is shown in Figure 1.

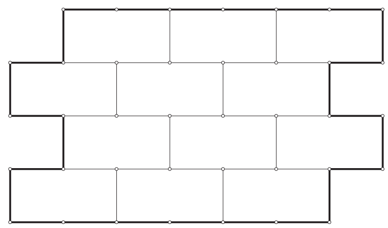

Figure 1: The (8×5)-wall and its outer cycle

The main result of Graph Minors V [34] says the following.

Theorem A.2 *For every positive integer r, there exists a constant $f(r)$ such that if a graph G has tree-width at least $f(r)$, then G contains an r-wall as a (topological) minor.*

The best currently known upper bound for $f(r)$ is given in [36], (see also [7, 30]). It is 20^{64r^5}, and 20^{2r^5} for the $(r \times r)$-grid minor. The best known lower bound on $f(r)$ is of order $\Theta(r^2 \log r)$, see [36]. But for planar graphs, Robertson, Seymour and Thomas [36] proved the following result which yields essentially best possible bound $6r$ for the existence of an r-wall.

Theorem A.3 *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.*

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 (see Figure 1) bounds a disk in S and H is contained in this disk. The following theorem was proved by Thomassen [40].

Theorem A.4 *For every r and g, there is a value $f(g, r)$ satisfying the following. If a graph G is embedded in a surface of Euler genus at most g and 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 $f(g, r)$.*

A.3 Greedy algorithm for finding a maximal matching

Below we describe an algorithm to find a maximal matching M in the graph induced by T for some vertex set T.

Algorithm A.5 *Let M be the empty graph.*

For $v \in T$, let v be unmarked.

For each edge $uv \in E(T)$:

If u is unmarked and v is unmarked then

Add u and v to $V(M)$ and add uv to $E(M)$

Mark u and v.

We claim that M is a maximal matching in S. Suppose not. Let $uv \in E(S)$, $u, v \notin V(M)$. When uv is considered in the algorithm, at least one of u or v was marked. So one of u or v was in $V(M)$. But no vertex added to M is later removed, a contradiction. Let us observe that Algorithm A.5 can be easily implemented so that its running time is linear.