An algorithm for embedding graphs in the torus*

Martin Juvan† and **Bojan Mohar**‡
Department of Mathematics, University of Ljubljana
1111 Ljubljana, Slovenia

Abstract

An efficient algorithm for embedding graphs in the torus is presented. Given a graph G, the algorithm either returns an embedding of G in the torus or a subgraph of G which is a subdivision of a minimal nontoroidal graph. The algorithm based on [13] avoids the most complicated step of [13] by applying a recent result of Fiedler, Huneke, Richter, and Robertson [5] about the genus of graphs in the projective plane, and simplifies other steps on the expense of losing linear time complexity.

1 Introduction

There are several efficient (linear time) algorithms for testing planarity of a given graph (Hopcroft and Tarjan [12], Booth and Lueker [2], Fraysseix and Rosenstiehl [8], Shih and Hsu [27]). Some of these algorithms have been implemented and used in diverse applications. It is not difficult to extend original planarity testing algorithms to yield an embedding in the plane if the graph G is planar [3] (see also [24] or [8]). They can also be extended so that they discover a Kuratowski subgraph of G if G is not planar [30, 31]. These algorithms have been implemented, for example, in LEDA [19], but with the detection of Kuratowski subgraphs several troubles have been reported [17]. This led the authors of LEDA to decide to implement the following simple quadratic time algorithm instead [18]: Given a linear time planarity testing algorithm, an *edge-by-edge elimination procedure* can be applied to find a Kuratowski subgraph in quadratic time; given a nonplanar graph G and an edge $e \in E(G)$, test planarity of $G - e$. If $G - e$ is nonplanar, remove e and proceed with the graph $G - e$ since there must be a Kuratowski subgraph avoiding e. Otherwise, e is contained in every Kuratowski subgraph of G. Now, repeat the procedure for the remaining edges of the (reduced) graph. The same approach can also be used for other surfaces.

Embeddings in the torus also have a long history. Filotti [6] presented a polynomial time algorithm for embedding cubic graphs in the torus. Filotti, Miller, and Reif [7] extended that technique to obtain, for an arbitrary surface of genus $g > 0$, a polynomial time algorithm of complexity $\mathcal{O}(n^{cg})$, where c is a constant. Next, Robertson and Seymour proved that each surface has a finite list of forbidden minors [25]. This result implies that for each surface Σ there is a finite list of minimal forbidden subgraphs. (A graph K is a *minimal forbidden subgraph* for Σ if K has no vertices of degree less than 3, cannot be embedded in Σ, and for every edge $e \in E(K)$, $K - e$ can be embedded in Σ.) A graph G cannot be embedded in Σ if and only if it contains a subdivision of a minimal forbidden subgraph for Σ. Kuratowski's theorem states that K_5 and

*Supported in part by the Ministry of Science and Technology of Slovenia, Research Project J1-0502-0101-98.
†e-mail: martin.juvan@fmf.uni-lj.si
‡e-mail: bojan.mohar@uni-lj.si

$K_{3,3}$ are the only minimal forbidden subgraphs for the plane. The only other surface for which the complete list of minimal forbidden subgraphs is known is the projective plane. The list for the projective plane contains 103 graphs [9, 1]. The result of Robertson and Seymour also yields an algorithm of cubic complexity for testing embeddability in any fixed surface since the presence of a minor can be tested in cubic time [26]. Djidjev and Reif [4] improved the algorithm of [7] by presenting a polynomial time algorithm, for each fixed orientable surface, where the degree of the polynomial is fixed, and Mohar [21] recently presented a linear time embedding algorithm for embedding graphs in an arbitrary fixed surface. All these algorithms are more of theoretical than of practical value since it is not clear how to implement them efficiently and their analysis involves very large constants. Let us recall that the general problem of determining the genus [28], or the non-orientable genus [29] of graphs is NP-hard.

In 1994, Juvan, Marinček, and Mohar [13] presented a linear time algorithm for embedding graphs in the torus. That algorithm uses a complicated step which is explained (with a proof of correctness) in [15]. The torus embedding algorithm presented in this paper has a great advantage to bypass that step by using a recent result of Fiedler, Huneke, Richter, and Robertson [5] about the genus of graphs in the projective plane (see Section 5).

Previous polynomial time algorithms for embeddability in the torus have not been implemented. In our present work we explain the main steps of such an algorithm so that it can be implemented in a reasonable amount of time. In order to achieve this, we sacrifice an order or two in the computational complexity which increases from linear to cubic (or even to n^4 if we want to exhibit a minimal forbidden subgraph for the torus by the edge-by-edge elimination). Our motivation was to describe an algorithm which could be implemented and added to the LEDA package [19]. Let us observe that several people seeking for torus embedding algorithms have even considered "practical" exponential algorithms. One such attempt is reported in [23].

2 Basic definitions

The 2-cell embeddings in orientable surfaces can be described combinatorially by specifying the *local rotations*: for each vertex v of the graph G we list its neighbors (or incident edges) in a cyclic order, $\pi_v = (v_1 v_2 \dots v_d)$, which corresponds to the cyclic order of edges emanating from v on the surface (taken in the clockwise direction around v as determined by the specified orientation of the surface). The collection of local rotations $(\pi_v \mid v \in V(G))$ determines the set of *facial walks* (or *faces*) which correspond to the combinatorial boundaries of the faces of the embedding. We refer to [10, 22] for more details.

Embeddings in nonorientable surfaces admit a similar description. Since nonorientable surfaces do not admit a global orientation, a local orientation is chosen in a small neighborhood of each vertex. Each edge $e = uv \in E(G)$ is then associated with a *signature* 1 or -1, depending on whether the chosen local orientations at u and v agree along e or not (cf. [10, 22]). In the algorithms of this paper we shall only use embeddings in the projective plane where an embedding of a subgraph K of G in the projective plane is given by Figure 1. The rest of the graph is embedded in the faces of K, and the subgraph in each of the faces can be described as an (orientable!) embedding in the plane.

The *face-width* $\mathrm{fw}(G, \Pi)$ of a nonplanar embedding Π of G is the minimum integer k such that there is a sequence v_1, F_1, v_2, F_2,…,v_k, F_k of distinct vertices and facial walks of G and there is a noncontractible simple closed curve in the surface of the embedding intersecting G in $v_1, \dots, v_k$ only and passing through the elements of that sequence (so $v_i \in F_{i-1} \cap F_i$, $i = 1, \dots, k$, where indices are taken modulo k).

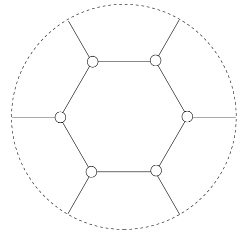

Figure 1: An embedding of $K_{3,3}$ in the projective plane.

Let K be a subgraph of G. A *K-bridge* in G is a subgraph B of G which is either an edge $e \in E(G) \backslash E(K)$ with both endpoints in K, or a connected component of $G - V(K)$ together with all edges (and their endpoints) between this component and K. The vertices of $B \cap K$ are the *vertices of attachment* of B, and B is *attached to* each of these vertices. A vertex of K of degree different from 2 is a *branch vertex* of K. A *branch* of K is a path in K whose endpoints are branch vertices but no internal vertex on this path is a branch vertex. If a K-bridge is attached to a single branch e of K, it is said to be *local* (on e).

A graph G is 3-*connected modulo* K if for every vertex set $X \subset V(G)$ with at most 2 elements, every connected component of $G - X$ contains a branch vertex of K.

3 Basic algorithms

Our algorithm for embedding graphs in the torus uses the subroutines listed below. In all of them, G is a graph and K is a subgraph of G.

2-CONNECTED_COMPONENTS(G): Returns the 2-connected components (blocks) of G. This algorithm can be easily implemented to run in linear time by using a simple extension of DFS.

3-CONNECTED_COMPONENTS(G): Returns the 3-connected components of G (see [11] for definitions and a linear time algorithm; let us remark that it is rather complicated to implement this algorithm correctly).

PLANAR(G): Returns true if and only if G is planar.

PLANAR_EMBED(G): If G is planar, a call to this subroutine returns an embedding of G in the plane.

KURATOWSKI_SUBGRAPH(G): If G is nonplanar, a call to this subroutine returns a subgraph K of G which is a subdivision of K_5 or $K_{3,3}$. If G is 3-connected and distinct from K_5, then K is a subdivision of $K_{3,3}$.

Efficient algorithms for the above tasks have been mentioned in the introduction ([12, 3, 31]). The additional requirement that K is a subdivision of $K_{3,3}$ if $G \neq K_5$ is 3-connected can be achieved in linear time as follows: Suppose that K is a subdivision of K_5. If K has a branch e which is not just an edge, then there is a vertex v in the interior of e and a path P from v to a vertex u of K which does not belong to e such that $P \cap K = \{u, v\}$. The other possibility is that $K = K_5$. Then there are 3 paths P_1, P_2, P_3 joining $V(K)$ and a vertex $v \in V(G) \backslash V(K)$. Now it is easy to see that $K \cup P$ or $K \cup P_1 \cup P_2 \cup P_3$ (respectively) contains a subdivision of $K_{3,3}$.

PROJECTIVE_PLANAR(G): Returns true if and only if G is projective planar. This routine and its derived versions below assume that G is 3-connected.

PROJECTIVE_PLANAR_EMBED(G): Returns an embedding of G in the projective plane if G can be embedded there.

PROJECTIVE_PLANAR_OBSTRUCTION(G): If G cannot be embedded in the projective plane, the subroutine returns a subgraph K of G which is obtained from a minimal forbidden subgraph for the projective plane by subdividing edges and adding at most 2 branches. Moreover, if K can be embedded in the torus, then K is a subdivision of a 3-connected graph. See Section 5.

FACE_WIDTH(G,Π): Given an embedding Π of G in the projective plane, this routine returns the face-width of Π.

EMBEDDING_EXTENSION(G, K, Π): If the embedding Π of K can be extended to an embedding $\tilde{\Pi}$ of G in the same surface, then this routine returns such an extension $\tilde{\Pi}$; otherwise it returns $\emptyset$. Its implementation is needed only in a few special cases described in Section 4.

4 Embedding extension problems

Suppose that a subgraph K of G is Π-embedded in some surface. The *embedding extension problem* asks if it is possible to *extend* Π to an embedding of the whole graph G in the same surface. This problem is rather complicated in general (see [16, 21] for more details) but we need to solve only some special cases.

As a preprocessing to any of the embedding extension problems we first change K so that local K-bridges disappear. Formally, we need G to be 3-connected modulo K. Then we construct a subgraph K' of G which is obtained from K by replacing some branches of K by other branches (so that K' is homeomorphic to K) such that there are no local K'-bridges in G (cf. [14]). A linear time algorithm for this task is given in [20]. Next, we determine all K-bridges in G (by a simple modification of DFS) and determine for every K-bridge in which faces of K it can be embedded (cf. Subsection 4.1).

4.1 Disk embedding extension

Suppose that F is a Π-facial cycle and $\mathcal{B}$ is a set of K-bridges in G whose vertices of attachment are all on F. To check if all bridges of $\mathcal{B}$ can be (simultaneously) embedded in the face F, we form an auxiliary graph $\tilde{K}$ obtained from $F \cup \mathcal{B}$ by adding a new vertex w joined to all vertices in F. Then we just check planarity of $\tilde{K}$. This test is also used separately for every K-bridge to see in which faces of K it can be embedded. With appropriate bookkeeping this can be performed in linear time.

4.2 2-restricted embedding extensions

This type of embedding extension problems occurs when each K-bridge in G is restricted to have at most two *admissible* embeddings in the Π-faces of K. This problem can be reduced to 2-SAT and can be easily solved in time $\mathcal{O}(n^2)$ as discovered by Filotti, Miller, and Reif [7]. (It can also be solved in linear time by more complicated "geometric" approach as shown in [16].)

It may happen that some K-bridges have embeddings in more than two faces. If the number of such bridges and their distinct embeddings is bounded, then we may run the 2-restricted embedding extension algorithm for all combinations of embeddings of these exceptional bridges and hence solve the problem by solving bounded number of 2-restricted embedding extension problems. It can be shown that this approach works, in particular, if Π is a closed 2-cell embedding.

In one of the cases which occur in 4.3 below, we have to solve a 2-restricted embedding extension problem where some bridges have more embeddings (in the same face) but we only consider such extension where bridges use only one or two of these possibilities. Such restrictions are considered to be part of the input for EMBEDDING_EXTENSION(G, K, Π).

4.3 1-singular face embedding extension in the torus

Suppose that G is 3-connected and that K is a subgraph of G such that K is 2-connected, cannot be embedded in the projective plane, and that it contains a Kuratowski subgraph K_0 such that no K_0-bridge in K is local.

Proposition 4.1 *Let G and K be as above. If Π is an embedding of K in the torus, then:*

(a) *All facial walks, except possibly one, are cycles.*

(b) *If F is a facial walk which is not a cycle, then F is of the form $F = AeBe^-$ or $F = AxBx$ where e is a branch of K and x is a branch vertex of K, and A, B are disjoint simple paths in G. If F' is a facial cycle distinct from F which contains an endvertex v of e or contains $v = x$ (respectively), then F' cannot intersect both A and B in vertices distinct from v.*

(c) *If $\mathcal{U}$ is a set of vertices that are not all contained in the same branch of K, then there are at most two facial walks that contain all vertices of $\mathcal{U}$.*

Let G, K, and Π be as in Proposition 4.1. We would like to check if Π can be extended to an embedding of G. To test the extendibility of Π we distinguish four cases and in each of them we transform the embedding extension problem in question to one or two 2-restricted embedding extension problems.

(1) If *all facial walks are cycles*, then the embedding is 2-restricted (by Proposition 4.1(c)), and we apply the results of Subsection 4.2.

Otherwise, *there is a facial walk C which is not a cycle*. Let $C = AeBe^-$ (respectively, $C = AxBx$) be as in Proposition 4.1. Now we distinguish the following three possibilities:

(2) *There is a K-bridge R in G which is attached to (the interiors of) A and B and to a vertex on e (respectively to x)*: Let P be a path in R connecting the interiors of A and B. Embed P in C. The obtained embedding of $K \cup P$ is a closed 2-cell embedding and the results of Subsection 4.2 apply.

(3) *There are K-bridges R_1, R_2 attached to A and B (respectively) and attached to vertices x_1, x_2 on e (respectively) such that x_1 is closer to B on e than x_2*. In this case, let P_i be a path in R_i from A (if $i = 1$) or B (if $i = 2$) to x_i, $i = 1, 2$. Embed P_1 and P_2 in C. Again, the obtained embedding of $K \cup P_1 \cup P_2$ is closed 2-cell and the results of Subsection 4.2 can be used. Note that there are two possibilities how to embed $P_1 \cup P_2$ in C and that we have to test the extendibility of each of them.

(4) *Otherwise*: there is a vertex y on e (or $y = x$) such that no K-bridge in G attached to A has an attachment on e closer to B than to y, and similarly for bridges attached to B and e. Let K' be the graph obtained from K by cutting e at y so that y gives rise to two new vertices y_1, y_2 and the edges of K incident to y become incident either to y_1, y_2 so that the embedding Π gives rise to an embedding of K' in the plane. Similarly we form a graph G' obtained from G by cutting at y. This operation is such that G has an embedding in the torus extending Π if G' is planar. On the other hand, if G' is not planar, then in every extension of Π, there is a path in the face C joining the interiors of A and B. In this case, extendibility of Π can again be checked by the algorithm of Subsection 4.2.

5 Embeddings in the projective plane

In this section we study embeddings of 3-connected graphs in the projective plane. The aim of the projective plane phase of the toroidal algorithm is to find an appropriate subgraph of G that has only “nice” embeddings in the torus (cf. Proposition 4.1). By searching for such a subgraph, it may happen that we find an embedding of the graph in the torus or a minimal forbidden subgraph for the torus, in which case we may stop.

Let G be a 3-connected nonplanar graph. We would like to decide efficiently whether G can be embedded in the projective plane. If yes, we would like to construct an embedding scheme describing such an embedding. If no, we would like to exhibit a subgraph of G homeomorphic to a minimal forbidden subgraph for the projective plane. The algorithm performing this task is based on the embedding extension algorithm of Subsection 4.2.

The algorithm works as follows. Let K_0 be a Kuratowski subgraph of G (homeomorphic to $K_{3,3}$). Note that every embedding of K_0 in the projective plane is a closed 2-cell embedding. Moreover, no K_0-bridge in G can be embedded in more than two faces of K_0. This implies that given an embedding of K_0 in the projective plane, the question whether the given embedding can be extended to an embedding of the whole graph is in fact a 2-restricted embedding extension problem (cf. Subsection 4.2). We therefore try to extend every embedding of K_0 in the projective plane (there are precisely 6 such embeddings) to an embedding of G.

```
Algorithm PROJECTIVE_PLANAR(G)
{ Determine if a nonplanar graph G can be embedded in the projective plane. If it can, the
obtained embeding can be retrieved via PROJECTIVE_PLANAR_EMBED. }
K0 := KURATOWSKI_SUBGRAPH(G);
for every embedding Π of K0 in the projective plane do
    Π' := EMBEDDING_EXTENSION(G, K0, Π);
    if Π' ≠ ∅ then                                   { We have found an embedding. }
        PROJECTIVE_PLANAR_EMBED := Π';
        return true;
return false;                          { G cannot be embedded in the projective plane. }
```

There are two possible outcomes of Algorithm Projective_planar. If the graph is not embeddable in the projective plane, we first apply edge-by-edge elimination procedure to obtain a subgraph K homeomorphic to a minimal forbidden subgraph for the projective plane. If K is not 2-connected, then it is also a minimal forbidden subgraph for the torus. If it is homeomorphic to a 3-connected graph, then it is possible to eliminate local K_0-bridges in K (where K_0 is a Kuratowski subgraph of K). Then the pair G, K satisfies assumptions of Proposition 4.1. Therefore K has only “nice” embeddings in the torus (see Proposition 4.1).

The remaining possibility is that K is 2-connected, but not homeomorphic to a 3-connected graph. Then $K = K_1 \cup K_2$ where $K_1 \cap K_2 = \{x, y\} \subseteq V(G)$ and where K_2 contains all branch vertices of K_0. Change K such that there are no local K-bridges in G. Let P be a path in $G - x - y$ from $K_1 - x - y$ to $K_2 - x - y$. If $K \cup P$ is not a subdivision of a 3-connected graph, then there is a branch e of K_0 such that x, y lie on e. In this case we assume that P lands on e as close as possible to the ends of e. (There may be two candidates for P, one on each side). Finally, there is a path Q joining a vertex of e “covered” by P with a vertex on a different branch of K. Then $K \cup P \cup Q$ is a graph obtained from K by adding one or two branches and either has no embedding in the torus, or it is homeomorphic to a 3-connected graph.

Algorithm Projective_planar_obstruction(G)

{ Returns a subgraph K of G homeomorphic to a minimal forbidden subgraph for the projective plane plus at most 2 branches such that K is either homeomorphic to a 3-connected graph, or it cannot be embedded in the torus. }
$K := G$;
for all $e \in E(G)$ do
    if not PROJECTIVE_PLANAR($K - e$) then
        $K := K - e$;
Extend K as described above;
return K;

The second possible outcome of PROJECTIVE_PLANAR is that we obtain an embedding Π of G in the projective plane. In this case, the face-width of Π determines if G can be embedded in the torus or not.

Theorem 5.1 ([5]) *Let G be a nonplanar graph and Π an embedding of G in the projective plane. Then* $\mathrm{genus}(G) = \lfloor \mathrm{fw}(G, \Pi)/2 \rfloor$.

From Theorem 5.1 we conclude, that a nonplanar graph embedded in the projective plane can be embedded in the torus if and only if the face-width of its projective plane embedding is equal to 2 or 3.

To determine the face-width of Π we first construct its vertex-face incidence graph $G(\Pi)$. This is a bipartite graph whose bipartition classes are the vertex set of G and the faces of Π. A vertex is adjacent to a face if it is lies on the boundary of the face. Note that $G(\Pi)$ is also embedded in the projective plane in a natural way (all faces are quadrangles). It is easy to see that twice the face-width of Π is equal to the *edge-width* of $G(\Pi)$, i.e., the length of a shortest noncontractible cycle in $G(\Pi)$. When embeddings in the projective plane are presented by local rotations and signatures, a closed walk is noncontractible if and only if it contains an odd number of edges with signature -1.

To find the edge-width of an embedded graph, for every vertex $v \in V(G)$ the length of a shortest noncontractible closed walk containing v is determined. This is done by a breadth-first search starting at v. Denote by T_v the obtained tree. For every vertex $u \in V(G)$ we also determine the distance from v and the number of edges with signature -1 on the path between v and u in T_v. It is easy to see that there is always a shortest noncontractible closed walk through v that contains exactly one edge not in T_v. Using the precomputed information it is easy to check which edges from $E(G) \backslash E(T_v)$ determine noncontractible closed walks and to compute the length of a shortest one. The above task can obviously be accomplished in linear time. Algorithm FACE_WIDTH takes the minimum over all vertices $v \in V(G)$ divided by 2 and hence determines the face-width of G in quadratic time. If we are interested only in checking if the face-width of the embedding is at most some constant k (in our case $k = 3$), the time complexity can be reduced to $\mathcal{O}(k \cdot |V(G)|)$ by a slight modification of the algorithm described above.

If the face-width of G is greater than 3, then the edge-by-edge elimination procedure (where we keep the originally obtained embedding Π) gives us a subgraph K of G such that its face-width is 4 and every proper subgraph of K has smaller face-width. By Theorem 5.1 K is a subdivision of a minimal forbidden subgraph for the torus. This subgraph can be found in cubic (or quadratic, if we use the improved face-width computation) time.

On the other hand, if $k := \mathrm{fw}(G, \Pi) \in \{2, 3\}$, let v_1, F_1, v_2, F_2,…,v_k, F_k be a sequence of distinct vertices and faces of G determining the face-width. By changing local rotations and signatures at vertices $v_1, \ldots, v_k$ only one obtains an embedding of G in the torus [5].

6 Embedding graphs in the torus

The main algorithm of this paper either finds an embedding of the given graph G in the torus or finds a subgraph K of G which is a subdivision of a minimal forbidden subgraph for the torus. This algorithm can be implemented so that its time complexity is linear. However, we describe some of its steps with quadratic (or even cubic) time complexity to gain in simplicity and transparency.

By a preprocessing of the given input graph G of order n, we may assume that $|E(G)| \leq 3n+1$ since the graph with more than $3n$ edges cannot be embedded in the torus. (This is a well known consequence of Euler's formula.)

6.1 2-separable embeddings

The algorithm first finds a "suitable" subgraph K of the input graph G and by possibly reducing G ensures that G is 3-connected modulo K (or K itself has no embedding in the torus). This is achieved as follows. First we check for the 2-connectivity of G. There must be exactly 1 nonplanar block in G (if there is none, G is planar, if there are 2 or more, G is nontoroidal). Next we compute the 3-connected components of G. Here, the situation is more involved. The 3-connected components of G are structured in a "tree-like" manner. Let H_1 be a 3-connected component of G that corresponds to a leaf in this tree. This means that $G = H_1' \cup H_2'$ where $H_1' \cap H_2' = \{x, y\} \subseteq V(G)$ is the corresponding 2-separator. Then $H_1 = H_1' + xy$ (where xy is the corresponding *virtual edge*), and let $H_2 = H_2' + xy$. If H_1 is planar, then every embedding of H_2 in the torus can be extended to an embedding of G by replacing e by H_1'. Therefore we may assume that H_1 is not planar. Next we test planarity of H_2. If it is planar, we have reduced the problem of toroidality of G to that of H_1 (which is 3-connected). Otherwise, we distinguish two possibilities. If both H_1' and H_2' are planar, then their embeddings can be combined to an embedding of G in the torus. Otherwise, let K_1 and K_2 be Kuratowski subgraphs in H_1 and H_2, respectively, and let $\tilde{K}_1$ and $\tilde{K}_2$ be the corresponding Kuratowski subgraphs in G in which the possible occurrence of e is replaced by a path in H_2' and H_1', respectively. It can be achieved that $\tilde{K}_1 \cap \tilde{K}_2$ is either empty or contained in a closed branch of $\tilde{K}_1$ (since H_1' or H_2' is nonplanar). Let $K := \tilde{K}_1 \cup \tilde{K}_2$. Then every embedding of K in the torus is a closed 2-cell embedding. It remains to achieve that G is 3-connected modulo K. If K is not 2-connected, then K has no embedding in the torus. Otherwise, we construct an auxiliary graph G^+ by adding to G a new vertex w adjacent to every branch vertex of K. Then there is a 3-connected component H of G^+ containing K. If there exists another nonplanar 3-connected component of G^+, then K plus a Kuratowski subgraph in this component determine a nontoroidal graph. Otherwise, the problem of toroidality of G is reduced to the same problem for $G' := H - w$ (which is 3-connected modulo K). Finally, to find an embedding of G' in the torus, we repeat, for every embedding of K in the torus, the 2-restricted embedding extension algorithm of Subsection 4.2. If there is no embedding in the torus, we find a minimal forbidden subgraph by the edge-by-edge elimination. If there was more than one nonplanar 3-connected component of G, then this whole process (which we call 2-SEPARABLE_EMBED) either finds an embedding of G in the torus or finds a minimal forbidden subgraph. Note that each of these two outcomes can occur at two different stages.

6.2 The algorithm

Subsection 6.1 describes reduction to 3-connected case. Having a nonplanar 3-connected graph, the results of Section 5 are applied. They either give an embedding, a minimal forbidden subgraph for the torus, or a subgraph K which satisfies assumptions of Proposition 4.1. Embedding extensions of K can be determined by applying results of Section 4.

Algorithm TORUS_EMBED(G)
{ This algorithm returns an embedding Π of genus ≤ 1 or a subgraph $\tilde{K}$ of G that is a minimal forbidden subgraph for the torus. }
{ G is a simple graph of order n with at most $3n+1$ edges }
Let $\{G_1, \ldots, G_p\}$ be the 2-CONNECTED_COMPONENTS(G);
$s := 0$;
for $i := 1$ to p do if not PLANAR(G_i) then
    $s := s + 1$; $t := i$; $H_s :=$ KURATOWSKI_SUBGRAPH(G_i);
if $s = 0$ then $\Pi :=$ PLANAR_EMBED(G); return(Π);
if $s \geq 2$ then $\tilde{K} := H_1 \cup H_2$; return($\tilde{K}$);
{ Otherwise $s = 1$ }
Let $Q := G_t$ be the nonplanar block of G.
Let $\{Q_1, \ldots, Q_r\}$ be the 3-CONNECTED_COMPONENTS(Q);
if more than one graph Q_i ($i = 1, \ldots, r$) is nonplanar then 2-SEPARABLE_EMBED(G);
{ Otherwise precisely one Q_i is nonplanar. }
Let Q' be the nonplanar 3-connected component;
if PROJECTIVE_PLANAR(Q') then
    $\Pi' :=$ PROJECTIVE_PLANAR_EMBED(Q');
    if FACE_WIDTH(Q', Π') ≥ 4 then
        $\tilde{K} :=$ minimal subgraph of Q' of face-width ≥ 4; return($\tilde{K}$);
    else Π' determines an embedding of Q' in the torus;
else { Q' is not projective planar }
    $K :=$ PROJECTIVE_PLANAR_OBSTRUCTION(Q');
    for every embedding Π of K in the torus do
        $\Pi' :=$ EMBEDDING_EXTENSION(G, K, Π);
        if $\Pi' \neq \emptyset$ then break;
    if $\Pi' = \emptyset$ then { Q' is nontoroidal }
        Let $\tilde{K}$ be a minimal forbidden subgraph for the torus in Q';
        return($\tilde{K}$);
Combine Π' with planar embeddings of $\{Q_1, \ldots, Q_r\} \backslash \{Q'\}$ and $\{G_1, \ldots, G_p\} \backslash \{G_t\}$ to an embedding Π of G in the torus;
return(Π);

References

[1] D. Archdeacon, *A Kuratowski theorem for the projective plane*, J. Graph Theory 5 (1981) 243–246.

[2] K. S. Booth, 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.

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

[4] H. Djidjev, J. H. Reif, *An efficient algorithm for the genus problem with explicit construction of forbidden subgraphs*, 23rd Ann. ACM STOC, New Orleans, LA, 1991, pp. 337–347.

[5] J. R. Fiedler, J. P. Huneke, R. B. Richter, N. Robertson, *Computing the orientable genus of projective graphs*, J. Graph Theory 20 (1995) 297–308.

[6] I. S. Filotti, *An algorithm for imbedding cubic graphs in the torus*, J. Comput. System Sci. 20 (1980) 255–276.

[7] I. S. Filotti, G. L. Miller, J. Reif, *On determining the genus of a graph in $\mathcal{O}(v^{\mathcal{O}(g)})$ steps*, 11th Ann. ACM STOC, Atlanta, GA, 1979, pp. 27–37.

[8] H. de Fraysseix, P. Rosenstiehl, *A depth-first search characterization of planarity*, Ann. Discrete Math. 13 (1982) 75–80.

[9] H. Glover, J. P. Huneke, C. S. Wang, *103 graphs that are irreducible for the projective plane*, J. Combin. Theory Ser. B 27 (1979) 332–370.

[10] J. L. Gross, T. W. Tucker, *Topological Graph Theory*, Wiley-Interscience, New York, 1987.

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

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

[13] M. Juvan, J. Marinček, B. Mohar, *Embedding graphs in the torus in linear time*, Lect. Notes Comput. Sci. 920 (1995) 360–363.

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

[15] M. Juvan, J. Marinček, B. Mohar, *Obstructions for simple embeddings*, submitted.

[16] M. Juvan, B. Mohar, 2*-restricted extensions of partial embeddings of graphs*, submitted.

[17] K. Mehlhorn, Private communication.

[18] K. Mehlhorn, P. Mutzel, S. Näher, *An implementation of the Hopcroft and Tarjan planarity test and embedding algorithm*, Technical Report Max-Planck-Institut für Informatik, No. MPI-I-93-151, 1993.

[19] K. Mehlhorn, S. Näher, *Leda: A Platform for Combinatorial and Geometric Computing*, Cambridge University Press, 1999.

[20] B. Mishra, R. E. Tarjan, *A linear-time algorithm for finding an ambitus*, Algorithmica 7 (1992) 521–554.

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

[22] B. Mohar, C. Thomassen, *Graphs on Surfaces*, Johns Hopkins University Press, to appear.

[23] E. Neufeld, W. Myrvold, *Practical toroidality testing*, Proc. 8th Ann. ACM-SIAM SODA, New Orleans, LA, 1997, pp. 574–580.

[24] T. Nishizeki, N. Chiba, *Planar graphs: theory and algorithms*, Annals of Discrete Mathematics, 32, North-Holland Publishing Co., Amsterdam-New York, 1988.

[25] N. Robertson, P. D. Seymour, *Graph minors. VIII. A Kuratowski theorem for general surfaces*, J. Combin. Theory Ser. B 48 (1990) 227–254.

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

[27] W. K. Shih, W. L. Hsu, *A simple test for planar graphs*, manuscript.

[28] C. Thomassen, *The graph genus problem is NP-complete*, J. Algorithms 10 (1989) 568–576.

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

[30] S. G. Williamson, *Embedding graphs in the plane – algorithmic aspects*, Ann. Discrete Math. 6 (1980) 349–384.

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