JOURNAL OF ALGORITHMS **15**, 482–502 (1993)

Projective Planarity in Linear Time

BOJAN MOHAR*

Department of Mathematics, University of Ljubljana, Jadranska 19, 61111 Ljubljana, Slovenia

Received April 1992; accepted February 1993

A linear time algorithm for embedding graphs in the projective plane is presented.

1. INTRODUCTION

The problem of constructing embeddings of graphs in surfaces is of practical and of theoretical interest. The practical issues arise, for example, in problems concerning VLSI and also in several other applications, since graphs embedded in low genus surfaces can be handled more easily. Theoretical interest comes from the fact that the genus parameter of a graph is an important graph invariant and that graphs of bounded genus have interesting properties.

There are very efficient (linear time) algorithms which for a given graph determine whether the graph can be embedded in the 2-sphere. The first such algorithms were obtained by Hopcroft and Tarjan [11] back in 1974. There are several other linear time planarity algorithms (Booth and Lueker [3], Fraysseix and Rosenstiehl [8], Williamson [24, 25]). The extensions of original algorithms produce also an embedding (rotation system) whenever a graph is found to be planar [4], or find a forbidden Kuratowski subgraph K_5 or $K_{3,3}$ if a graph is found to be non-planar [24, 25] (see also [14]).

It is known [21] that the general problem of determining the genus, or the non-orientable genus of graphs is NP-hard. However, for every fixed surface there is a polynomial time algorithm which checks if a given graph

*Supported in part by the Ministry of Research and Technology of Slovenia, Research Project P1-0210-101-92.

0196-6774/93 $5.00

can be embedded in the surface. Such algorithms were found first by Filotti *et al.* [7]. For a fixed orientable surface S of genus g they discovered an algorithm with time complexity $O(n^{\alpha g+\beta})$ (α, β are constants) which tests if a given graph of order n can be embedded in S. Unfortunately, their algorithm is practically not useful, even in the simplest case when S is the torus. For every fixed surface an $O(n^3)$ algorithm can be devised using graph minors [16, 20]. Robertson and Seymour recently improved their $O(n^3)$ algorithms to $O(n^2 \log n)$ [17–19]. A constructive version is described by Archdeacon in [1]. The running time is proportional to n^{10} but with a little additional care it could be decreased to $O(n^6)$. Unfortunately, also these algorithms are only of theoretical interest. A disadvantage is that the list of forbidden minors is unknown for any surface different from the two-sphere and the projective plane. Even for the projective plane whose forbidden minors are known [2, 9], the algorithms based on checking for the presence of forbidden minors are consuming too much time, since their running time estimates involve enormous constants.

The main motivation for designing a fast projective plane algorithm is our goal to obtain a fast (linear time and acceptable constants) algorithms for checking embeddability of graphs in any fixed surface. The projective plane algorithm of this paper and the torus algorithm of Juvan, Marinček, and the author [12] are the building blocks used in the general case [13].

The projective plane is the simplest (compact) surface after the two-sphere. No practically applicable algorithms for embedding graphs in the projective plane have appeared in print so far. Recently J. Fiedler *et al.* found that the orientable genus testing for graphs which can be embedded in the projective plane is a polynomial time problem [6]. This result opened the theoretical interest (in addition to the possible applications in practice) in algorithms for testing if a given graph can be embedded in the projective plane. In [15] an algorithm for constructing embeddings into the projective plane is described, and it is claimed that it works in time proportional to n^3. Unfortunately, the algorithm of [15] seems to have a gap. If an embedding is found, then all is correct, but if the algorithm fails, it does not mean that the graph cannot be embedded in the projective plane. We see a way how to fix the gaps, but then the worst-case time complexity becomes exponential. (The unavoidable mistakes in [15] were independently observed by Williamson (private communication).)

In the present paper we describe a new algorithm which finds an embedding of a given graph into the projective plane if such an embedding exists. Its worst-case time and space complexity is linear. The constants involved in the linear time complexity bound are acceptably small from the practical point of view, and we are working towards an implementation of our algorithm.

Concerning the time complexity of our algorithm, we assume a random-access machine (RAM) model, the so-called *unit-cost* RAM, where operations on integers, whose value is $O(n)$, need only constant time (n is the order of the given graph). This model was introduced by Cook and Reckhow [5].

At the end of this section let us formulate our main result. The rest of the paper is devoted to its proof.

THEOREM 1.1. *There is an algorithm which for a given simple graph G of order n determines in time $O(n)$ if G has an embedding into the projective plane. In case of the positive answer it also finds an embedding.*

It should be mentioned that the number of edges of a simple graph of order n which is embeddable in the projective plane is at most $3n$. This follows easily from the Euler's formula. If the given graph has more than $3n$ edges, then it is either not simple, or it is not embeddable in the projective plane.

Our embedding algorithm uses extensively a linear time planarity algorithm. For this purpose one may take any one of such algorithms. In the initial step we also need to exhibit a Kuratowski subgraph if the graph is not planar but at any later stage we do not need this. In case we only want to check if a graph is embeddable in the projective plane, we only need to check planarity without actually constructing planar embeddings.

Embeddings in the projective plane can be combinatorially described by specifying a *rotation system* (for each vertex v of G we have the cyclic permutation π_v of its neighbours, representing their circular order around v on the surface), together with a *signature* λ: $E(G) \to \{-1, 1\}$ having the property that a cycle of G has an even number of edges e with $\lambda(e) = -1$ if and only if the cycle is contractible on the projective plane (cf. [10]). For embeddings in orientable surfaces, in particular the two-sphere, we only need to give the rotation system (we may assume that the signature is everywhere equal to one).

COROLLARY 1.2. *There is an $O(n^2)$ algorithm for determining the orientable genus of graphs embeddable in the projective plane.*

Proof. Let G be a given projective planar graph. We may assume that G is simple, since omitting loops and replacing multiple edges by simple edges does not change the genus of the graph. Check first if G is planar. If yes, we are done. Otherwise find an embedding of G in the projective plane and from now on consider G as an embedded graph.

A family of faces of the graph G embedded in the projective plane is *essential* if the union of the closures of faces in this family contains a non-contractible curve. The smallest number of faces forming an essential family is called the *face-width* of the embedded graph G. We denote it by

$\rho(G)$. The face-width can be determined in the following way. Construct first the vertex-face incidence graph H (also called the angle graph, or the radial graph). Its vertices are the union of $V(G)$ and the set of faces of G. For each incident vertex-face pair (v, F) we have an edge between v and F in H. Moreover, H is embedded in the projective plane in the obvious way. The graph H can easily be determined in $O(n)$ time using the "face tracking" algorithm (see [10, p. 115]). One can at the same time determine the rotation system and the corresponding signature λ_H of H. It is easy to see that the length of a shortest non-contractible cycle of H is equal to $2\rho(G)$. Therefore, to compute $\rho(G)$ it suffices to determine a shortest essential cycle of H. An efficient algorithm for this problem was obtained by Thomassen [22]. It consists of the following simple procedure. For each vertex $u \in V(H)$ perform the breadth-first search (BFS) starting at u. When a new edge $e = v_1v_2$ is encountered at the BFS, it either enters the BFS tree T, or else it determines a cycle C, together with the edges that are already in T. If this cycle is non-contractible and contains u then this is a shortest non-contractible cycle of H containing u. If C is non-contractible but $u \notin V(C)$, then u does not lie on a shortest non-contractible cycle of G. Let $\eta(v)$ denote the parity of the number of edges having the signature -1 on the path in T from the root u to v. This is easily constructed at the time of the BFS. Then the number of edges on C with negative signature has the same parity as $\eta(v_1) + \eta(v_2)$ ($+1$ if $\lambda_H(e) = -1$). This determines (in constant time) whether C is contractible or not. Repeating the BFS with each vertex as the root we find a shortest non-contractible cycle. The worst-case time complexity is linear for each breadth-first search, so the complete procedure for determining $\rho(G)$ is quadratic in n.

Compute now $g = \lfloor \rho(G)/2 \rfloor$. Since we already excluded the possibility that G is planar, g is equal to the orientable genus of the graph G [6].

If necessary, one can also obtain an embedding of G into the orientable surface of genus g using the constructed minimal non-contractible cycle of H. See [6] for details. □

2. Basic Definitions

We consider finite undirected graphs. Our concern is testing for embeddability into a given surface. Therefore it may be assumed that all graphs are simple. Throughout the paper, G will always be a simple graph of order n.

We will mainly consider graphs embedded in surfaces. All surfaces are assumed to be closed and without boundary. By $\tilde{\Sigma}_1$ we will denote the projective plane. If G is a graph embedded in a surface S, the connected components of $S \setminus G$ are called *faces* of G in S. If every face is homeomorphic to an open disk in the plane then the embedding is said to be a *2-cell embedding*. The boundary of every face of a 2-cell embedded graph G determines a closed walk consisting of the sequence of edges of G following the boundary of the face. This closed walk is called a *facial walk* of the face. (It is determined up to the choice of the initial point and the direction.) If every facial walk is a cycle of the graph then the embedding is said to be *closed*.

Let K be a subgraph of G. A *relative K-component* is a subgraph of G which is either an edge $e \in E(G) \setminus E(K)$ (together with its endpoints) which has both endpoints in K, or it is a connected component of $G - V(K)$ together with all edges (and their endpoints) between this component and K. Sometimes we will use just *K-component*, or also a *bridge* of K, to mean a relative K-component. Each edge of a K-component R having an endpoint in K is a *foot* of R. The vertices of $R \cap K$ are the *vertices of attachment* of R.

Consider a subgraph K of G. A vertex whose degree in K is different from two is called a *main vertex* of K. The paths in K joining pairs of main vertices and having all internal vertices of degree 2 in K, are called *branches* of K. If a relative K-component is attached at a single branch of K it is said to be *local*. Those relative K-components which are not local are called *global*.

Let K be a subgraph of G and C a cycle in K. Two relative K-components B_1, B_2 *overlap* on C if on C they have three vertices of attachment to K in common, or there are distinct vertices a, b, c, d on C (in this order) such that a and c are vertices of attachment of B_1, and b, d are attachments of B_2. If K is two-cell embedded in a surface S and F is a face bounded by a cycle C then we also say that B_1, B_2 *overlap* in the face F if they overlap on C. The following proposition is well known.

PROPOSITION 2.1. *Let K be a subgraph of G. Suppose that K is two-cell embedded in a surface S and that a cycle C of K bounds a face F of K. If $B_1, B_2, \ldots, B_t$ are relative K-components and each of them admits an embedding in F, then their union $B_1 \cup \cdots \cup B_t$ has an embedding in F if and only if no two of the bridges B_i, B_j, $i \neq j$, overlap on C.*

A related result, which we shall use without explicitly referring to it, is the following proposition.

PROPOSITION 2.2. *Let K be a subgraph of G. Suppose that K is two-cell embedded in a surface S and that a cycle C of K bounds a face F of K. Let B be a relative K-component and ψ_1, ψ_2 extensions of the embedding of K to an embedding of $K \cup B$ such that B is embedded in F under each of them. Then*

ψ_1 can be extended to an embedding of G in S if and only if ψ_2 can be extended to an embedding of G.

3. The Graphs K_5 and $K_{3,3}$ in the Projective Plane

We will use some properties of the embeddings of Kuratowski graphs K_5 and $K_{3,3}$ in the projective plane. Let us first recall a lemma.

Lemma 3.1. *Let G be a 2-connected graph embedded in the projective plane. If G is not planar then this is a closed two-cell embedding.*

Proof. Clearly an embedding which is not two-cell gives rise to an embedding of G in the two-sphere. Suppose that the embedding is not closed. Then there is a face F whose facial walk is the union of a cycle C and a closed walk W, such that C and W have only a vertex, say x, in common. Let γ be a simple closed curve on $\tilde{\Sigma}_1$ which follows C. Then γ can be moved by a homotopy in the face F so that the obtained curve γ' intersects the graph only in x. Since G is 2-connected, it is easy to see that γ' is an essential curve on $\tilde{\Sigma}_1$. Cutting along γ' we obtain a disk in the plane with the double occurrence of x on its boundary. It is clear that this gives rise to an embedding of G in the plane. □

Theorem 3.2. *Up to automorphisms of K_5, the equivalence classes of embeddings of K_5 in the projective plane are collected in Figs.* 1a *and* b.

Proof. By Lemma 3.1 every embedding of K_5 in the projective plane is a closed two-cell embedding. Therefore every face is of size at most five. It follows by the Euler's formula and a standard counting argument that K_5

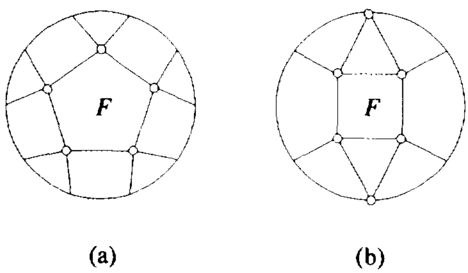

Fig. 1. Embeddings of K_5.

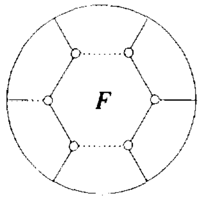

FIG. 2. The embedding of $K_{3,3}$ in the projective plane.

has no triangular embedding into $\tilde{\Sigma}_1$. So there is a face of size five or four. It is easy to see that in each among these two possibilities we have a unique embedding of K_5 as shown in Fig. 1a and b, respectively. □

THEOREM 3.3. *Up to automorphisms of $K_{3,3}$ the only equivalence class of embeddings of $K_{3,3}$ in the projective plane is represented in Fig.* 2.

Proof. As in the proof of Theorem 3.2 we find that every embedding is a closed two-cell embedding and that there is a face of size five or six. Since $K_{3,3}$ is bipartite, it must be of size six. Then it is easy to see that the remaining three edges can be added in a unique way as shown in Fig. 2. □

Remark 3.4. It is easy to check that the embedding of K_5 of Fig. 1a gives rise to 12 non-equivalent embeddings of K_5 (as a labelled graph). Similarly, Fig. 1b represents 15 embeddings of K_5, and the embedding in Fig. 2 represents six non-equivalent embeddings.

COROLLARY 3.5. *Let G be a graph, K a subgraph of G homeomorphic either to K_5 or $K_{3,3}$ and let B be a global bridge of K which is attached to at least one non-main vertex of K. If K is embedded in the projective plane such that B can be added to it in two different faces, then one of the faces is the face F as displayed in Figs.* 1a, 1b, *or* 2.

Proof. Consider one of the possible embeddings of K. Since B is attached to an interior vertex of a branch, say e, of K, the two faces in which B can be embedded share e on their boundary. But since B is global, they must have in common a vertex which does not lie on e. By checking all the possibilities we see that in each case one of the faces is the face F as displayed in Figs. 1a, 1b, and 2, respectively. □

4. Elimination of Local Bridges

In this section, G will be a given graph and K its subgraph. G is *3-connected modulo* K if for every vertex set $X \subset V(G)$ with at most two elements, every connected component of $G - X$ contains a main vertex of K. This is obviously equivalent to the following condition: If $G^+(K)$ is the graph obtained from G by adding three mutually adjacent new vertices whose additional neighbours are the main vertices of K, then $G^+(K)$ is 3-connected. In this paper we will use the notion of 3-connected modulo K only in cases when K will be homeomorphic to a 3-connected graph. In this case the condition is equivalent to the requirement that G is 3-connected.

In our final algorithm K will be a Kuratowski subgraph of G embedded in the projective plane. The following result shows that we will be able to reduce the problem to the case when G is 3-connected modulo K.

We need some additional definitions. Let G be a 2-connected graph. A pair of vertices x, y of G is a *separation pair* if there exist edge-disjoint subgraphs G_1, G_2 of G whose union is G, their intersection is $\{x, y\}$, and each of them contains at least two edges. If x, y is a separation pair and G_1, G_2 the corresponding graphs, then the graphs $G_1' = G_1 + xy$ (G_1 with a new edge xy) and $G_2' = G_2 + xy$ are called *split graphs* of G. The new edges xy added to G_1 and G_2 are called *virtual edges*. Note that by adding virtual edges we may obtain new parallel edges. Dividing G into two split graphs G_1' and G_2' is called *splitting*. Reassembling the two split graphs over a common virtual edge is called *merging*. Suppose now that we split the graph G, then split the obtained split graphs, and so on, until no further splitting is possible. The obtained graphs are called *split components* of G. If we merge all triple bonds (over the same virtual edges, of course) into larger bonds, and merge triangles into larger cycles, as much as possible, we obtain graphs which are called *3-connected components* of G. The split components are not necessarily unique, but the 3-connected components of G are [23]. The 3-connected components are either 3-connected graphs, bonds, or cycles. Every 3-connected component is homeomorphic to a subgraph of G.

Proposition 4.1. *Let K be a subgraph of a graph G. If K is closed 2-cell embedded in a surface S then we can in time proportional to $|E(G)|$ fulfil one of the following tasks*:

(a) *Change G to a graph G' which is homeomorphic to a subgraph of G, and change K to a subgraph K' of G' which is homeomorphic to K, has the same main vertices, and is embedded in S the same way as K (with the main vertices fixed). Moreover, G' is 3-connected modulo K', and the* *embedding of K' can be extended to an embedding of G' if and only if the embedding of K can be extended to an embedding of G. Every extension of the embedding of K' to an embedding of G' gives rise in linear time to an extension of the embedding of K to an embedding of G.*

(b) *Prove that the embedding of K cannot be extended to an embedding of G.*

Proof. Let $G^+ = G^+(K)$ be the graph as defined above. By the algorithm of Hopcroft and Tarjan [11] one can determine the 3-connected components of G^+ in linear time (this algorithm also takes care of the case when G^+ is not two-connected). Let G_1 be the 3-connected component of G^+ which contains the three vertices added to G. Then the graph G' obtained from G_1 by deleting the three new vertices is homeomorphic to a subgraph G_2 of G which contains all the main vertices of K. It is easy to see that we may choose G_2 so that K is its subgraph. Since K is closed two-cell embedded in S, every split component (except G_1) is planar if the embedding of K can be extended to an embedding of G. (So we either prove (b), or have plane embeddings of the split components different from G_1.) The same holds for 3-connected components of G^+. It follows that the embedding of the subgraph K' of G' corresponding to K has an extension to an embedding of G' if and only if the embedding of K has an extension to an embedding of G.

It remains to show how one can extend in linear time an embedding of G' to obtain an embedding of G in the projective plane. G is obtained from G' by a sequence of mergings. We can obtain plane embeddings of all the 3-connected components Q_i (except G_1) of G^+ in time $O(|V(Q_i)|)$, so all together in time $O(n)$. It remains to show how we can obtain embeddings of graphs obtained by merging a graph embedded in the projective plane with a graph embedded in the 2-sphere. Suppose we have a closed 2-cell embedding of a graph H_1 in the projective plane given by the rotation system π_v, $v \in V(H_1)$, and the signature λ_1. Let the planar embedding of a 3-connected component H_2 of G be given by a rotation system on H_2, and let H be the graph obtained from H_1 and H_2 by merging along the virtual edge $e = uv \in E(H_1) \cap E(H_2)$. We need to describe the rotation system and the signature of H which correspond to the embedding of H in the projective plane. The time needed for this must be $O(|V(H_2)|)$. But this is not difficult. We leave it to the reader to show that the following procedure does the job. The rotations of H at vertices in $H_i \setminus \{u, v\}$, $i = 1, 2$, are equal to the rotations in H_i. The rotation at u is obtained in such a way that in π_u the edge e is replaced by the sequence of edges in H_2 following e in the rotation of H_2. We do the same at v if $\lambda_1(e) = 1$. In the case when $\lambda_1(e) = -1$ we replace e in π_v by the sequence of edges adjacent to v in H_2 (except e), but this time in

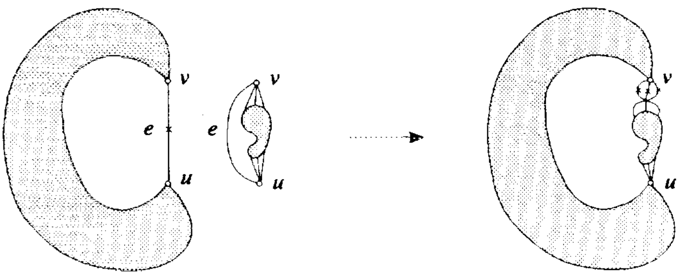

FIG. 3. Merging embeddings in case when $\lambda(\mathbf{e}) = -1$.

the reverse order. See Fig. 3 (the crosses on edges designate the negative signature). Finally, to obtain the signature λ of H we use the signature of H_1 on $E(H_1) \setminus \{e\}$, define the edges of $E(H_2) \setminus F_v$, where $F_v = \{f \in E(H_2) | f$ is adjacent to v in $H_2\}$, to have the signature one, and the edges in $F_v \setminus \{e\}$ to have the signature $\lambda_1(e)$. □

Once having the graph 3-connected modulo its Kuratowski subgraph K we will need to eliminate local bridges of K. This is established by the following result.

PROPOSITION 4.2. *Let K be a subgraph of a graph G such that G is 3-connected modulo K. If K is closed 2-cell embedded in a surface S then we can in time proportional to $|E(G)|$ fulfil one of the following:*

(a) *Change K to a subgraph K' of G which is homeomorphic to K, has the same main vertices, and is embedded in S the same way as K. In particular, G is also 3-connected modulo K'. Moreover, there are no local bridges of K'.*

(b) *Prove that the embedding of K cannot be extended to an embedding of G.*

Proof. Given K, we first determine all local bridges of K. For each branch e of K, define the graph H_e as the union of e, together with all local bridges attached at e, and with the additional edge e' joining the endvertices of e. Find an embedding of H_e in the plane. (If H_e is not planar then the embedding of K cannot be extended to an embedding of G since the embedding of K is closed.) Clearly, H_e is 2-connected. Therefore a face F containing e' is bounded by a cycle consisting of e' and a path P_e. Replace in K the branch e by the path P_e. Denote by K'' the graph obtained by performing this operation at all branches of K. Note that K'' may still have local bridges but it is easy to see that any global bridge of K is contained in a global bridge of K''. Repeat now the above procedure from the beginning with the graph K''. If H_e'' is the auxiliary graph used in the process, note that there are two possibilities to choose the path P_e''. If we take the same embedding as before (since P_e'' is contained in the corresponding graph used in the first step) then in one of the cases, P_e'' is equal to e. Make sure to always take the other one. Since G is 3-connected modulo K, it is also 3-connected modulo K''. Therefore by changing the branch e of K'' to P_e'' we eliminate the local bridges attached at this branch.

Let K' be the resulting subgraph of G. It follows by the above that G' has no local bridges. All other required properties follow by construction.

It is well known that the algorithms used in the above construction of K' (finding all local bridges, planarity) are linear and since the graphs H_e are pairwise edge-disjoint, the total time is also linear in $|E(G)|$. □

5. Testing for Planarity

Planarity testing and embedding algorithms will be used in several situations. We will describe three examples. Whenever referring to one of these cases we will say that we are *testing for planarity*.

Let K be a subgraph of a graph G and suppose that K is 2-cell embedded in a surface S.

Example 1. Let F be a face of K whose boundary is a cycle C. Let $B_1, \ldots, B_k$ be bridges of K with all their vertices of attachment on C. Let the graph Q be the union $C \cup B_1 \cup \cdots \cup B_k$, together with an additional vertex z adjacent to all vertices of attachment of the bridges B_i ($i = 1, \ldots, k$) on C. It is easy to see that Q is planar if and only if the bridges $B_1, \ldots, B_k$ can be simultaneously embedded in the face F. Therefore, by *testing for planarity* of Q we can determine if the bridges $B_1, \ldots, B_k$ can be simultaneously embedded in F. In particular, we can check with this procedure if a single bridge can be embedded in a (closed) face (using just one bridge, i.e., $k = 1$). See also Example 2. If each B_i ($i = 1, \ldots, k$) admits an embedding in F then we can also test for their overlapping on C by using the above procedure.

Example 2. Using the above idea we can test in linear time $O(n)$ which bridges of K can be embedded in F. The crucial idea is that we can construct all the graphs Q_i (cf. Example 1) corresponding to the bridges B_i of K ($i = 1, \ldots, k$) simultaneously and spend only $O(n)$ time. In Q_i we only need those vertices of C which are attachments of the bridge B_i. Going around the boundary C of F we have to remember for each bridge

B_i which was the last vertex of attachment of B_i that we have met. When we meet another foot of B_i we can then immediately join the last vertex with the new one. The time required for these operations is linear in $\sum_{i=1}^{k} |E(B_i)| + |V(C)|$. It follows that the time is also linear in n.

EXAMPLE 3. Let F, F' be faces of K bounded by cycles C, C', respectively. Suppose that $C \cap C' = e \cup \{x\}$, where e is a branch of K and x is a vertex. If a and b are the endpoints of e then denote by P_1, P_1' the segments on C and C', respectively, between a and x. Similarly, let P_2, P_2' be the segments from b to x. We assume that the segments are chosen in such a way that they do not contain e. Let $B_1, \ldots, B_k$ be bridges of K with all their vertices of attachment on $C \cup C'$. Let the graph Q be the union $C \cup C' \cup B_1 \cup \cdots \cup B_k$, together with two additional vertices z_1, z_2 such that z_1 is adjacent to a, x, and all vertices of attachment of the bridges B_i $(i = 1, \ldots, k)$ on $P_1 \cup P_1'$, and z_2 is adjacent to b, x and the attachments on $P_2 \cup P_2'$. It is easy to see that Q is planar if and only if the bridges $B_1, \ldots, B_k$ can be simultaneously embedded in $F \cup F'$. Again, the time complexity is linear in n.

6. SKEW-PLANARITY ALGORITHM

In this section we basically describe an algorithm for embedding graphs in the Möbius band. More precisely, suppose that we have a graph G of order n and a subgraph K of G homeomorphic to K_4 and embedded in the Möbius band as shown in Fig. 4. (Note that four of the edges are embedded on the boundary, while the edges e and e' join points on the boundary but lie otherwise in the interior.) Suppose that K has no local bridges and that G is 3-connected modulo K. Denote by F and F' the faces of K, and let e, e' be the branches on the intersection of their

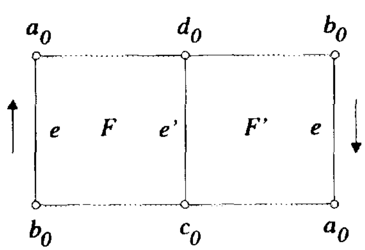

FIG. 4. $\mathbf{K}_4$ in the Möbius band.

boundaries. Our aim is to find out in linear time whether the embedding of K can be extended to an embedding of G.

THEOREM 6.1. *Under the notation and assumptions given above, there is a linear time algorithm which finds an embedding extension to G of the given embedding of K in the Möbius band if such an extension exists, and otherwise reports that no such an extension exists.*

Proof. Denote by a_0, b_0 the endvertices of e (with a_0 on the top), and let c_0, d_0 be the endvertices of e' (with d_0 on the top). See Fig. 4. The algorithm is as follows:

Step 1. For each bridge of K find out in which of the faces it can be embedded. If a bridge B_i contains n_i vertices then the time used by B_i for this operation is $O(n_i)$ with $O(n)$ total pre-processing time (cf. Section 5, Example 2). All together, the time is linear in n. Split the bridges into three groups: those embeddable in F but not in F', those embeddable only in F', and those which can be embedded in both F and F'. If there is a bridge which has no embedding in F or F', then G has no embedding extending the embedding of K. Try to embed all the bridges of the first type in F using a planarity algorithm. Do the same in F' with bridges which have no embedding in F. If we are not successful in one or the other case, then there is no embedding of G extending the given embedding of K. For the remaining bridges which have two embeddings with respect to K, we use the further steps described below to find out if all of them can be embedded at the same time.

Mark as *labelled* all bridges that have been determined in which face they will be embedded and consider the remaining bridges as *unlabelled*.

Step 2. Set $a := a_0$, $b := b_0$, $c := c_0$, and $d := d_0$. In the general step of our algorithm we will have the same situation as at this point. There are vertices a, b on e, where a is above b (with respect to the directions determined by Fig. 4), and we have vertices c, d on e' with d on the top. We assume that ∂F is oriented such that the order of the vertices is a, b, c, d. Similarly, $\partial F'$ is oriented so that their order is a, b, d, c. We will from time to time use *segments* on the boundaries of these faces. Then the segment xy will mean the segment from x to y in the direction agreed above (in the face F if not mentioned otherwise). Some of the bridges of K are already decided to be embedded in F or in F', respectively. The decision has been made exactly for all those bridges that are labelled. We will actually embed them in Step 6. Therefore we do not need to check now that the bridges determined to be embedded in a particular face do not overlap. All the remaining bridges have all their attachments to e

between a and b (possibly at a, or b), and all attachments to e' are in the segment cd.

Among all labelled bridges that are assigned to F and have a vertex of attachment in the open segment da on ∂F, determine their attachment x_a on the segment ac_0 which is as close to c_0 as possible, but different from c. (Now this means that $x_a \neq c_0$ but later on when c will be changed, the same definition will permit the possibility $x_a = c_0$. This also means that when c will change its value from c_0 to another one, we may need to update the value x_a.) If there is no appropriate bridge, then we set $x_a := a$. (We leave it to the reader to verify how one can determine x_a in linear time with a certain pre-processing in Step 1, so that also the later updating of x_a, when the vertices a, b, c, d will change, will not take more than linear time all over the whole algorithm.) The significance of x_a lies in the fact that each unlabelled bridge B attached between a and x_a (possibly at a but not at x_a) overlaps in F with a labelled bridge assigned to F, and B should therefore be embedded in F'. We note that for such a bridge B we will not check its overlapping with other bridges in F', since this will come out in Step 6 anyway.

Define similarly x_b, x_c, x_d, and introduce the corresponding vertices x'_a, x'_b, x'_c, x'_d by considering the bridges that have been assigned to F'.

Step 3. We will (possibly) return to the beginning of Step 3 several times but we note that each time we either embed a new bridge or move a, b or c, d closer towards each other. Therefore we do this only $O(n)$ times.

If $a = b$ or $c = d$ then go to Step 5.

If $a \neq x_a$ then for all unlabelled bridges B attached at a repeat the following. Assign B to be embedded in F' and mark it labelled. Since we will change a afterwards, change x'_a and x'_c according to the definition in Step 2 (pretending that a was already moved towards b) if necessary, i.e., if an attachment of B different from d on the segment ad_0 on $\partial F'$ is closer to d_0 than x'_a, then x'_a should be changed to such an attachment closest to d_0. Similarly for x'_c. If $a = a_0$, it may also be necessary to change x_c and x'_d. When finished with all the unlabelled bridges at a, label all these bridges and change a to be the next vertex on e towards b. If necessary, change x_a, x'_a, x_b, etc., and then return to the beginning of Step 3.

We now have $a = x_a$. Next we check if $a = x'_a$, and if not, we repeat the above procedure in F'. Finally, do the same with b, c, d, so that we may assume from now on that $a = x_a = x'_a$, $b = x_b = x'_b$, etc.

If there is no unlabelled bridge attached at a then we move a to the next vertex towards b. After changing a we make sure to change x_a, x'_a, x_b, x'_b, etc. properly so that they are defined according to the definition given in Step 2. Then return to the beginning of Step 3. Do the same if there are no corresponding bridges at b, c, d, respectively.

From now on suppose that there are unlabelled bridges, A, B, C, D attached at a, b, c, d, respectively. If A is just the edge ad then we assign A to be embedded in F. One can easily see that ad does not overlap with any of the bridges previously assigned to F (unless some of the previously embedded bridges overlap with each other, in which case there is no embedding extension and we do not care if our algorithm does not work properly from this point on). Therefore our decision of embedding ad into F does not influence the embeddability of G. After assigning ad to F, mark this bridge as labelled and return to the beginning of Step 3.

If $D = ad$ we do the same. Suppose now that $A, D \neq ad$. We check next whether $A = D$. If so, we try to find another candidate for A (by possibly moving a towards b) and another candidate for D. To be more specific, we do not change A, D, a, or d until we find a proper alternative. In case of the search for the alternative for a, we look for the first bridge attached at a or as close to a as possible. If we find the bridge ad then we perform the same steps as described above. Otherwise we try to find the alternative for D that overlaps with A in F, or the alternative for A that overlaps with D. If such alternatives cannot be found, then we perform the same procedure as in the above case when A was just the edge ad, and after that return to the beginning of Step 3. From now on we may assume that we have either changed A (and possibly a), or D (and possibly d). Once we find whether to change A or D, if the vertex a or d has changed, we make sure to change $x_a, x'_a, x_b, x'_b, \ldots$ accordingly.

From now on we have $A \neq D$ and A, D overlap in F. It is easy to see that we still have $a \neq b$ and $d \neq c$. After clearing this part we make sure in the same way that $B \neq C$. Next we make sure that $A \neq C$ and $B \neq D$ by performing the same operation in the face F'. It may happen now that we obtain $A = D$ again. In this case we repeat all the procedure from the beginning. Finally, we end up with the case where $A \neq C, D$ and $B \neq C, D$. It may happen, however, that $A = B$. In such a case the end of our suffering is quite close. We embed $A = B$ first in F and next in F', and in each case we solve the embeddability problem as follows. Let us only describe the case when we embed $A = B$ in F. Then the remaining bridges are uniquely determined in which face they should go if there is an embedding. For example, C and D go to F', all unlabelled bridges attached at a point strictly between a and b must go to F', etc. Once we determine for all the bridges to be either in F or in F', we go to Step 6. We undertake the same steps if $C = D$.

Step 4. Now we have a, b, c, d distinct vertices and we have unlabelled bridges A, B, C, D which are pairwise distinct. By construction it follows that A and D overlap in F, B and C overlap in F, A and C overlap in F', and B and D overlap in F'. It is easy to see that only the following three cases can occur.

Case 4.1. A, B overlap in F, or C, D overlap in F': Then the only possibility for embedding A, B, C, D is to embed A and B in F', and embed C and D in F. Mark A, B, C, D as labelled and return to Step 3.

Case 4.2. A, B overlap in F', or C, D overlap in F: Then the only possibility for embedding A, B, C, D is to embed A and B in F, and embed C and D in F'. Mark A, B, C, D as labelled and return to Step 3.

Case 4.3. A, B do not overlap in F, F', and C, D do not overlap in F, F': Then A and B have only one vertex of attachment on e' and this attachment is the same for both of them. The same holds for the attachment of C and D on e. We have two possibilities how to embed A, B, C, D in F and F'. We have to check for both of them. But in each case the problem becomes quite simple—we can solve the embeddability of the remaining bridges by testing for planarity. Let us describe the details only for the case when we embed A, B in F and C, D in F'. Then all bridges are uniquely determined in which of the faces F, or F' they should be put. There is only one exception—if there is a bridge (an edge xy) attached only to the attachment x of A and B on e', and to the attachment y of C and D on e. If this bridge is really there, then we consider two subcases: we first put xy in F and check with the Step 6, and then put xy in F' and do the same. If xy is not among the bridges, we go to Step 6 as well.

Step 5. Now we have the following situation. Some bridges are already determined in which faces they will be put (the labelled bridges). The remaining bridges are attached to a segment cd on e', and to the single vertex a $(= b)$ on e. (We might have reached this point from another alternative, $c = d$, but for the sake of simplicity of notation we assume the above situation.) Construct the auxiliary graph G' as follows (see Fig. 5; note that the points a on the left and the right represent the same vertex). Its vertices are the vertices on the branch e', together with vertices $a, f_u, f_b, f'_u, f'_b, z_u, z_b$ and some other vertices corresponding to the bridges of K. The vertex z_u is adjacent to a, f_u, d_0, f'_u, vertex z_b is adjacent to a, f_b, c_0, f'_b, the neighbours of a are $f_u, f_b, f'_u, f'_b, z_u, z_b$, etc., as shown in Fig. 5. For each bridge Q of K we also add to G' the bridge Q with the following modification. If Q is attached to a vertex in the open segment da on ∂F and is determined to be in F then we replace all such attachments with an attachment at the vertex f_u. Similarly, the attachments on the bottom segment ac give rise to an attachment at f_b. The bridges in F' similarly attach to f'_u for the upper open segment ad, and

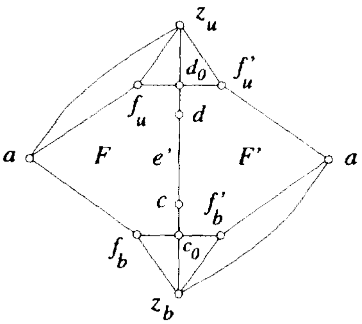

FIG. 5. The auxiliary graph G'.

attach to f'_b for the bottom segment ca. The attachments at a and on cd remain untouched. (We note that this is the step where we can not according to Proposition 2.2 replace the bridges by single vertices adjacent to the attachments. But this is restricted only for the labelled bridges that are attached to a and the segment cd only.) One can check that the decision about the planarity of G' will tell us if there is a possibility of the embedding of G extending the given embedding of K, and if yes, then it will tell us in which of the faces F, F' the remaining bridges should be put. Moreover, any such choice will be appropriate.

Step 6. Once decided which bridges should be embedded in F and which in F' we solve our problem by testing for planarity (cf. Section 5, Example 1), first in F and once again in F'. If we succeed in obtaining an embedding in each of the cases then the initial embedding task is complete. On the other hand, if the planarity test fails either in F or in F' then we know that G has no embedding extending the given embedding of K. This ends the algorithm.

One can easily check that the above algorithm really needs only linear time. The crucial fact is that each step can either be performed in constant time, or it contains a series of operations which need not be constant in each step but that all together there is a linear bound on the number of these operations. We have tried to comment on the least obvious cases within the description of the algorithm. □

7. Projective Planarity in Linear Time

After the preparation in previous sections we are ready to present the final algorithm for embedding graphs in the projective plane. Its linearity then proves our main result, Theorem 1.1. The algorithm runs as follows:

Step 1. Given the graph G on n vertices, check if G is simple and $|E(G)| \le 3n$. If not then STOP.

Step 2. Check if G is planar. In case of the positive answer return its plane embedding rotation system and STOP. Otherwise let K be a Kuratowski subgraph of G.

Step 3. Change G, K in such a way that G is 3-connected modulo K (Proposition 4.1). Let G and K denote the new graphs. If there was a non-trivial reduction then make sure at the end to change the embedding of G into an embedding description of the original graph (cf. the proof of Proposition 4.1).

Step 4. Change K so that there are no local bridges of K (Proposition 4.2).

Step 5. For every equivalence class of embeddings of K (cf. Section 3) try to extend it to an embedding of G. If successful in at least one of the cases then G has an embedding in the projective plane, otherwise not. So, fix an embedding of K in the projective plane and try to extend it as follows.

Step 5.1. For each face of K determine which bridges can be embedded in this face. This can be done by testing for planarity (cf. Section 5, Example 2) in linear time since K has only four or six faces.

Step 5.2. Let $\mathbf{B}_0$ be the set of bridges of K which are attached to at least two main vertices of K, or their attachments to K are not limited to two of the branches of K. Since G is 3-connected modulo K, the number of bridges in $\mathbf{B}_0$ is bounded by a constant if G admits an embedding in the projective plane. (The reader is invited to find the best possible constant.) Therefore there is a bounded number of possibilities how to embed K together with these bridges in $\tilde{\Sigma}_1$. By Proposition 2.2 we only need to know in which of the faces particular bridges from $\mathbf{B}_0$ are embedded. The following steps should be repeated for each of the possibilities.

Step 5.3. We assume now that an embedding of K in the projective plane is fixed and that the bridges of $\mathbf{B}_0$ are distributed to faces of K. We will decide for each of the remaining bridges in which of the faces it should be embedded. Once having made the decision for all of the bridges, we will perform the planarity test for the bridges in each of the faces to decide whether an embedding of G of the determined kind exists, or not. From now on we only discuss the way how to split the bridges among the faces. First of all, our job is trivial for those bridges which were found in Step 5.1 to have an embedding in a single face. The remaining bridges all have exactly two distinct embeddings with respect to K. Each of them is attached to an interior vertex of a branch of K and to exactly one other branch. Let $\mathbf{B}_1$ be the set of these bridges. By Corollary 3.5, one of the two faces allowing an embedding of a bridge from $\mathbf{B}_1$ is the face F (as denoted in Figs. 1 and 2). We say that the bridges in $\mathbf{B}_1$ that admit embeddings into the faces F and F' are of *type* F'.

Step 5.4. For each face $F' \neq F$ choose all the faces that have been determined to be in F', and add all the faces of type F'. By testing for planarity we can check if all these faces can be simultaneously embedded in F'. If this happens then we will embed all of them in F' since they will not block any of the remaining bridges. If not, then in every embedding of these bridges at least one of the bridges of type F' will be embedded in F.

Step 5.5. For each face $F' \neq F$ for which the above test was not successful we check if all bridges that have been assigned to F or to F', together with the bridges of type F', can be simultaneously embedded in the union of F and F'. For this task we either test for planarity (cf. Section 5, Example 3), or use the skew-planarity algorithm of Section 6. We call this procedure the *F'-test*. Of course, we should obtain the positive answer to all necessary F'-tests if G has the required embedding. Thus we assume that the F'-tests are all positive.

If K is embedded as in Fig. 1a then, since each bridge is attached to a non-main vertex, the F'-test is required for at most one face $F' \neq F$. (Otherwise there is no required embedding.) The embedding is obtained if and only if this test (if necessary at all) is positive.

If K is embedded as in Fig. 1b then there is only one face F' which might need to be tested for. The result is at hand.

In the last case of Fig. 2 the situation is slightly more complicated. There is either just one F'-test that is necessary (which we solve as in the above cases), or there are two of them, say F' and F''. In this case, after an embedding of G is reached, the only bridges of type F' or F'' that will be embedded in F will be attached at one side only to a main vertex of K, say x, and to a branch e' (respectively e'') on the other side of the face F. To check for such a possibility we put all the bridges of type F', that do not attach as required, to be in F'. Similarly for F''. (Note that F' and F'' are uniquely determined, but for x there are two choices which should be considered separately.) Finally, we solve the problem about the embeddability by testing for planarity (cf. Example 3 of Section 5), separately for F, F', and for F, F''. In both of these tests we have to construct a similar auxiliary graph G' as used in Step 5 of the skew-planarity algorithm in Section 6 (Fig. 5). This ends the procedure for distributing the bridges among the faces of K.

Using the results of previous sections, it is easy to see that the running time of the algorithm is linear. Step 1 is trivial. Step 2 can be performed in linear time by the results of [24, 25]. Steps 3 and 4 are linear by Propositions 4.1 and 4.2. Step 5 is performed only a constant number of times. The same is true if we add all the possibilities for several cases which arise in 5.2. Steps 5.1, 5.3, and 5.4 are easily seen to be linear since the number of faces F' for which the planarity tests are done is bounded by a small constant. The same holds for the F'-tests in 5.5.

Acknowledgment

I am greatly indebted to Martin Juvan for several discussions on the skew-planarity algorithm presented in Section 6.

References

1. D. Archdeacon, The complexity of the graph embedding problem, *in* "Topics in Combinatorics and Graph Theory" (R. Bodendiek and R. Henn, Eds.), pp. 59–64, Physica-Verlag, Heidelberg, 1990.
2. D. Archdeacon, A Kuratowski theorem for the projective plane, *J. Graph Theory* **5** (1981), 243–246.
3. 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.
4. N. Chiba, T. Nishizeki, S. Abe, and T. Ozawa, A linear algorithm for embedding planar graphs using *PQ*-trees, *J. Comput. System Sci.* **30** (1985), 54–76.
5. S. A. Cook and R. A. Reckhow, Time bounded random access machines, *J. Comput. System Sci.* **7** (1976), 354–375.
6. J. R. Fiedler, J. P. Huneke, R. B. Richter, and N. Robertson, Computing the orientable genus of projective graphs, preprint, 1989.
7. I. S. Filotti, G. L. Miller, and J. Reif, On determining the genus of a graph in $O(v^{O(g)})$ steps, *in* "Proceedings, 11th Annu. ACM STOC, Atlanta, Georgia, 1979," pp. 27–37.
8. H. de Fraysseix and P. Rosenstiehl, A depth-first search characterization of planarity, *Ann. Discrete Math.* **13** (1982), 75–80.
9. H. Glover, J. P. Huneke, and 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 and T. W. Tucker, "Topological Graph Theory," Wiley-Interscience, New York, 1987.
11. J. E. Hopcroft and R. E. Tarjan, Efficient planarity testing, *J. Assoc. Comput. Mach.* **21** (1974), 549–568.
12. M. Juvan, J. Marinček, and B. Mohar, Embedding graphs in the torus, in preparation.
13. M. Juvan, J. Marinček, and B. Mohar, Efficient algorithm for embedding graphs in arbitrary surfaces, in preparation.
14. A. Karabeg, Classification and detection of obstructions to planarity, *Linear and Multilinear Algebra* **26** (1990), 15–38.

15. B. Peruničić and Z. Durić, An efficient algorithm for embedding graphs in the projective plane, *in* "Graph Theory (Dubrovnik, 1985)" (D. Acketa *et al.*, Eds.), pp. 159–171, University of Novi Sad, Novi Sad, 1986.
16. N. Robertson and P. D. Seymour, Graph minors. VIII. A Kuratowski theorem for general surfaces, *J. Combin. Theory Ser. B* **48** (1990), 255–288.
17. N. Robertson and P. D. Seymour, Graph minors. XIII. The disjoint paths problem, preprint, revision of February 1992.
18. N. Robertson and P. D. Seymour, Graph minors. XXI. Graphs with unique linkages, preprint, 1992.
19. N. Robertson and P. D. Seymour, Graph minors. XXII. Irrelevant vertices in linkage problems, preprint, 1992.
20. N. Robertson and P. D. Seymour, An outline of a disjoint paths algorithm, *in* "Paths, Flows, and VLSI-Layout" (B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, Eds.), pp. 267–292, Springer-Verlag, Berlin, 1990.
21. C. Thomassen, The graph genus problem is NP-complete, *J. Algorithms* **10** (1989), 568–576.
22. C. Thomassen, Embeddings of graphs with no short noncontractible cycles, *J. Combin. Theory Ser. B* **48** (1990), 155–177.
23. W. T. Tutte, "Connectivity of Graphs," Univ. Toronto Press, Toronto/Oxford Univ. Press, London, 1966.
24. S. G. Williamson, Embedding graphs in the plane—Algorithmic aspects, *Ann. Discrete Math.* **6** (1980), 349–384.
25. S. G. Williamson, Depth-first search and Kuratowski subgraphs, *J. Assoc. Comput. Mach.* **31** (1984), 681–693.