

Testing Planarity of Geometric Automorphisms in Linear Time^{*}

Christoph Buchheim¹ ^{**} and Seok-Hee Hong² ^{***}

¹ Universität zu Köln, Institut für Informatik,
Pohligstraße 1, 50969 Köln, Germany
`buchheim@informatik.uni-koeln.de`

² NICTA (National ICT Australia) and School of Information Technologies,
University of Sydney, Australia
`shhong@it.usyd.edu.au`

Abstract. It is a well-known result that testing a graph for planarity and, in the affirmative case, computing a planar embedding can be done in linear time. In this paper, we show that the same holds if additionally we require that the produced drawing be symmetric with respect to a given automorphism of the graph. This problem arises naturally in the area of automatic graph drawing, where symmetric and planar drawings are desired whenever possible.

Key words: graph drawing, automorphisms, symmetries, planarity

1 Introduction

Both a small number of edge crossings and the display of symmetry have been identified as important criteria for the readability of two-dimensional drawings of a graph [26]. In the area of automatic graph drawing, the aim is to develop nice and well-readable drawings algorithmically. This leads to the two important problems of minimizing the number of edge crossings and maximizing the amount of symmetry in a layout of a given graph.

In particular, planar graphs play an important role in this area. One of the most prominent results in this context is the existence of linear time algorithms for testing planarity of a given graph and, in the affirmative case, for computing a planar embedding [3, 21].

On the other hand, symmetric drawings of graphs are clearly preferable to non-symmetric ones, as the complexity of the drawing for the human viewer decreases to the same degree as the symmetric structure in the drawing increases. Furthermore, symmetric drawings are regarded as more beautiful in general.

^{*} An extended abstract of this paper has been published in the proceedings of ISAAC 2002 [4].

^{**} The first author is supported by the Marie Curie Research Training Network ADONET 504438 funded by the EU. This paper was partially written when he was visiting the University of Sydney.

^{***} The second author was supported by a grant from the Australian Research Council. National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

For planar graphs, one can find maximally symmetric two-dimensional drawings in linear time [16–19]. In general, it is an NP-complete problem to decide whether a given graph has any non-trivial automorphism that is induced by a symmetric drawing [24]. Nevertheless, exact approaches for general graphs have been devised, using branch-and-cut [6] or group-theoretic methods [1]. Moreover, several heuristic approaches to the problem of drawing graphs symmetrically have been proposed [8, 14, 22]. It was also shown that the well-known spring embedder method has a tendency to display symmetry [12]. An important relaxation of the symmetry detection problem is to consider nearly symmetric drawings [7, 9, 10].

In the following, we assume throughout that a geometric automorphism π of some graph G is given, i.e., an automorphism of G that is induced by some two-dimensional drawing of G [12]. In general, there are many different drawings realizing π in this case; see Figs. 1(a) and 1(b) as an example. In particular, the number of edge crossings will vary for different visualizations of π .

In a former paper [5], we discussed the problem of minimizing the number of crossings over all possible drawings of G inducing π . With respect to this criterion, the drawing in Fig. 1(b) would be preferable to the one in Fig. 1(a), as it has less crossings. We showed that computing a drawing of a given geometric automorphism with a minimum number of crossings is NP-hard, even if strongly restricted in several variations.

In the present paper, we focus on the planarity problem for geometric automorphisms: given a geometric automorphism π of a graph G , is there a planar drawing of G inducing π ? We show that this problem can be solved in linear time. In the case of a positive answer, we can also construct such a drawing in linear time. Our results thus generalize the corresponding results for testing planarity of graphs. See Fig. 2 for examples of drawings displaying planar symmetries.

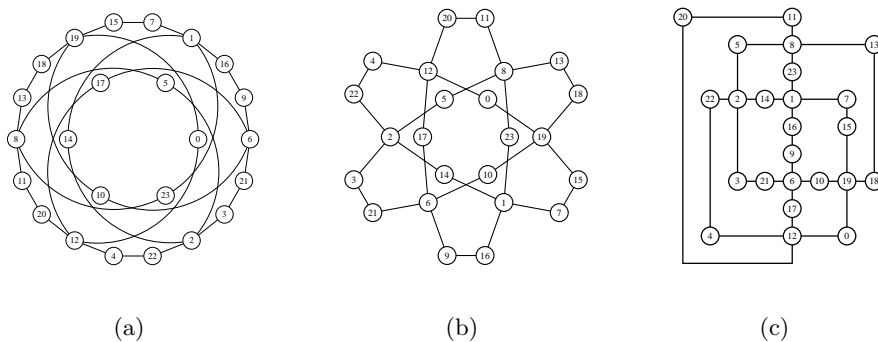


Fig. 1. Three drawings displaying the same graph. The graph is planar, as shown by the drawing (c), but the rotational automorphism of order six that is displayed in (a) and (b) cannot be displayed by any planar drawing.

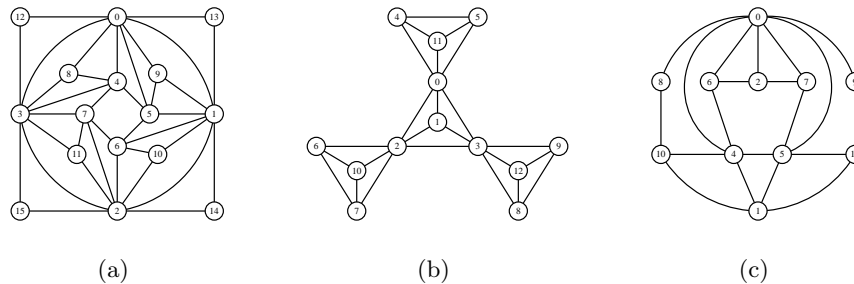


Fig. 2. Three drawings displaying planar automorphisms of different graphs. In (a), all automorphisms are rotations of order four, while (b) displays rotations of order three and three reflections. Drawing (c) displays only one reflectional automorphism.

Our problem is related to the problem of finding a planar automorphism group of maximum size. In particular, our algorithm is built up in a way similar to the algorithm for the latter problem presented in [16–19], which is based on the connectivity of the graph: the case of a triconnected graph is settled first, then SPQR-trees are used to reduce the biconnected case to the triconnected one, then the block-cutvertex-tree is used to reduce the oneconnected case to the biconnected one, and finally the disconnected case is solved. However, our result does not follow from the results in [16–19], as the symmetry detection algorithm cannot be forced to detect the given automorphism π . On the other hand, the fact that in our problem we deal with a specific automorphism allows a very natural construction of the desired planar drawing, if one exists, and thus leads to new insight about the connection between symmetry and planarity.

The remainder of this paper is organized as follows: in Section 2, we introduce the necessary definitions and give a more formal statement of the problems we consider. In Section 3, we recall the SPQR-tree data structure and examine the connection between automorphisms of a graph and its SPQR-tree. The main part is Section 4, where we show how to test planarity of geometric automorphisms in linear time. Section 5 concludes. In Appendix A, we illustrate our algorithm by an example.

2 Preliminaries

Throughout this paper, we consider a simple graph $G = (V, E)$ with n vertices and m edges. A *drawing* of a graph is a representation of its vertices by distinct points of the plane and of its edges by arbitrary curves in the plane that connect the corresponding points but do not intersect any other vertex point. A drawing of G is *planar* if no two curves representing edges of G intersect, i.e., have a common point except for possibly at a vertex point. A graph is said to be *planar* if it admits any planar drawing.

An *automorphism* of G is a permutation $\pi: V \rightarrow V$ of its vertices such that two vertices v and w are adjacent if and only if $\pi(v)$ and $\pi(w)$ are. The *order* of an automorphism π is the smallest positive integer k such that π^k is the identity. An induced subgraph $G' = (V', E')$ of G is *fixed* by π if $\pi(V') = V'$. In particular, a vertex v is fixed if $\pi(v) = v$, and an edge (v, w) is fixed if the set $\{v, w\}$ is fixed, i.e, if v and w are either both fixed or exchanged by π . If G' is a fixed subgraph, then the *restriction* $\pi|_{G'}$ of π to G' , defined by $\pi|_{G'}(v) = \pi(v)$ for all $v \in V'$, is an automorphism of G' . The *orbit* of a subgraph $G' = (V', E')$ of G under π is the graph with vertex set $\{\pi^k(v) \mid v \in V', k \text{ integer}\}$ and edge set $\{(\pi^k(v), \pi^k(w)) \mid (v, w) \in E', k \text{ integer}\}$. This in particular defines the orbit of a single vertex or a single edge. In the following, we will denote automorphisms by their cycle representations, i.e., by listing their nontrivial vertex orbits.

A drawing of G is called *reflectionally symmetric* if it is fixed by some non-trivial reflection of the plane in an axis, the *reflection axis*. Analogously, a *rotationally symmetric drawing* of G is a drawing of G fixed by some non-trivial rotation of the plane around a center point, the *rotation center*. A drawing is called *symmetric* if it is fixed by any isometry of the plane, or, equivalently, if it is reflectionally or rotationally symmetric [12]. Note that a drawing can be both rotationally and reflectionally symmetric.

A *geometric automorphism* of G is an automorphism induced by a symmetric drawing of G [12]. An automorphism π is thus geometric if there exist a drawing D of G into the plane and an isometry of the plane that fixes D and that induces π . In this case, we also say that D *displays* π , or that D is a *drawing* of π . An example of a non-geometric automorphism is given in Fig. 3. Note that a drawing of π is at the same time a drawing of π^e , for every integer e . A *reflection (rotation)* of G is an automorphism of G induced by a reflectionally (rotationally) symmetric drawing of G .

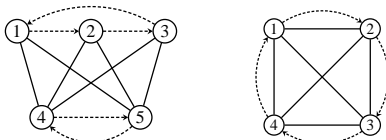


Fig. 3. The automorphism $(123)(45)$ of $K_{2,3}$ (left) is not geometric, while the automorphism (1234) of K_4 displayed on the right is geometric but not planar.

For a given symmetric drawing D inducing a rotation π of order k , there is exactly one $e \in \{1, \dots, k\}$ such that rotating D by $360/k$ degrees in clockwise order around the rotation center maps each vertex v to $\pi^e(v)$. The number e is the *exponent* of the drawing D ; see Fig. 4.

An automorphism of G is *planar (reflectionally planar, rotationally planar)* if it is induced by a planar (reflectionally planar, rotationally planar) drawing of G [16–19]. In this paper, we address the *symmetric planarity problem*:

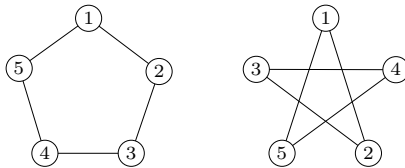


Fig. 4. Drawings of the rotation $\pi = (12345)$ having exponents one and three.

Problem (SPL): Given a geometric automorphism π of G , decide whether it is planar. If it is, compute a planar drawing of G inducing π .

Observe that an automorphism might be reflectionally planar without being rotationally planar, or vice versa. To give a simple example, consider the cyclic graph C_4 on the vertices 1, 2, 3, 4. Then the permutation of vertices (13)(24) is both a reflection and a rotation. Moreover, it is rotationally planar, but not reflectionally planar; see Fig. 5. The opposite is true for the permutation (12)(34) of the same graph. Because of these difficulties, we consider the two following problems separately. Solving both problems suffices for solving (SPL).

Problem (REFPL): Given a geometric automorphism π of G , decide whether it is reflectionally planar. If it is, compute a reflectionally planar drawing of G inducing π .

Problem (ROTPL): Given a geometric automorphism π of G , decide whether it is rotationally planar. If it is, compute a rotationally planar drawing of G inducing π .

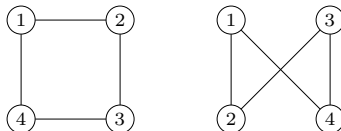


Fig. 5. The automorphism (13)(24) of the graph C_4 is planar as a rotation (left), but non-planar as a reflection (right).

3 SPQR-trees

The SPQR-tree data structure has been introduced in 1996 by Di Battista and Tamassia [11] and has since been used successfully in a variety of algorithms for automatic graph drawing. Its purpose is to divide a biconnected graph into its triconnected components [20, 28] and to allow to handle these components

efficiently. In this paper, we use a slightly simplified version of SPQR-trees in that we do not deal with Q-nodes explicitly.

First we define the triconnected components of G by a decomposition rule: as long as there is a separation pair $\{v, w\}$ of G that separates two subgraphs G_1 and G_2 having at least two edges each, we add a virtual edge (v, w) to both G_1 and G_2 and decompose the two resulting graphs recursively. In this way, we end up with a collection of three types of multi-graphs: triconnected graphs, pairs of vertices connected by triple edges (*triple bonds*), and cycles of length three (*triangles*). We then obtain the triconnected components of G by merging triple bonds on the same pair of vertices into one *bond*, and by merging adjacent triangles into *polygons*; whenever two components are merged, the two virtual edges representing the respective other component are deleted.

Now the SPQR-tree T corresponding to G is defined as follows. The nodes of T are in one-to-one correspondence to the triconnected components of G . The *skeleton* of a node v of T is the triconnected component represented by v . Nodes representing polygons are called *S-nodes*, nodes representing bonds are *P-nodes*, and nodes representing triconnected graphs are *R-nodes*. Now each skeleton is a graph defined on the nodes of G , with edges that either belong to G or are virtual edges representing other triconnected components. Two nodes in T are defined to be adjacent if and only if the corresponding skeletons contain virtual edges representing each other.

An example of a biconnected graph along with its SPQR-tree is given in Fig. 6. Virtual edges are represented by dotted lines. The SPQR-tree of a graph can be computed in linear time [15]; this is true also with our small modification.

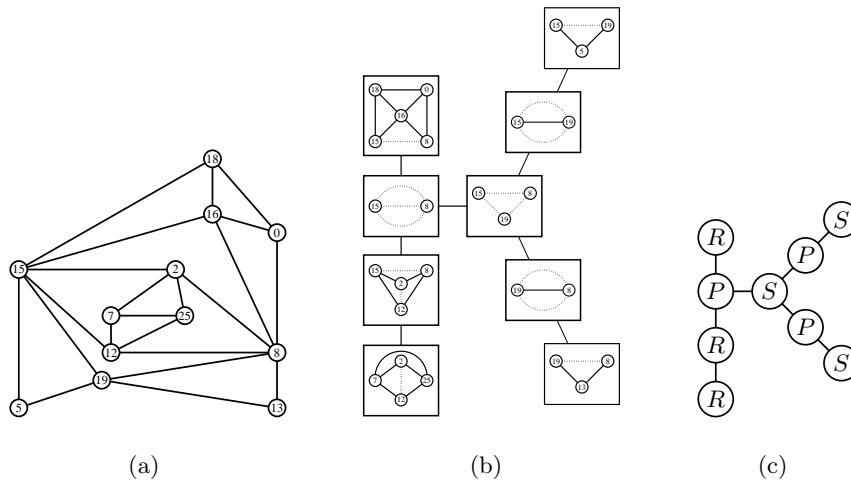


Fig. 6. A biconnected graph (a) and its SPQR-tree (c); the skeletons are shown in (b).

Coming back to symmetries, observe that any automorphism π of G induces an automorphism π_T of T . If some node of T is fixed by π_T , we further get an induced automorphism on the skeleton ν corresponding to that node, which we will simply call the restriction of π to ν , denoted by $\pi|_\nu$. It is easy to see—and will be used in the following—that the restriction of a geometric automorphism to any fixed skeleton is a geometric automorphism of the same type again, i.e., remains a rotation or reflection. Furthermore, the restriction of a planar automorphism to a skeleton remains planar.

Lemma 1. *For every automorphism π of G , the nodes of T with skeletons fixed by π induce a subtree of T .*

Proof. Consider the induced automorphism π_T of T . If there was a path P in T connecting two fixed nodes but containing non-fixed nodes, then the nodes in P and $\pi_T(P)$ would induce a subgraph of T containing a cycle. \square

Lemma 2. *Let π be any automorphism of G . Then either there is a skeleton of T fixed by π or there is a separation pair of G fixed by π .*

Proof. Consider the center of T , i.e., the set C of nodes maximizing the minimal distance to any leaf of T . If C contains a single node, this node must be fixed by all automorphisms of T , in particular by π_T . The corresponding skeleton is thus fixed by π . Otherwise C contains two adjacent nodes. In this case, the connecting edge is fixed by π_T , i.e., the corresponding separation pair of G is fixed by π . \square

In a similar way as the triconnected components of biconnected graphs can be managed by SPQR-trees, the more elementary block-cutvertex-trees are used to manage the biconnected components (blocks) of a connected graph. Every block and every cutvertex of G is represented by a node in this tree. Two nodes are adjacent if and only if one corresponds to a cutvertex c and the other to a block B such that c belongs to B . Note that block-cutvertex-trees can be computed in linear time [27].

4 Testing Planarity

In this section, we show that the problem (SPL) can be solved in linear time. First observe that planarity does not depend on whether we restrict ourselves to straight-line drawings or not. This follows from

Theorem 1 (Mani [23]). *Every triconnected planar graph can be drawn as the skeleton of a polytope in the three-dimensional Euclidean space such that every automorphism of the graph is induced by an isometry of the polytope.*

Note that we allow all isometries of the polytope here, including those that do not correspond to physical rotations or reflections, like reflections in a point.

Theorem 2. *For every planar drawing of an automorphism π , there exists a planar drawing of π with straight-line edges that induces the same embedding of the underlying graph.*

Proof. Consider a planar drawing of G inducing the automorphism π and let F be its outer face, which is necessarily fixed by π . First assume that G is triconnected. Choose a polytope P as in Mani's Theorem. Then F corresponds to a facet of P fixed by π , so that an appropriate projection of P to the plane yields a planar straight-line drawing of π with F as its outer face. As G is triconnected, the induced embedding is the same as the original one.

If G is not triconnected, we claim that there is a triconnected graph G' with a planar automorphism π' such that G is a fixed subgraph of G' and $\pi'|_G = \pi$; the result for π' then implies the one for π . In fact, if G is biconnected, the graph G' can be obtained by the star triangulation [19], which adds a new vertex into each face of G that is adjacent to each vertex of the face. If G is one-connected or disconnected, it is easy to make it biconnected without losing symmetry. \square

It is a well-known result by Fáry [13] that every planar graph admits a planar straight-line drawing. Theorem 2 shows that this still holds under the additional requirement that a given automorphism be displayed.

In the following, we will prove our main result. The outline of our algorithm to solve problem (SPL) is similar to the one of the planar automorphism detection algorithm presented in [16–19]. In particular, we also consider the triconnected, biconnected, one-connected, and disconnected cases one after another. We split up the proof according to these cases.

The general idea is the following: for triconnected graphs, the result follows from Mani's Theorem. For a k -connected graph with $k = 0, 1, 2$, we consider its $(k + 1)$ -connected components and distinguish those being fixed by the given automorphism from those not being fixed. On the fixed components, a geometric automorphism of G induces a geometric automorphism of the same type, so we can apply the $(k + 1)$ -connected case. The non-fixed components can be divided into orbits with respect to the given automorphism; components in the same orbit are isomorphic. For every orbit, we can pick one component, draw it using an arbitrary planar drawing algorithm, and draw the other components in the same orbit by copying this drawing along the isometry corresponding to the given automorphism. The only tricky question is how to glue together all single component drawings preserving planarity.

In the first step of our algorithm, we check the graph G for planarity, which can be done in linear time [3, 21]. In the case of a negative result, we know that π is neither reflectionally planar nor rotationally planar, so in the following we may always assume that G is planar and hence $m \in O(n)$, as G is a simple graph by our general assumption.

Lemma 3. *The problems (REFPL) and (ROTPL) can be solved in $O(n)$ time for triconnected graphs.*

Proof. The combinatorial embedding of G is unique up to mirroring in this case. We choose one of the two embeddings and check whether there is a face F fixed by π such that $\pi|_F$ is reflectionally (rotationally) planar. As F is a cycle, this can easily be done in linear time. If the answer is negative, we can state that π is not reflectionally (rotationally) planar. Indeed, the outer face in any reflectionally

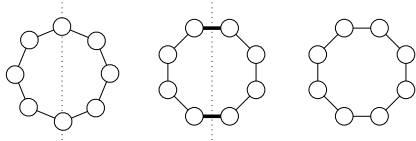


Fig. 7. The skeleton of an S-node drawn reflectionally planar (left and middle) and rotationally planar (right); fixed edges are drawn boldly.

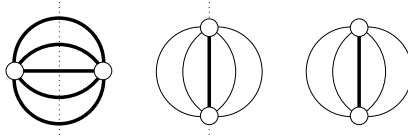


Fig. 8. The skeleton of a P-node drawn reflectionally planar (left and middle) and rotationally planar (right); fixed edges are drawn boldly.

(rotationally) planar drawing of π must be fixed by π and the restriction of π to the outer face is surely reflectionally (rotationally) planar; compare Lemma 1 in [19].

Otherwise, if such a face F exists, we can use the same projection as in the proof of Theorem 2 to show that the unique topological embedding of G with outer face F can be realized by a reflectionally (rotationally) planar drawing of π . A linear time algorithm for computing a nice drawing is devised in [19]. \square

Lemma 4. *The problems (REFPL) and (ROTPL) can be solved in $O(n)$ time for biconnected graphs.*

Proof. Consider the SPQR-tree T of G . First determine for every node of T whether its skeleton is fixed by π or not. For every fixed skeleton ν , we next check whether $\pi|_{\nu}$ is reflectionally (rotationally) planar. For R-nodes, we can use Lemma 3, as the skeleton of an R-node is triconnected. For S-nodes and P-nodes, this is trivial: the skeleton of an S-node is a cycle, so that reflectional (rotational) planarity of $\pi|_{\nu}$ can be checked easily in this case; see Fig. 7. The skeleton of a P-node is a bunch of parallel edges, so that $\pi|_{\nu}$ in this case is reflectionally planar if and only if either the two vertices are exchanged or at most one of the edges is fixed and the remaining edges are divided into orbits of size two. It is rotationally planar if and only if the two vertices are exchanged and at most one of the edges is fixed and the remaining edges are divided into orbits of size two; see Fig. 8.

In the event of any negative answer, i.e., if there is any fixed skeleton ν such that $\pi|_{\nu}$ is not reflectionally (rotationally) planar, we can state that π is not reflectionally (rotationally) planar. Otherwise, we claim that π is reflectionally (rotationally) planar and construct a reflectionally (rotationally) planar drawing inducing π as follows.

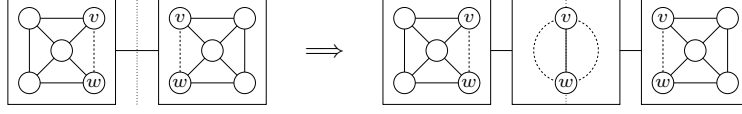


Fig. 9. The case of a fixed separation pair.

First, we may assume that some skeleton of T is fixed by π . If this is not true, by Lemma 2, some separation pair $\{v, w\}$ of G is fixed by π ; let f be the corresponding edge of T . We then introduce a new edge (v, w) fixed by π , which can be removed in the final drawing. In other words, we split up f by introducing a new P-node that is fixed by π ; see Fig. 9.

Now choose any fixed skeleton ν and compute any reflectionally (rotationally) planar drawing of the geometric automorphism $\pi|_{\nu}$, using Lemma 3 for R-nodes and trivial algorithms for P-nodes and S-nodes. If there is any other skeleton fixed by π , then by Lemma 1 there is a fixed skeleton neighboring ν in T , so that one of the virtual edges e of ν must be fixed by $\pi|_{\nu}$; either e is reversed by $\pi|_{\nu}$ or both end vertices of e are fixed. Let e represent the skeleton ν' and let $\{v, w\}$ be the corresponding separation pair. We can now replace the drawing of e by a reflectionally (rotationally) planar drawing of $\pi|_{\nu'}$ whose outer face contains the separation pair $\{v, w\}$.

More precisely, we compute a planar drawing of $\pi|_{\nu'}$ and remove the virtual edge (v, w) corresponding to ν in this drawing. The vertices v and w then share a face F in this drawing of $\pi|_{\nu'}$. As F contained the fixed edge (v, w) before, it must be fixed by $\pi|_{\nu'}$. We may assume that F is the outer face of the computed planar drawing of $\pi|_{\nu'}$; otherwise we can obtain this situation by applying the transformation $(x, y) \mapsto (x/(x^2 + y^2), y/(x^2 + y^2))$ to the drawing of $\pi|_{\nu'}$, where we assume that the former center point of (v, w) is the origin of \mathbf{R}^2 . Notice that this transformation is not linear, but it is injective on $\mathbf{R}^2 \setminus \{0\}$, so that the transformed drawing is planar again. Furthermore, the property of the drawing being rotationally (reflectionally) symmetric is preserved.

Replacing the virtual edge e in the drawing of $\pi|_{\nu}$ by the so constructed drawing of $\pi|_{\nu'}$ now yields a planar drawing of $\pi|_{\nu \cup \nu'}$. The two drawings fit together as the skeletons ν and ν' share the vertices v and w . Obviously, the drawings replacing edges can be made narrow enough to make sure that the whole drawing remains planar. We iterate this replacement process until all fixed skeletons are drawn, i.e., until all remaining virtual edges in the drawing are non-fixed; see Fig. 10.

Finally, we have to replace these non-fixed virtual edges in a symmetrical way. For this, we choose one edge out of each orbit of virtual edges not fixed by $\pi|_{\nu}$. For each chosen edge e , we consider the subgraph G' of G represented by e in ν , i.e., the subgraph resulting from the recursive replacement of virtual edges starting from e . We compute any planar drawing of G' with the separation pair of e in the outer face; this is possible as the separation pair is connected outside G' and G is planar. For all k , we then replace the non-fixed virtual edge $\pi^k(e)$ by

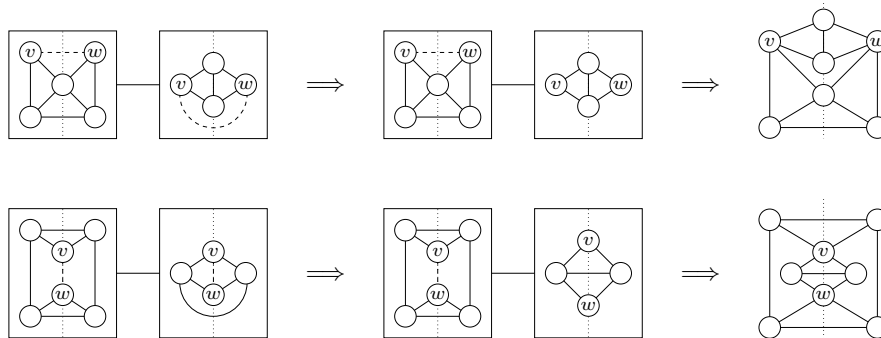


Fig. 10. Replacing fixed virtual edges by fixed skeletons.

the image of this drawing under the isometry corresponding to π^k . Again, the drawings of subgraphs replacing edges can be chosen narrow enough to make sure that the whole drawing remains planar. \square

Lemma 5. *The problems (REFPL) and (ROTPL) can be solved in $O(n)$ time for one-connected graphs.*

Proof. For one-connected graphs, we consider the block-cutvertex-tree of the graph G instead of its SPQR-tree. For each block B of G that is fixed by π , we first check whether $\pi|_B$ is reflectionally (rotationally) planar, using Lemma 4. In the event of any negative result, we can state that π is not reflectionally (rotationally) planar. Otherwise, the problems (REFPL) and (ROTPL) are handled differently. Note that at this point we cannot claim yet that π is planar, other than in the biconnected case.

We start with the easier problem (ROTPL), where we first check whether the number of fixed blocks is greater than one. If so, we can state that π is not rotationally planar; see Theorem 9 in [18]. Otherwise, we can show that π is rotationally planar by constructing a drawing as follows.

If there is exactly one fixed block B , we can compute a rotationally planar drawing of $\pi|_B$ by Lemma 4. Then we can add the remaining non-fixed blocks symmetrically, one orbit after another. For every such block B' containing a cut vertex c already drawn, we first compute any planar drawing of B' with c on its outer face. Then, for all k , we attach the image of this drawing under the isometry corresponding to π^k to the cut vertex $\pi^k(c)$, thus adding a drawing of $\pi^k(B')$. Obviously, this can be done without creating any edge crossings; see Fig. 11(a) for an illustration. Notice that c might be fixed by π , in this case all vertices $\pi^k(c)$ are identical.

Next, assume that no block of G is fixed by π . Then some cut vertex c of G must be fixed by π . Now the subgraphs of G connected by c can be divided into orbits, drawn in an arbitrary planar way with c on the outer face, and placed around c in a similar way as in the case of a fixed block; see Fig 11(b).

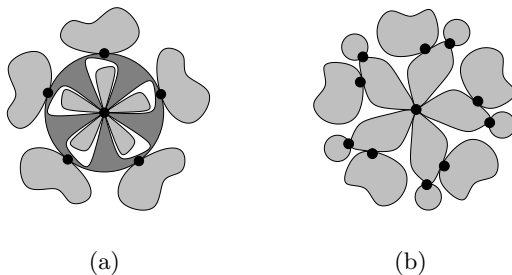


Fig. 11. Reduction to biconnected graphs in the rotation case; the fixed block in (a) is darkly shaded.

Now consider the problem (REFPL). We claim that π is reflectionally planar if and only if for every fixed block B the restriction $\pi|_B$ is reflectionally planar and for every fixed cut vertex c in B , either c has at most one fixed neighbor in B or c belongs to at most one fixed block (namely B). Using Lemma 4, this property can be checked in linear time.

It is easy to see that π cannot be reflectionally planar otherwise, i.e., if the restriction to some fixed block is not reflectionally planar or if there are two fixed blocks B_1 and B_2 sharing a cut vertex c such that in B_1 the vertex c has two fixed neighbors v_1 and v_2 . In the second case, the vertex c would have to lie between v_1 and v_2 on the reflection axis in any drawing of π . It is easy to see that any drawing of $\pi|_{B_2}$ would have to cross either one of the edges (c, v_1) and (c, v_2) or a path connecting v_1 and v_2 in B_1 not using c , which exists as B_1 is biconnected; see Fig. 12 for an example.

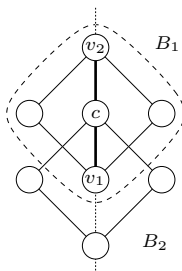


Fig. 12. A non-planar reflection with planar blocks.

So it remains to show that a reflectionally planar drawing of π can be computed in linear time if the above property holds. If no block of G is fixed by π , there must be a fixed cut vertex c . In this case, we can show as in the rotation

case that π is reflectionally planar. So assume that there is at least one block B of G fixed by π . We first compute a drawing of $\pi|_B$ using Lemma 4. If there is another fixed block, there must be a fixed block B' sharing a fixed cut vertex c with B . We then compute a planar drawing of $\pi|_{B'}$ with c on the outer face. By the property checked in the first step of the (REFPL) case, the cut vertex c has at most one fixed neighbor, so we know that at least one face containing c is fixed by π , i.e., contains some free segment of the reflection axis in the given drawing of $\pi|_B$. We can insert the drawing of $\pi|_{B'}$ into this fixed face, attaching it to the drawing of $\pi|_B$ at the vertex c such that the resulting drawing is a planar drawing of $\pi|_{B \cup B'}$; see Fig. 13(a).

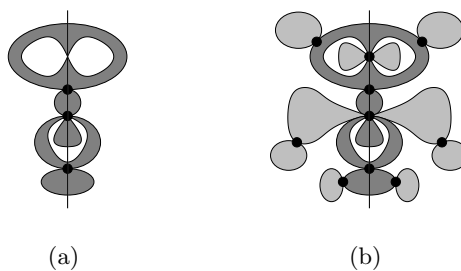


Fig. 13. Reduction to biconnected graphs in the reflection case. Fixed blocks are first drawn one after another (a), then non-fixed blocks are added symmetrically (b).

Now we can add the remaining fixed blocks iteratively. However, it can now happen that we have to add a block B at a cut vertex c such that two other blocks B_1 and B_2 containing c have been drawn before, and such that both contain a fixed neighbor of c . In this case, the fact that π is a geometric automorphism implies that B does not contain a fixed neighbor of c , as otherwise the fixed vertex c would have three fixed neighbors. Thus, we can temporarily remove the drawing of $\pi|_{B_1}$, say, insert a drawing of $\pi|_B$ in the freed face, and reinsert the drawing of $\pi|_{B_1}$ inside the face of B containing a free segment of the reflection axis; see Fig. 14.

Finally, the non-fixed blocks have to be added. This can be done as in the rotation case: for every block B not drawn yet but sharing a cut vertex c with a block already drawn, we compute any planar drawing of B with c on its outer face. We then insert this drawing into any face adjacent to c . A mirror image of this drawing is added to $\pi(c)$ as a drawing of $\pi(B)$. Obviously, this can be done without creating any edge crossings; see Fig. 13(b). \square

Lemma 6. *The problems (REFPL) and (ROTPL) can be solved in $O(n)$ time for disconnected graphs.*

Proof. For the problem (REFPL), we first check whether the restriction of π to every fixed connected component is reflectionally planar, using Lemma 5. If not,

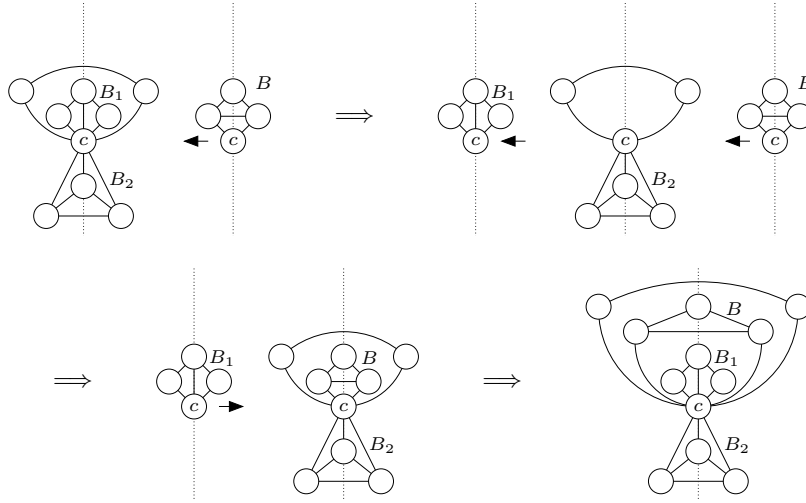


Fig. 14. Deciding the nesting order of fixed blocks in the reflection case.

we can state that the automorphism π is not reflectionally planar. Otherwise, we can easily arrange reflectionally planar drawings of all fixed components along the reflection axis. The remaining components are divided into pairs of isomorphic subgraphs mapped to each other by π . Thus we can construct a planar drawing for one subgraph of each pair and mirror the drawing with respect to the reflection axis to get a symmetric drawing for the other subgraph. Obviously, this can be done without creating edge crossings; see Fig. 15.

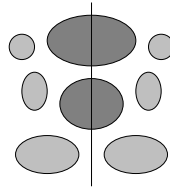


Fig. 15. Reduction to connected graphs in the reflection case.

For the problem (ROTPL), it is easy to see that the automorphism π is rotationally planar if and only if its restriction to each fixed connected component is rotationally planar with the same exponent e , and a drawing of π can easily be constructed by drawing the fixed components around each other; see Fig. 16. Note that only one connected component of the graph can contain a fixed vertex if π is a rotation.

It remains to explain how to find a common exponent for all fixed components or to state that there is none. Fig. 17 shows that this is crucial in the rotation

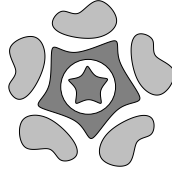


Fig. 16. Reduction to connected graphs in the rotation case.

case. So let G' be any fixed component of G . We claim that either there exists an integer e such that any planar drawing of $\pi|_{G'}$ has exponent e or $-e$, or there is a planar drawing of $\pi|_{G'}$ for any exponent; moreover, for every component, all possible exponents can be determined in linear time. Then it is easy to check whether there is a common exponent for all fixed components, and if there is one, to construct a drawing with this exponent for the whole graph.

To prove our claim, first assume that G' has a cut vertex fixed by $\pi|_{G'}$. Then the exponent is arbitrary. A drawing with a given exponent can be produced by permuting the components of G' resulting from deleting c , according to this exponent. Otherwise, if G' does not contain a fixed cut vertex, it remains connected after removing any fixed vertex—as π is a rotation, there can be at most one fixed vertex. Let v be any non-fixed vertex of G' . Choose any path from v to $\pi|_{G'}(v)$ without fixed vertices and determine e such that $(\pi|_{G'})^e(v)$ is the first vertex after v on this path that belongs to the orbit of v . Such e exists, as $\pi|_{G'}(v)$ belongs to the path. Then e and $-e$ are the only possible exponents of a planar drawing of $\pi|_{G'}$, as otherwise any path from v to $(\pi|_{G'})^e(v)$ would have to cross its own image under π . \square

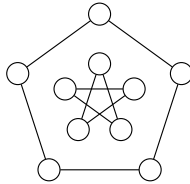


Fig. 17. A non-planar rotation with planar components.

We can now conclude the proof of

Theorem 3. *Problem (SPL) can be solved in $O(n)$ time.*

Proof. The automorphism π is planar if and only if it is induced by a reflectionally or a rotationally symmetric planar drawing of G [12]. Hence the result follows from Lemmas 3 to 6. \square

5 Conclusion

We showed that planarity of geometric automorphisms can be tested in linear time and that planar drawings can be computed in linear time whenever they exist. This is true in spite of the fact that the crossing minimization problem for geometric automorphisms is very hard [5]. However, as in the case of graphs without fixed automorphisms, one can use the presented algorithm for a heuristic approach to the crossing minimization problem via the computation of maximal planar symmetric subgraphs of G with respect to π : starting with an empty graph H and traversing all edge orbits of π , one can check in linear time whether adding the current edge orbit to H would destroy planarity of $\pi|_H$ or not. In the former case, the edge orbit is discarded, while in the latter case it is added permanently to H . It is clear that the resulting geometric automorphism $\pi|_H$ is maximally planar.

Finally, we would like to mention that our algorithm, as presented, can only check single geometric automorphisms for planarity. However, the group of all automorphisms displayed by a specific drawing of the graph cannot always be generated by a single automorphism. More precisely, it can be a dihedral group, jointly generated by a rotational and a reflectional symmetry [12]. It would thus be desirable to have an algorithm checking simultaneous planarity of two geometric automorphisms, i.e., checking whether there is a planar drawing of the graph displaying both. We are convinced that the algorithm presented in this paper can be extended to this case.

References

1. D. Abelson, S. Hong, and D. Taylor. A group-theoretic method for drawing graphs symmetrically. In S. G. Kobourov and M. T. Goodrich, editors, *Graph Drawing 2002*, volume 2528 of *LNCS*, pages 86–97. Springer-Verlag, 2002.
2. S. Bachl, F. J. Brandenburg, and D. Gmach. Computing and drawing isomorphic subgraphs. *Journal of Graph Algorithms and Applications*, 8(2):215–238, 2004.
3. K. Booth and G. Lueker. Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13:335–379, 1976.
4. C. Buchheim and S. Hong. Crossing minimization for symmetries. In P. Bose and P. Morin, editors, *ISAAC 2002*, volume 2518 of *LNCS*, pages 563–574. Springer-Verlag, 2002.
5. C. Buchheim and S. Hong. Crossing minimization for symmetries. *Theory of Computing Systems*, 38(3):293–311, 2005.
6. C. Buchheim and M. Jünger. Detecting symmetries by branch & cut. In P. Mutzel, M. Jünger, and S. Leipert, editors, *Graph Drawing 2001*, volume 2265 of *LNCS*, pages 178–188. Springer-Verlag, 2001.
7. C. Buchheim and M. Jünger. An integer programming approach to fuzzy symmetry detection. In G. Liotta, editor, *Graph Drawing 2003*, volume 2912 of *LNCS*, pages 166–177. Springer-Verlag, 2004.
8. H. Carr and W. Kocay. An algorithm for drawing a graph symmetrically. *Bulletin of the ICA*, 27:19–25, 1999.
9. H. Chen, H. Lu, and H. Yen. On maximum symmetric subgraphs. In J. Marks, editor, *Graph Drawing 2000*, volume 1984 of *LNCS*, pages 372–383. Springer-Verlag, 2001.
10. M. Chuang and H. Yen. On nearly symmetric drawings of graphs. In *IV 2002*, pages 489–494, 2002.
11. G. Di Battista and R. Tamassia. On-line planarity testing. *SIAM Journal on Computing*, 25(5):956–997, 1996.
12. P. Eades and X. Lin. Spring algorithms and symmetry. *Theoretical Computer Science*, 240(2):379–405, 2000.
13. I. Fáry. On straight lines representations of planar graphs. *Acta Scientiarum Mathematicarum*, 11:229–233, 1948.
14. H. de Fraysseix. An heuristic for graph symmetry detection. In J. Kratochvíl, editor, *Graph Drawing 1999*, volume 1731 of *LNCS*, pages 276–285. Springer-Verlag, 1999.
15. C. Gutwenger and P. Mutzel. A linear time implementation of SPQR trees. In J. Marks, editor, *Graph Drawing 2000*, volume 1984 of *LNCS*, pages 77–90. Springer-Verlag, 2001.
16. S. Hong and P. Eades. Symmetric layout of disconnected graphs. In T. Ibaraki, N. Katoh, and H. Ono, editors, *ISAAC 2003*, volume 2906 of *LNCS*, pages 405–414. Springer-Verlag, 2003.
17. S. Hong and P. Eades. Drawing planar graphs symmetrically II: Biconnected planar graphs. *Algorithmica*, 42(2):159–197, 2005.
18. S. Hong and P. Eades. Drawing planar graphs symmetrically, III: Oneconnected planar graphs. *Algorithmica*, 44(1):67–100, 2006.
19. S. Hong, B. McKay, and P. Eades. Symmetric drawings of triconnected planar graphs. In *SODA 2002*, pages 356–365, 2002.

20. J. Hopcroft and R. Tarjan. Dividing a graph into triconnected components. *SIAM Journal on Computing*, 2:135–158, 1973.
21. J. Hopcroft and R. Tarjan. Efficient planarity testing. *Journal of the ACM*, 21:549–568, 1974.
22. R. J. Lipton, S. C. North, and J. S. Sandberg. A method for drawing graphs. In *ACM Symposium on Computational Geometry*, pages 153–160, 1985.
23. P. Mani. Automorphismen von polyedrischen Graphen. *Mathematische Annalen*, 192:279–303, 1971.
24. J. Manning. Computational complexity of geometric symmetry detection in graphs. In *Great Lakes Computer Science Conference*, volume 507 of *LNCS*, pages 1–7. Springer-Verlag, 1989.
25. J. Manning and M. J. Atallah. Fast detection and display of symmetry in outer-planar graphs. *Discrete Applied Mathematics*, 39(1):13–35, 1992.
26. H. Purchase. Which aesthetic has the greatest effect on human understanding? In G. Di Battista, editor, *Graph Drawing '97*, volume 1353 of *LNCS*, pages 248–261. Springer-Verlag, 1997.
27. R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1:146–160, 1972.
28. W. Tutte. *Graph Theory*, volume 21 of *Encyclopedia of Mathematics and Its Applications*. Addison-Wesley, 1984.

A An Example

In this appendix, we illustrate the construction of a planar drawing in the reflectionally symmetric case by an example. The graph G is displayed in Fig. 18. The given automorphism π is marked by the dashed arrows showing which vertices of G are mapped to each other by π , all other vertices are fixed.

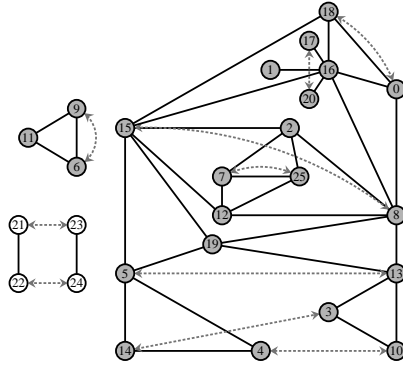


Fig. 18. The graph G and the automorphism π . Vertices belonging to a fixed connected component are shaded; two components are fixed, two are not.

As G is disconnected, the first step of the algorithm consists of determining the connected components of G and dividing them into fixed components and pairs of non-fixed components being mapped to each other by π .

We first draw the large fixed component G' of G shown in Fig. 19. For this, we have to compute its block-cutvertex-tree. The cut vertices are marked in Fig. 19.

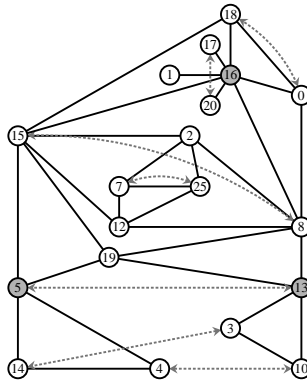


Fig. 19. The connected component G' with cut vertices shaded.

We continue the construction of a planar drawing of π with the largest block G'' of G' , shown in Fig. 20 along with its SPQR-tree T .

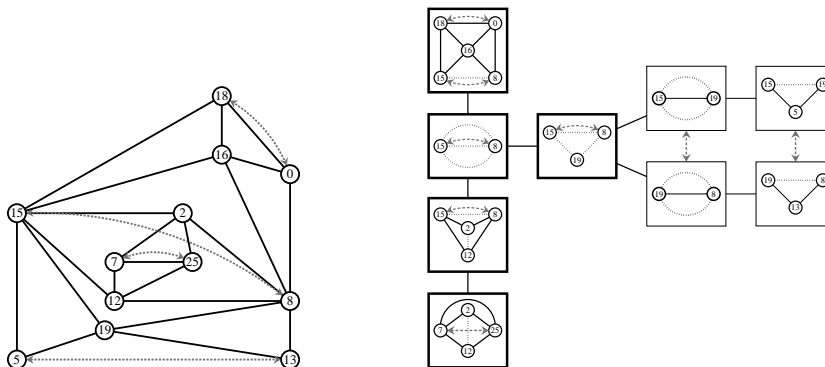


Fig. 20. The block G'' and its SPQR-tree T . The nodes of T corresponding to fixed skeletons are marked by bold frames.

We can choose any of the fixed nodes in T to start the recursive construction of the drawing. Here, we choose the R-node drawn at the top of the SPQR-tree. The step-by-step replacement of fixed edges by drawings of fixed skeletons is displayed in Fig. 21.

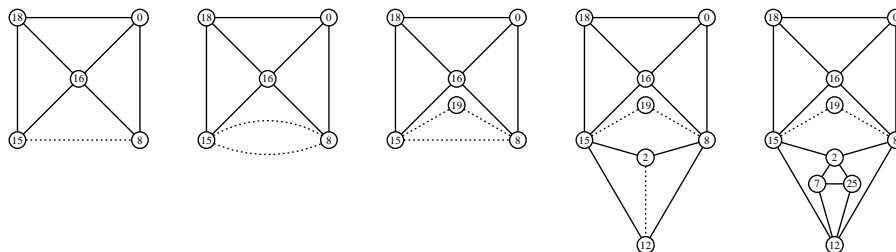


Fig. 21. Replacing fixed edges recursively.

Note that choosing a different replacement order would have led to different embeddings of G'' . After replacing all fixed edges, the remaining non-fixed edges are replaced in a symmetrical way, see Fig. 22. Finally, the remaining blocks and connected components can be added as shown in Figs. 23(a) and 23(b).

Finally, consider the example given in Fig. 1. The graph is biconnected, its SPQR-tree is a star with an R-node at the center and twelve S-nodes around it.

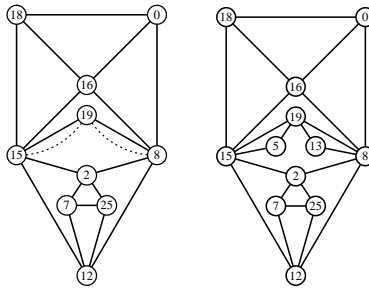


Fig. 22. Replacing non-fixed edges.

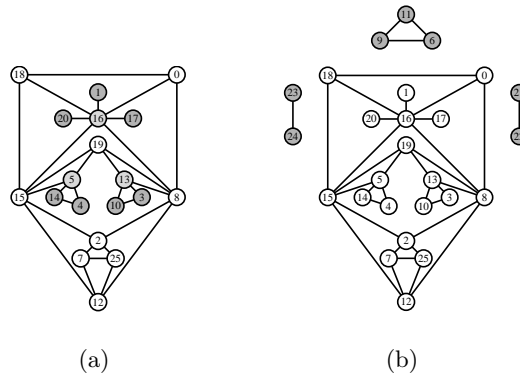


Fig. 23. Adding the remaining blocks (a) and connected components (b) of G' .

The triconnected graph corresponding to this R-node has no face that is fixed by the rotation displayed in Figs. 1(a) and 1(b), hence this rotation is not planar.