A Linear-Time Algorithm for Testing Outer-1-Planarity*,**

Seok-Hee Hong[1], Peter Eades[1], Naoki Katoh[2], Giuseppe Liotta[3], Pascal Schweitzer[4], and Yusuke Suzuki[5]

[1] University of Sydney, Australia
{seokhee.hong,peter.eades}@sydney.edu.au
[2] Kyoto University, Japan
naoki@archi.kyoto-u.ac.jp
[3] University of Perugia, Italy
liotta@diei.unipg.it
[4] ETH, Switzerland
pascal@mpi-inf.mpg.de
[5] Niigata University, Japan
y-suzuki@math.sc.niigata-u.ac.jp

Abstract. A graph is *1-planar* if it can be embedded in the plane with at most one crossing per edge. A graph is *outer-1-planar* if it has an embedding in which every vertex is on the outer face and each edge has at most one crossing. We present a linear time algorithm to test whether a graph is outer-1-planar. The algorithm can be used to produce an outer-1-planar embedding in linear time if it exists.

1 Introduction

A recent research topic in topological graph theory is the study of graphs that are *almost planar* in some sense. Examples of such almost planar graphs are *1-planar* graphs, which can be embedded in a plane with at most one crossing per edge.

Ringel [3] introduced 1-planar graphs in the context of simultaneously coloring vertices and faces of planar graphs. Subsequently, various aspects of 1-planar graphs have been investigated. Borodin [4] gives colouring methods for 1-planar graphs. Pach and Toth [5] prove that a 1-planar graph with n vertices has at most $4n - 8$ edges, which is a tight upper bound. There are a number of structural results on 1-planar graphs [6], and *maximal* 1-planar embeddings [7] (a 1-planar embedding of a graph G is *maximal*, if no edge can be added without violating the 1-planarity of G).

The class of 1-planar graphs is not closed under edge contraction; accordingly, computational problems seem difficult. Korzhik and Mohar proved that testing 1-planarity

* This paper is an extended abstract. For omitted proofs, see the full version of this paper [1]. The problem studied in this paper was initiated at the Port Douglas Workshop on Geometric Graph Theory, June, 2011, held in Australia, organized by Peter Eades and Seok-Hee Hong, supported by IPDF funding from the University of Sydney.

** Independently, another linear time algorithm is reported in [2].

S. Wismath and A. Wolff (Eds.): GD 2013, LNCS 8242, pp. 71–82, 2013.

of a graph is NP-complete [8]. On the positive side, it has been shown that the problem of testing maximal 1-planarity of a graph G can be solved in linear time if a *rotation system* (i.e., the circular ordering of edges for each vertex) is given by Eades et al [9].

The existence of a 1-planar embedding does not guarantee the existence of a straight-line 1-planar drawing, as shown by Eggleton [10] and Thomassen [11]. However, recently Hong et al. [12] give a linear time testing algorithm, and a linear time drawing algorithm to construct such a drawing if it exists. Very recently, the more general problem on straight-line drawability of embedded graphs is studied by Nagamochi [13].

Eggleton [10] introduced the investigation of *outer-1-planar graphs*: a graph is outer-1-planar if it has a 1-planar drawing in which every vertex is on the outer face. Examples of outer-1-planar graph drawings are shown in Fig. 1(a), (b), (c) and (d); in Fig. 1(e) a graph that has no outer-1-planar drawing is illustrated. Eggleton describes a number of geometric, topological, and combinatorial properties of outer-1-planar graphs.

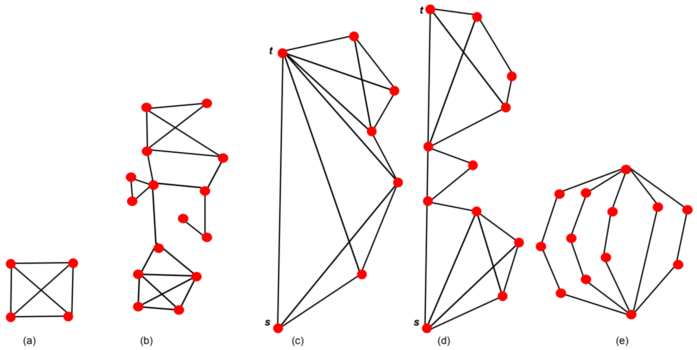

Fig. 1. *(a), (b), (c) and (d) are examples of outer-1-planar graph drawings. (a) illustrates the only triconnected outer-1-planar graph. (c) and (d) are examples of graphs that are one-sided outer-1-planar (OSO1P) with respect to* (s,t). *(d) illustrates a graph with no outer-1-planar drawing.*

In this paper, we investigate algorithmics of outer-1-planar graphs. More specifically, we describe a linear time algorithm to test outer-1-planarity of a given graph G.

Theorem 1. *There is a linear time algorithm to test whether a graph is outer-1-planar. The algorithm produces an outer-1-planar embedding if it exists.*

To prove Theorem 1, we define a sub-class of outer-1-planar graphs as follows. Suppose that G is a graph with vertices s and t. Let $G_{+(s,t)}$ denote the graph obtained by adding the edge (s,t), if this edge is not already in G. If $G_{+(s,t)}$ has an outer-1-planar embedding in which the edge (s,t) is completely on the outer face, then we say that G is *one-sided-outer-1-planar (OSO1P)* with respect to (s,t). Examples of such graphs are shown in Fig. 1(c) and (d). For these graphs, we prove the following result.

Theorem 2. *There is a linear time algorithm to test whether a biconnected graph G is one-sided outer-1-planar with respect to a given edge (s,t) of G. The algorithm produces a one-sided outer-1-planar embedding if it exists.*

Section 4 describes an algorithm to test whether a graph is one-sided outer-1-planar, and Section 5 shows how to use one-sided outer-1-planarity to test outer-1-planarity. The adaption of the algorithms of Sections 4 and 5 to construct an embedding is straightforward, and described in Section 6. In conclusion, Section 7 cites drawing algorithms and discusses future work.

2 Terminology

In this Section we define the terminology used throughout the paper.

A *topological graph* or *embedding* $G = (V, E)$ is a representation of a simple graph in the plane where each vertex is a point and each edge is a Jordan arc between the points representing its endpoints.

Two edges *cross* if they have a point in common, other than their endpoints. The point in common is a *crossing*. To avoid some pathological cases, some standard non-degeneracy conditions apply: (1) two edges intersect in at most one point; (2) an edge does not contain a vertex other than its endpoints; (3) no edge crosses itself; (4) edges must not meet tangentially; (5) no three edges share a crossing point; and (6) no two edges that share an endpoint cross.

A topological graph is *1-planar* if no edge has more than one crossing. A graph is *1-planar* if it has a 1-planar embedding. A graph is *outer-1-planar* if it has a 1-planar embedding in which every vertex is on the outer face. The aim of this paper is to give an algorithm to test whether a graph is outer-1-planar, and to provide an outer-1-planar embedding if it exists.

Our algorithm uses an *SPQR tree* to represent the decomposition of a biconnected graph into triconnected components. We recall some basic terminology of SPQR trees; for details, see [14]. Each node ν in the SPQR tree is associated with a graph called the *skeleton* of ν, denoted by $\sigma(\nu)$. There are four types of nodes ν in the SPQR tree: (1) S-nodes, where $\sigma(\nu)$ is a simple cycle with at least 3 vertices; (2) P-nodes, where $\sigma(\nu)$ consists of two vertices connected by at least 3 edges; (3) Q-nodes, where $\sigma(\nu)$ consists of two vertices connected by a real edges and a virtual edge; and (4) R-nodes, where $\sigma(\nu)$ is a simple triconnected graph. We treat the SPQR tree as a rooted tree by choosing an arbitrary node as its root. Note that every leaf is a Q-node and that the root is not a Q-node.

Let ρ be the parent of an internal node ν. The graph $\sigma(\rho)$ has exactly one *virtual edge* e in common with $\sigma(\nu)$; this is the *parent virtual edge* of $\sigma(\nu)$, and a *child virtual edge* in $\sigma(\rho)$. We denote the graph formed by the union of $\sigma(\nu)$ over all descendants ν of ρ by G_ρ.

If G is an outer-1-planar graph, then $\sigma(\nu)$ and G_ν are outer-1-planar graphs, using the embedding induced from G. If G_ν is a one-sided outer-1-planar (OSO1P) graph with respect to the parent virtual edge (s,t) of ν then we say that ν is a one-sided outer-1-planar (OSO1P) node with respect to (s,t).

For this paper, we need to define a specific type of S-node. Suppose that μ is an S-node with parent separation pair (u, v). A *tail at* u for μ is a Q-node child (that is, a real edge) with parent virtual edge (u, x) for some vertex x.

Further, we need to define a specific type of P-node. A P-node ν is *almost one-sided outer-1-planar (AOSO1P) with respect to (the directed edge)* (s, t) if G_ν consists of a parallel composition of an OSO1P graph with respect to (s, t) and an S-node μ such that μ has a tail at t and μ is OSO1P with respect to (s, t). See Fig. 2 for examples.

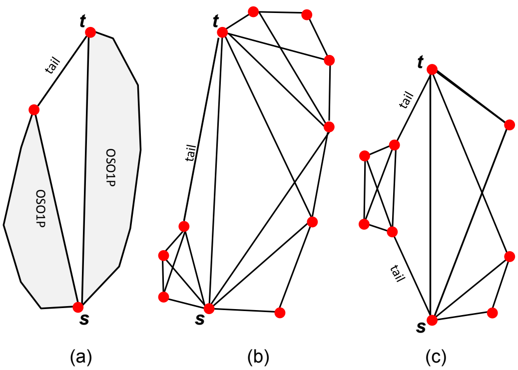

Fig. 2. *An AOSO1P graph consists of a parallel composition of an OSO1P graph and an OSO1P S-node with a tail. (a) The general shape of a graph that is AOSO1P with respect to* (s, t)*. (b) A graph that is AOSO1P with respect to* (s, t)*. (c) A graph that is AOSO1P with respect to both* (s, t) *and* (t, s)*.*

3 Structural Results

In this Section we present structural results that support the algorithms defined in the subsequent sections.

First we note that the only triconnected outer-1-planar graph is K_4, embedded as depicted in Fig. 1(a).

Lemma 1. *If* G *is outer-1-planar and triconnected, then* G *is isomorphic to* K_4 *and every outer-1-planar drawing of* G *has exactly one crossing.*

Proof. Suppose that G is an outer-1-planar embedding of a triconnected graph; we can assume that G is maximal in the sense that no edge can be added without destroying the property of outer-1-planarity. Eggleton [10] shows that the outer face is a simple cycle γ.

Suppose that (a, b) and (c, d) are a pair of edges, neither on γ, that cross at point p. Suppose that a precedes c in clockwise order around γ. Suppose that there is at least one vertex v that lies between a and c on γ, as shown in Fig. 3.

All edges incident with the vertices between a and c on γ must have both endpoints in the region r bounded by γ, the curve ap and the curve cp. Removing a and c separates v from the remainder of the graph; this contradicts the triconnectivity of G, and we can deduce that there is no such vertex v.

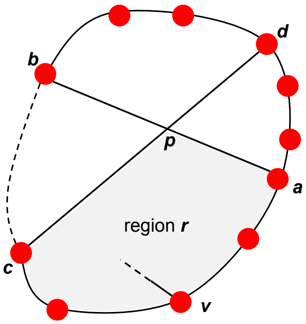

Fig. 3. *Here the pair a, c must be a separation pair, since all edges incident with vertices between a and c on the outer face γ must have both endpoints in the region r.*

Using the same argument, we can show that the only vertices on γ are a, b, c and d; thus G is K_4. □

Secondly, we note that we can restrict our attention to the biconnected case.

Lemma 2. *A graph is outer-1-planar if and only if its biconnected components are outer-1-planar.*

Next we present a simple fact about how an edge can cross a cycle in an outer-1-planar embedding.

Lemma 3. *Suppose that γ is a cycle in an outer-1-planar graph G and G' is an outer-1-planar embedding of G. Suppose that an edge (u, v) is not on γ but crosses an edge of γ in G'. Then either u or v is on γ.*

The next result is a relatively technical but fundamental Lemma about embeddings of paths which share endpoints, illustrated in Fig. 4. A path from a vertex s to a vertex t is *non-trivial* if it contains more than two vertices. If an edge from a path p_1 crosses an edge from a path p_2 then we say that p_1 *crosses* p_2.

Lemma 4. *Suppose that P is a set of paths between two vertices s and t. Let G be the union of the paths in P, and let G' be an outer-1-planar embedding of G. Then $|P| \leq 5$, and:*

(a) If $|P| \geq 3$ and an edge from one non-trivial path $p_1 \in P$ crosses an edge from another non-trivial path $p_2 \in P$ then this crossing occurs between an edge incident with s and an edge incident with t.
(b) If $|P| = 3$ and all paths in P are non-trivial, then there are two paths p_1 and p_2 in P such that there is exactly one crossing between edges of p_1 and edges of p_2; furthermore, every edge in the third path is on the outer face.
(c) If $|P| = 3$ and one path in P is trivial and is on the outer face, then there are two paths p_1 and p_2 in P such that there is exactly one crossing between edges of p_1 and edges of p_2.

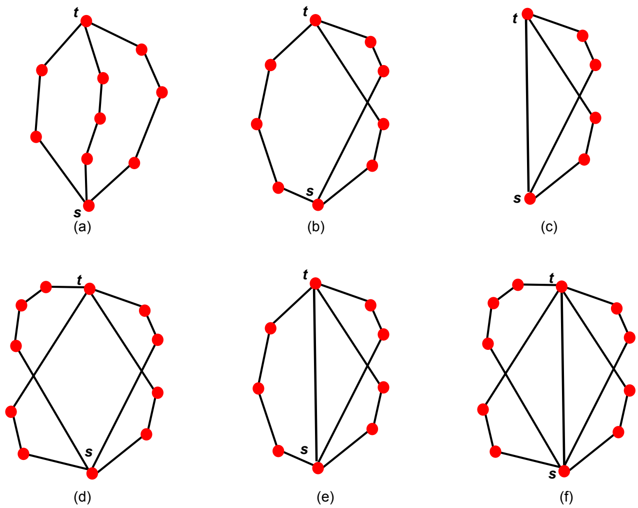

Fig. 4. *Embeddings of paths that share endpoints. (a) A planar embedding. (b) An outer-1-planar embedding of 3 non-trivial paths. (c) An outer-1-planar embedding of 3 paths, where one path is trivial. (d) An outer-1-planar embedding of 4 non-trivial paths. (e) An outer-1-planar embedding of 4 paths, where one path is trivial. (f) An outer-1-planar embedding of 5 paths.*

(d) If P contains 4 non-trivial paths, then we can divide P into two pairs of paths $\{p_1, p_2\}$ and $\{p_3, p_4\}$ such that there is exactly one crossing between edges of p_1 and edges of p_2 and exactly one crossing between edges of p_3 and edges of p_4, and there are no other crossings.
(e) If $|P| = 4$ and P contains a trivial path, then there are two paths p_1 and p_2 in P such that there is exactly one crossing between edges of p_1 and edges of p_2, and one non-trivial path in P is on the outer face.
(f) If $|P| = 5$ then one path in P is trivial, and we can divide the other paths into two pairs of paths $\{p_1, p_2\}$ and $\{p_3, p_4\}$ such that there is exactly one crossing between edges of p_1 and edges of p_2 and exactly one crossing between edges of p_3 and edges of p_4, and there are no other crossings.

4 Testing OSO1P and AOSO1P

In this Section we describe a linear time algorithm that takes a graph G and vertices s and t of G as input and tests whether G has an OSO1P or AOSO1P embedding with respect to (s, t).

From Lemma 2, we only need to consider biconnected graphs. Our algorithm computes the SPQR tree T and then works from the leaves of T upward toward the root, computing boolean labels $OSO1P(\nu, s, t)$ and $AOSO1P(\nu, s, t)$ that indicate whether

ν is OSO1P or AOSO1P with respect to (s,t). The label $OSO1P(\nu,s,t)$ is computed for each node ν of T, and the label $AOSO1P(\nu,s,t)$ is computed for every P-node ν.

Algorithm test-One-Sided-Outer-1-Planar

1. Construct the SPQR tree T of G.
2. Traverse T bottom up, and for each node ν of T with parent virtual edge (s,t):
   (a) `if` ν is a Q-node `then return true`.
   (b) `elseif` ν is an R-node `then return` $OSO1P(\nu,s,t)$ as described in Section 4.1, using the values $OSO1P(\nu',s',t')$ for each child ν' of ν with child virtual edge (s',t').
   (c) `elseif` ν is a P-node `then return` $OSO1P(\nu,s,t)$ and $AOSO1P(\nu,s,t)$, as described in Section 4.2, using the values $OSO1P(\nu',s',t')$ for each child ν' of ν with child virtual edge (s',t').
   (d) `else` $/*$ ν is an S-node $*/$ `then return` $OSO1P(\nu,s,t)$, as described in Section 4.3, using the values $OSO1P(\nu',s',t')$ and $AOSO1P(\nu,s',t')$ for each child ν' of ν with child virtual edge (s',t').

The time complexity of Step 1 is linear [14,15]. Step 2(a) is trivial and takes constant time for each Q-node. We show below that Steps 2(b), 2(c), and 2(d) each take time proportional to the number of children of the node ν. Summing over all nodes of the SPQR tree T results in linear time for the whole algorithm.

The cases for R-nodes and P-nodes are quite straightforward, and we deal with them first in Sections 4.1 and 4.2. The extension for the computation of the AOSO1P property for P-nodes is again straightforward and described in Section 4.2. The case for S-nodes is a little more involved, and we deal with this in Section 4.3.

4.1 R-nodes

Lemma 1 can be generalised to R-nodes as in the following Lemma.

Lemma 5. *Suppose that ν is an R-node of the SPQR tree of a biconnected graph G; suppose that (u,v) is its parent virtual edge. Then G_ν is OSO1P with respect to (u,v) if and only if:*

1. *$\sigma(\nu)$ is isomorphic to K_4; and*
2. *an edge (u,a) of $\sigma(\nu)$ with $a \neq v$ incident with u represents a child Q-node of ν, an edge (v,b) of $\sigma(\nu)$ with $b \neq u$ represents a child Q-node of ν, and (u,a) crosses (v,b); and*
3. *for every child ν' of ν, ν' is OSO1P with respect to (c,d), where the parent virtual edge of ν' is (c,d).*

An algorithm to test whether an R-node ν is OSO1P can be derived directly from Lemma 5. It is clear that, given the boolean labels $OSO1P(\nu_i,s,t)$ for each child ν_i of ν, the algorithm runs in time proportional to the number of children of ν.

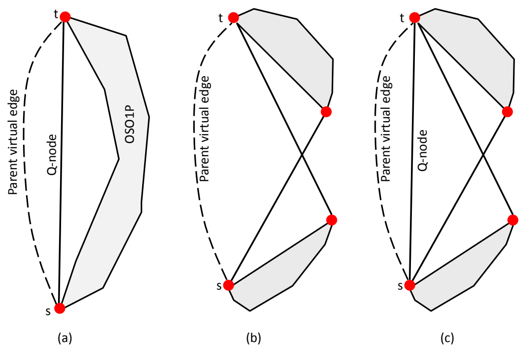

Fig. 5. *Possibilities for an OSO1P P-node*

4.2 P-nodes

Here we use Lemma 4 to show that a P-node can have at most three children, as depicted in Fig. 5. More specifically, we have the following Lemma:

Lemma 6. *Suppose that ν is a P-node of the SPQR tree of a biconnected graph G; suppose that (s, t) is its parent virtual edge. Then G_ν is OSO1P with respect to (s, t) if and only if either:*

(a) ν has two children, of which one is a Q-node (s, t), and the other is OSO1P with respect to (s, t); or
(b) ν has two children, of which one is an S-node with tail at s which is OSO1P with respect to (s, t), and the other is an S-node with tail at t which is OSO1P with respect to (s, t); or
(c) ν has three children, of which one is a Q-node (s, t), one is an S-node with tail at s which is OSO1P with respect to (s, t), and the other is an OSO1P S-node with tail at t which is OSO1P with respect to (s, t).

It is straightforward to extend Lemma 6 to test whether a node ν is AOSO1P, using the definition of AOSO1P together with Lemma 6.

Algorithms to test whether a P-node ν is OSO1P or AOSO1P can be derived directly from Lemma 6. It is clear that, given the boolean labels $OSO1P(\nu', s, t)$ for each child ν' of ν, the algorithm runs in time proportional to the size of the skeleton $\sigma(\nu)$ of ν.

4.3 S-nodes

Suppose that ν is an S-node with children $\nu_1, \nu_2, \ldots, \nu_k$, where the parent virtual edge of ν_i is (s_{i-1}, s_i), as shown in Fig. 6(a).

If every child ν_i is OSO1P with respect to (s_{i-1}, s_i), then clearly ν is OSO1P with respect to (s_0, s_k); however, the converse is false. Consider the example shown in Fig. 6(b). Here ν is OSO1P with respect to (s_0, s_k). However, the child ν_2 is not OSO1P

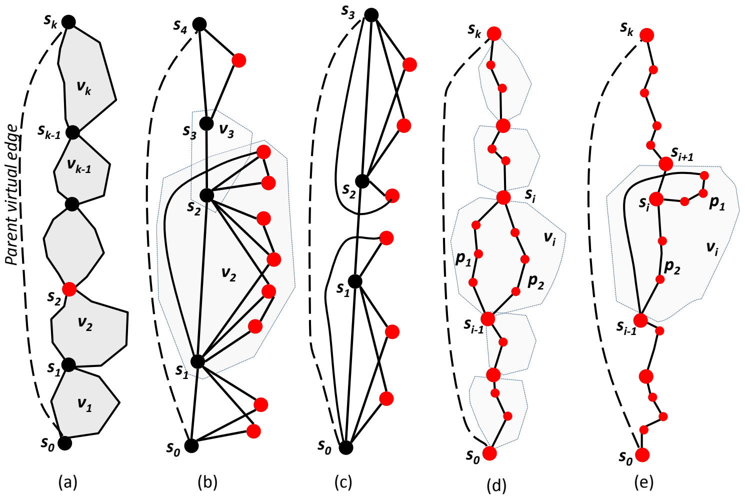

Fig. 6. *(a) An S-node. (b) An OSO1P S-node with a child (ν_2) that is not OSO1P. (c) An S-node that satisfies the conditions of Lemma 7 but is not OSO1P. (d) Two paths p_1 and p_2 in the graph G_{ν_i}. (e) The path p_1 crosses the edge (s_i, s_{i+1}).*

with respect to (s_1, s_2) (by Lemma 6). Note that ν_3 is a Q-node, and an edge from the skeleton of ν_2 crosses this edge. In fact the example shown in Fig. 6(b) illustrates the necessary conditions stated in the next lemma.

Lemma 7. *Suppose that ν is an S-node with children $\nu_1, \nu_2, \ldots, \nu_k$, where the parent virtual edge of ν_i is (s_{i-1}, s_i). Suppose that G_ν is OSO1P with respect to (s_0, s_k). Then for $1 \le i \le k$, either:*

(a) ν_i is OSO1P with respect to (s_{i-1}, s_i); or
(b) $i < k$, ν_i is AOSO1P with respect to (s_i, s_{i-1}), and ν_{i+1} is a Q-node; or
(c) $i > 1$, ν_i is AOSO1P with respect to (s_{i-1}, s_i), and ν_{i-1} is a Q-node.

Lemma 7 gives necessary conditions for an S-node to be OSO1P. However the conditions are not sufficient. Consider, for example, the graph shown in Fig. 6(c). This satisfies the necessary conditions as in Lemma 7, but it is not OSO1P. The problem is that the Q-node represented by the edge (s_1, s_2) has two crossings, one with an edge of the AOSO1P graph at the top and one with an edge of the AOSO1P graph at the bottom. Nevertheless, this situation does not occur when $k = 2$, and we shall show that in this case the conditions of Lemma 7 are sufficient. One can express sufficient conditions for an S-node to be OSO1P in a recursive way, as in the following Lemma. If ν is an S-node with children $\nu_1, \nu_2, \ldots, \nu_k$ then we denote the series combination of graphs $G_{\nu_1}, G_{\nu_2}, \ldots, G_{\nu_k}$ by $G(\nu_1, \nu_2, \ldots, \nu_k)$.

Lemma 8. *Suppose that ν is an S-node with children $\nu_1, \nu_2, \ldots, \nu_k$, where the parent virtual edge of ν_i is (s_{i-1}, s_i). Then G_ν is OSO1P with respect to (s_0, s_k) if and only if either:*

1. *G_{ν_1} is OSO1P with respect to (s_0, s_1) and $G(\nu_2, \nu_3, \ldots, \nu_k)$ is OSO1P with respect to (s_1, s_k); or*
2. *ν_1 is a Q-node, G_{ν_2} is AOSO1P with respect to (s_1, s_2), and $G(\nu_3, \nu_4, \ldots, \nu_k)$ is OSO1P with respect to (s_2, s_k); or*
3. *G_{ν_1} is AOSO1P with respect to (s_1, s_0), ν_2 is a Q-node, and $G(\nu_3, \nu_4, \ldots, \nu_k)$ is OSO1P with respect to (s_2, s_k).*

Lemma 8 leads to the recursive algorithm for S-nodes; see [1]. The algorithm runs in time proportional to the number of children of ν.

This completes the proof of Theorem 2.

5 Testing Outer-1-Planarity

Once we compute the labels $OSO1P(\nu, s, t)$ and $AOSO1P(\nu, s, t)$ for all internal nodes ν of the SPQR tree, we can test whether the whole graph (that is, the root ρ) is outer-1-planar. This requires separate tests depending on the type of the root node. See Fig. 7.

We can require the root node to be an R-node or a P-node, since if the SPQR tree contains no R-node and no P-node, then the graph is a cycle and thus outerplanar. Both tests for an R-node and a P-node are detailed below.

For R-nodes, we have the following Lemma.

Lemma 9. *Suppose that ρ is an R-node at the root of the SPQR tree. Then G is outer-1-planar if and only if*

1. *$\sigma(\rho)$ is isomorphic to K_4, and*
2. *at least two children of ρ are Q-nodes, and*
3. *for every child node ν' of $\sigma(\rho)$ with parent virtual edge (a, b), $G_{\nu'}$ is OSO1P with respect to (a, b).*

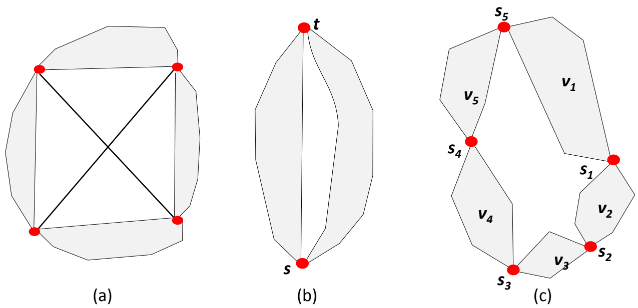

Fig. 7. *(a) An R-node at the root. (b) P-node at the root. (c) S-node at the root.*

It is clear that one can test the conditions of Lemma 9 in constant time, as long as the labels $OSO1P(\nu', a, b)$ for all children ν' of ρ have already been computed.

For P-nodes, the following result applies.

Lemma 10. *Suppose that ρ is a P-node at the root of the SPQR tree. Then G is outer-1-planar if and only if it is a parallel composition of two OSO1P graphs.*

Using Lemma 4, one can show that the number of children of a P-node ρ at the root is bounded (in fact, at most 5). It follows that the number of ways to partition the children is bounded by a constant. Thus we can define a constant time algorithm to implement Lemma 10 (given that the labels $OSO1P(\nu', a, b)$ for all children ν' of ρ have already been computed).

This completes the proof of Theorem 1.

6 Outer-1-Planar Embedding

One can construct a one-sided outer-1-planar embedding of an input graph G using an extension of the methods in Section 4. The methods for R-nodes and S-nodes described in Lemmas 5 and 6 define crossings; treating these crossings as dummy vertices gives a planar graph G^*. A one-sided outer-1-planar embedding of G is a specific planar embedding of G^*.

Every planar embedding of G^* is defined by an orientation and an ordering for nodes ν in the SPQR tree with respect to the parent separation pair of ν. For P-nodes, R-nodes, and S-nodes, it is possible to "flip" the orientation of ν around its parent separation pair. For P nodes, a left-right order for the children can be chosen. To produce an outer-1-planar embedding we use the same bottom-up strategy as in **Algorithm test-One-Sided-Outer-1-Planar** in Section 4. Throughout the algorithm we maintain an embedding; in particular we keep track of the outside face. At each node ν, we "flip" ν so that all vertices of G_ν lie on the outside face. Also, at each P-node ν, we order the children of ν so that all vertices of G_ν lie on the outside face. This requires linear time manipulation of the SPQR tree, using methods outlined in [14].

7 Conclusion

The algorithm presented in this paper takes a graph G as input and determines whether it has an outer-1-planar embedding. We show that if such an embedding does exist, then we can compute it in linear time.

Given the topological embedding computed by our algorithm, the methods of Eggleton [10] can be used to construct a straight-line drawing. In fact, Eggleton gives conditions that determine whether a given set of points support a straight-line outer-1-planar drawing. Dekhordi et al. [16] show further that every outer-1-planar topological embedding has a straight-line RAC (right-angle crossing) drawing, at the cost of exponential area.

Many algorithms for drawing outerplanar graphs exist, with a number of properties (see [17,18]). It would be interesting to see if these results extend to outer-1-planar graphs.

References

1. Hong, S.H., Eades, P., Katoh, N., Liotta, G., Schweitzer, P., Suzuki, Y.: A linear-time algorithm for testing outer-1-planarity. TR-IT-IVG-2013-01. Technical Report, School of IT, University of Sydney (June 2013)
2. Auer, C., Bachmaier, C., Brandenburg, F.J., Gleißner, A., Hanauer, K., Neuwirth, D., Reislhuber, J.: Recognizing outer 1-planar graphs in linear time. In: Wismath, S., Wolff, A. (eds.) GD 2013. LNCS, vol. 8242, pp. 107–118. Springer, Heidelberg (2013)
3. Ringel, G.: Ein Sechsfarbenproblem auf der Kugel. Abh. Math. Sem. Univ. Hamburg 29, 107–117 (1965)
4. Borodin, O.V.: Solution of the Ringel problem on vertex-face coloring of planar graphs and coloring of 1-planar graphs. Metody Diskret. Analiz. 41, 12–26 (1984)
5. Pach, J., Tóth, G.: Graphs drawn with few crossings per edge. Combinatorica 17, 427–439 (1997)
6. Fabrici, I., Madaras, T.: The structure of 1-planar graphs. Discrete Mathematics 307, 854–865 (2007)
7. Suzuki, Y.: Re-embeddings of maximum 1-planar graphs. SIAM J. Discrete Math. 24, 1527–1540 (2010)
8. Korzhik, V.P., Mohar, B.: Minimal obstructions for 1-immersions and hardness of 1-planarity testing. Journal of Graph Theory 72, 30–71 (2013)
9. Eades, P., Hong, S.-H., Katoh, N., Liotta, G., Schweitzer, P., Suzuki, Y.: Testing maximal 1-planarity of graphs with a rotation system in linear time - (extended abstract). In: Didimo, W., Patrignani, M. (eds.) GD 2012. LNCS, vol. 7704, pp. 339–345. Springer, Heidelberg (2013)
10. Eggleton, R.: Rectilinear drawings of graphs. Utilitas Mathematica 29, 149–172 (1986)
11. Thomassen, C.: Rectilinear drawings of graphs. Journal of Graph Theory 12, 335–341 (1988)
12. Hong, S.H., Eades, P., Liotta, G., Poon, S.H.: Fáry's theorem for 1-planar graphs. In: [19], pp. 335–346
13. Nagamochi, H.: Straight-line drawability of embedded graphs. Technical Report 2013-005, Department of Applied Mathematics and Physics, Kyoto University, Japan (2013)
14. Battista, G.D., Tamassia, R.: On-line maintenance of triconnected components with spqr-trees. Algorithmica 15, 302–318 (1996)
15. Gutwenger, C., Mutzel, P.: A Linear Time Implementation of SPQR-Trees. In: Marks, J. (ed.) GD 2000. LNCS, vol. 1984, pp. 77–90. Springer, Heidelberg (2001)
16. Dehkordi, H.R., Eades, P.: Every outer-1-plane graph has a right angle crossing drawing. Int. J. Comput. Geometry Appl. 22, 543–558 (2012)
17. Frati, F.: Straight-line drawings of outerplanar graphs in o(dn log n) area. Comput. Geom. 45, 524–533 (2012)
18. Knauer, K.B., Micek, P., Walczak, B.: Outerplanar graph drawings with few slopes. In: [19], pp. 323–334
19. Gudmundsson, J., Mestre, J., Viglas, T. (eds.): COCOON 2012. LNCS, vol. 7434. Springer, Heidelberg (2012)