Recognizing Outer 1-Planar Graphs in Linear Time*,**

Christopher Auer, Christian Bachmaier, Franz J. Brandenburg,
Andreas Gleißner, Kathrin Hanauer, Daniel Neuwirth, and Josef Reislhuber

University of Passau, 94030 Passau, Germany
{auerc,bachmaier,brandenb,gleissner,hanauer,neuwirth,reislhuber}
@fim.uni-passau.de

Abstract. A graph is outer 1-planar (*o1p*) if it can be drawn in the plane such that all vertices are on the outer face and each edge is crossed at most once. *o1p* graphs generalize outerplanar graphs, which can be recognized in linear time and specialize 1-planar graphs, whose recognition is $\mathcal{NP}$-hard.
Our main result is a linear-time algorithm that first tests whether a graph G is *o1p*, and then computes an embedding. Moreover, the algorithm can augment G to a maximal *o1p* graph. If G is not *o1p*, then it includes one of six minors (see Fig. 3), which are also detected by the recognition algorithm. Hence, the algorithm returns a positive or negative witness for *o1p*.

1 Introduction

Planar graphs are one of the most studied areas in graph theory and an important class in graph drawing. Outerplanar graphs are in turn an important subfamily of planar graphs. Here, all vertices are on the outer face and edges do not cross. Every outerplanar graph has at least two vertices of degree two, which is used for a recognition in linear time [16].

There were several attempts to generalize planarity to graphs that are "almost" planar in some sense. Such attempts are important as many graphs are not planar. One generalization are 1-planar graphs, which were introduced by Ringel [17] in an approach to color a planar graph and its dual. A graph is 1-planar if it can be drawn in the plane such that each edge is crossed at most once and incident edges do not cross. 1-planar graphs are a hot topic in graph drawing, see also [1, 4–6, 8, 9, 13, 15].

The combination of 1-planarity and outerplanarity leads to *o1p* graphs, which are graphs with an embedding in the plane with all vertices on the outer face and at most one crossing per edge. They were introduced by Eggleton [10] who called

* This work was supported in part by the Deutsche Forschungsgemeinschaft (DFG) grant Br835/18-1.

** A linear-time algorithm for testing outer 1-planarity was independently obtained by Hong et al. and appears in these proceedings [14].

them outerplanar graphs with edge crossing number one. He showed that edges of maximal *o1p* graphs do not cross in the outer face and each face is incident to at most one crossing, from which he concluded that every *o1p* graph has an *o1p* drawing with straight-line edges and convex (inner) faces. Thomassen [18] generalized Eggleton's result and characterized the class of 1-planar graphs which admit straight-line drawings by the exclusion of so-called B- and W-configurations in embeddings. These configurations were rediscovered by Hong et al. [13], who also provide a linear-time drawing algorithm that starts from a given embedding.

From the algorithmic perspective there is a big step from zero to some crossings. It is well-known that planar graphs can be recognized in linear time, and there are linear-time algorithms to construct an embedding and drawings, e. g., straight-line drawings and visibility representations in quadratic area. On the contrary, dealing with crossings generally leads to $\mathcal{NP}$-hard problems. It is $\mathcal{NP}$-hard to recognize 1-planar graphs [15], even if the graph is given with a rotation system, which determines the cyclic ordering of the edges at each vertex [2]. 1-planarity remains $\mathcal{NP}$-hard even if the treewidth is bounded [3]. There also is no efficient algorithm to compute the crossing number of a graph [12] and to compute the number of crossings induced by the insertion of an edge into a planar graph [5]. However, there is a linear-time recognition algorithm of Eades et al. [8] for maximal 1-planar graphs, which needs a given rotation system.

In this paper we study *o1p* graphs. Our main result is a linear-time recognition algorithm for *o1p* graphs. This is the first efficient algorithm for a test of 1-planarity that takes solely a graph as input. Our recognition algorithm is based on SPQR-trees. It analyzes its nodes and then either computes an *o1p* embedding or detects one of six minors. If the graph is *o1p*, it can be augmented to a maximal *o1p* graph. In a maximal *o1p* graph, adding a new edge violates its defining property. From the structure of a maximal *o1p* graph we derive that every *o1p* graph is planar. Thus, they are subgraphs of planar graphs with a Hamiltonian cycle, and the SPQR-tree reveals a treewidth of at most three.

2 Preliminaries

We consider simple, undirected graphs $G = (V, E)$ with n vertices and m edges. The graphs are biconnected, otherwise, the components are treated separately. A *drawing* of a graph is a mapping of G into the plane such that the vertices are mapped to distinct points and each edge is a Jordan arc between its endpoints. A drawing is *planar* if the Jordan arcs of the edges do not cross and it is *1-planar* if each edge is crossed at most once. Accordingly, a graph is planar (1-planar) if it has a planar (1-planar) drawing. Crossings of edges with the same endpoint, i. e., *incident* edges, are excluded. A planar drawing of a graph partitions the plane into *faces*. A face is specified by a cyclic sequence of edges that forms its boundary. The set of all faces forms the *embedding* of the graph. In 1-planar drawings, every crossing divides an edge into two *edge segments*. An uncrossed edge consists of one segment. Therefore, a face of a *1-planar embedding* is specified by a cyclic list of edge segments.

A graph G is *outerplanar* if it has a planar drawing with all vertices on one distinguished face. This face is referred to as the *outer face* and corresponds to the unbounded, external face in a drawing on the plane. G is *maximal* outerplanar if no further edge can be added without violating outerplanarity. Then, the edges on the outer face form a Hamiltonian cycle. A graph G is *outer 1-planar*, *o1p* for short, if it has a drawing with all vertices on the outer face and such that each edge is crossed at most once. G is *maximal o1p* if the addition of any edge violates outer 1-planarity.

In an *o1p* embedding, an edge is either *crossing* or *plane* (*non-crossing*). We say that it is *inner*, if none of its segments is part of the boundary of the outer face. Analogously, an edge is *outer*, if it is entirely part of this boundary. Observe that a crossed edge cannot be outer. If the embedding is maximal, we can classify every edge as *outer* or *inner*.

Maximal outerplanar graphs have a unique embedding. This does no longer hold for maximal *o1p* graphs. Consider a graph with 6 vertices and 11 edges consisting of two K_4s. If the left K_4 is fixed, the right can be flipped. In order to gain more insight into the structure of an *o1p* graph G, we consider its *SPQR-tree* $\mathcal{T}$. SPQR-trees were first introduced by Di Battista and Tamassia [7] and provide a description of how the graph is composed. Fig. 2(a) depicts an example graph along with its SPQR-tree in Fig. 2(b). In the definition we adopt here, the SPQR-tree is unrooted. The nodes of $\mathcal{T}$ either represent a series composition (S), a parallel composition (P), a single edge (Q), or a triconnected component (R). Associated with each node μ of $\mathcal{T}$ is a graph that is homeomorphic to a subgraph of G and called the *skeleton* $\mathrm{skel}(\mu)$ of μ. In its original definition, every edge $e = \{u, v\}$ of a skeleton, except for one of each Q-node, is a *virtual edge*, i. e., an edge that represents the subgraph of G which connects u and v. This subgraph is also referred to as the *expansion graph* $\mathrm{expg}(e)$ of e. For every virtual edge e in the skeleton of a node μ, there is another node ν that refines the structure of $\mathrm{expg}(e)$. We say that ν is the *refining* node $\mathrm{refn}(e)$ of e. This link is represented by an edge between μ and ν in $\mathcal{T}$ and we say that μ and ν are *adjacent* in $\mathcal{T}$. Therefore, every leaf of $\mathcal{T}$ is a Q-node. For simplification, we represent edges of the graph directly in the skeleton of an S-, P-, or R-node, so that we can neglect Q-nodes. We also call these edges *non-virtual*. Observe that all nodes are always as large as possible, so neither two S-nodes nor two P-nodes may be adjacent. For a more detailed introduction to SPQR-trees, the reader is referred to [7].

3 Recognition

There are linear-time algorithms for the recognition of (maximal) outerplanar graphs, that use the fact that there are at least two vertices of degree two. A single K_4 implies that this property no longer holds for *o1p* graphs. In contrast, the recognition of 1-planar graphs is $\mathcal{NP}$-hard [15], even if the graphs are given with a rotation system [2].

Algorithm 1 *o1p* Recognition

procedure TestOuter1Planarity(G)
  if G is not planar **then return** $\perp$
  $\mathcal{T} \leftarrow$ SPQR-tree of G
  for all R- and P-nodes $\mu \in \mathcal{T}$ **do**
    if μ is R-node **then**
      if skel(μ) $\neq K_4$ **or** contains vertex incident to > 2 virtual edges **then**
        return $\perp$ ▷ Lemma 1, Corollary 1
      for all neighbors ν of μ **do**
        if ν is S-node or R-node **then** insert plane edge ▷ Proposition 1
    else if μ is P-node **then**
      if skel(μ) contains > 4 virtual edges **then return** $\perp$ ▷ Corollary 1
      else if μ has only virtual edges **then** insert plane edge ▷ Lemma 4
  compute mapping $\mathcal{C}$
  $\mathbb{P}_F \leftarrow$ {fixable P-nodes} ; $\mathbb{P}_N \leftarrow$ {P-nodes with crossings, but none fixable}
  while $\mathbb{P}_F \cup \mathbb{P}_N \neq \emptyset$ **do**
    while $\mathbb{P}_F \neq \emptyset$ **do**
      remove next P-node π from $\mathbb{P}_F$ with fixable S-nodes σ_1, σ_2
      $z \leftarrow$FixCrossingAtPNode(G, $\mathcal{T}$, π, σ_1, σ_2)
      if $z = \perp$ **then return** $\perp$
      for all $\pi' \in z$ **do** update $\mathcal{C}$
        if π' is fixable **then** move π' from $\mathbb{P}_N$ to $\mathbb{P}_F$
    if $\mathbb{P}_N \neq \emptyset$ **then** ▷ Lemma 5
      choose any element π of $\mathbb{P}_N$ with S-nodes σ_1, σ_2 conformant to $\mathcal{C}$
      $z \leftarrow$ FixCrossingAtPNode(G, $\mathcal{T}$, π, σ_1, σ_2)
      for all $\pi' \in z$ **do** update $\mathcal{C}$
        if π' is fixable **then** move π' from $\mathbb{P}_N$ to $\mathbb{P}_F$
  for all S-/P-/R-nodes $\mu \in \mathcal{T}$ **do** fix embedding
  return 2-clique-sum of skeleton embeddings

Theorem 1. *There is a linear-time algorithm to test whether a biconnected graph G is o1p and, if so, returns an embedding.*

We prove this theorem by first establishing a number of necessary conditions for a graph to have an *o1p* embedding. At the same time, we implement a linear-time algorithm (Algorithm 1) that checks these conditions and, if positive, constructs an *o1p* embedding of the input graph. The algorithm starts by ensuring that the input graph is planar (cf. Corollary 4) and computes its SPQR-tree. Both subroutines take $\mathcal{O}(n)$ time [11]. Observe that, although the graph will be augmented during the next steps, it remains *o1p* and therefore also planar. Consequently, the number of nodes in $\mathcal{T}$ always is in $\mathcal{O}(n)$. In a second step, we show that the conditions are not only necessary, but also sufficient.

We start with two observations regarding *o1p* embeddings. For maximal 1-planar embeddings, a well-known fact is that every crossing induces a K_4. This holds in an even tightened form for *o1p* embeddings:

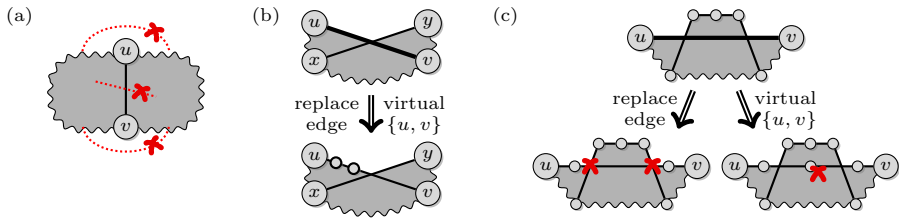

Fig. 1: (a): Proposition 2, (b) and (c): Proposition 3

Proposition 1 ([6]). *Let $\{a, b\}$ and $\{c, d\}$ be a pair of crossing edges in an o1p embedding of a maximal o1p graph. Then the vertices a, b, c, and d form a K_4 and the edges $\{a, b\}$, $\{b, c\}$, $\{c, d\}$, and $\{d, a\}$ are plane.*

Consider a plane, inner edge $\{u, v\}$ in an *o1p* embedding of a graph G. Then, $\{u, v\}$ "partitions" the embedding and the deletion of u and v disconnects G (cf. Fig. 1(a)).

Proposition 2. *Every plane, inner edge in an o1p embedding connects a separation pair.*

Let $\mathcal{T}$ be the SPQR-tree of an *o1p* graph G.

Lemma 1. *The skeleton of every R-node is a K_4.*

Proof. Recall that outerplanar graphs are series-parallel. Hence, the SPQR-tree of an outerplanar graph has no R-nodes. Let μ be an R-node in $\mathcal{T}$. Then, skel(μ) must be embedded such that at least two edges cross, e. g., edges $\{a, b\}$ and $\{c, d\}$. By Proposition 1, the vertices a, b, c, and d form a K_4.
There must be an embedding of skel(μ) such that all vertices are on the boundary of the same face. Suppose skel(μ) has more than four vertices. Then, at least one of $\{a, b\}$, $\{b, c\}$, $\{c, d\}$, and $\{d, a\}$ is an inner edge. By Proposition 1, all of them are plane. As an inner edge cannot be virtual, by Proposition 2, skel(μ) has a separation pair, so skel(μ) is not triconnected, a contradiction. □

Instead of considering the possible embeddings of G on the whole, we study those of the skeletons of the nodes in $\mathcal{T}$. As G is *o1p*, there must be an embedding of every skeleton of $\mathcal{T}$ such that the 2-clique-sums over all skeletons result in an *o1p* embedding of G. In short, a *2-clique-sum* combines two graphs by selecting an edge (2-clique) in each one and glueing them together at those edges. The selected edges are removed from the new graph. If the input graphs were embedded, the embedding is inherited for the 2-clique-sum such that in each case the other graph takes the position of the removed edge.

Consequently, we need to derive properties of *o1p* embeddings of skeletons. Like in usual *o1p* embeddings, there must be a face (the outer face) such that all vertices lie on its boundary. However, as virtual edges represent entire subgraphs, they demand special attention. Observe that the expansion graph of every virtual

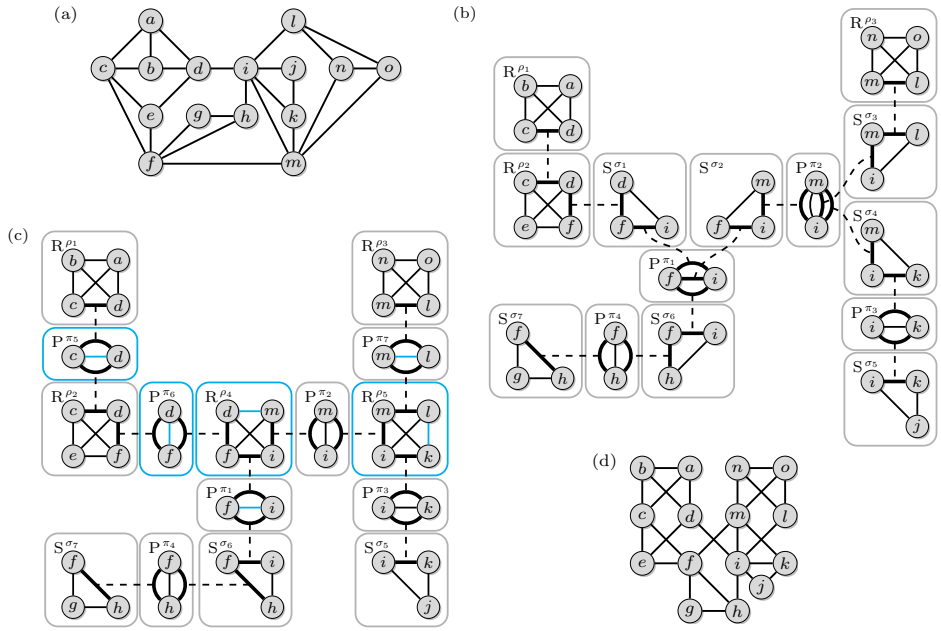

Fig. 2: Input graph (a), its SPQR-tree (b), the SPQR-tree after the algorithm (c) (new edges and nodes colored), and the found *o1p* embedding (d).

edge contains, besides the separation pair, at least one more vertex. Consider the virtual edge $\{u, v\}$ in Fig. 1(b). The crossing edge $\{x, y\}$ partitions $\{u, v\}$ into two segments, hence, $\text{expg}(\{u, v\})$ must be embedded such that it replaces the edge segment of $\{u, v\}$ that lies on the outer face. Suppose a virtual edge e is embedded such that it has at least two crossings. Then there is either an edge in $\text{expg}(e)$ that is crossed more than once or at least one vertex is enclosed between two crossings and hence does not lie on the outer face (cf. Fig. 1(c)).

Proposition 3. *Every virtual edge may consist of at most two edge segments and the embedding must be such that at least one segment is part of the boundary of the outer face.*

Observe that in contrast to the *o1p* embedding of the whole graph, we must allow the crossing of two virtual edges with a common end vertex in the embedding of a skeleton. We qualify the virtual edges that must always be embedded plane.

Lemma 2. *Let μ be a node of $\mathcal{T}$ and let $e = \{u, v\}$ be a virtual edge in* $\text{skel}(\mu)$*. If both u and v have degree > 1 in* $\text{expg}(e)$*, then e must be embedded plane.*

Proof. Suppose e is embedded such that it crosses another edge e', which can be virtual or not. In either case, e' may be crossed at most once. As $\text{skel}(\mu)$ is biconnected and $\text{expg}(e)$ contains at least one additional vertex, in $\text{expg}(e)$, either all edges incident to u or all edges incident to v must be crossed in order

to have all vertices lie on the outer face. If both u and v have degree > 1 in $\text{expg}(e)$, e' has at least two crossings. □

Note that unlike planar embeddings, neither the skeleton of an S-node nor that of an R-node has a unique *o1p* embedding. However, Lemma 2 limits the number of possible *o1p* embeddings for skeletons considerably:

Lemma 3. *Let μ be a node in $\mathcal{T}$. Then for every virtual edge $e = \{u, v\}$ in* $\text{skel}(\mu)$ *holds:*
If $\text{refn}(e)$ *is a P- or an R-node, then e must be embedded plane in* $\text{skel}(\mu)$.
If $\text{refn}(e)$ *is an S-node whose skeleton is the cycle graph $(u, c_1, c_2, \ldots, c_k, v, u)$, then e must be embedded such that the segment incident to u (v) lies on the outer face if the edge $\{u, c_1\}$ ($\{c_k, v\}$) is virtual.*

Proof. If $\text{refn}(e)$ is a P- or an R-node, both u and v have degree > 1 in $\text{expg}(e)$, hence by Lemma 2, e must be embedded plane. Suppose $\text{refn}(e)$ is an S-node whose skeleton is the cycle graph $(u, c_1, c_2, \ldots, c_k, v, u)$. As the embedding must be such that all vertices lie on the outer face, only the edges $\{u, c_1\}$ or $\{c_k, v\}$ may be crossed. Recall that by the structure of an SPQR-tree, if $\{u, c_1\}$ ($\{c_k, v\}$) is virtual, then $\text{refn}(\{u, c_1\})$ ($\text{refn}(\{c_k, v\})$) is either a P- or an R-node, so $\{u, c_1\}$ ($\{c_k, v\}$) must be embedded plane. □

Lemma 1, Proposition 3, and Lemma 3 allow us to draw the following conclusion:

Corollary 1. *Every virtual edge in an S-node must be embedded plane.*
The skeleton of every R-node contains at most four virtual edges, which must be embedded plane, and no vertex may be incident to more than two virtual edges.
The skeleton of a P-node may have at most 4 virtual edges.

The conditions for R-nodes are easily checked by Algorithm 1 in time $\mathcal{O}(1)$ per R-node. Additionally, if an R-node is adjacent to another R-node or an S-node, then one of the edges of the K_4 is not present. For an example, see the R-nodes ρ_1 and ρ_2 in Fig. 2(b). By Proposition 1, however, the edge may be inserted and is always plane. Observe that this introduces a new P-node π_5 in Fig. 2(c). As an R-node may have at most four neighbors and as the SPQR-tree can be updated in $\mathcal{O}(1)$ time, this modification takes constant time, too.

The following lemma allows us to insert a non-virtual edge in every P-node, if none is present. In Fig. 2(b), this would apply, e. g., to π_1.

Lemma 4. *Let u, v be the vertices in the skeleton of a P-node without non-virtual edges. Then, the insertion of $\{u, v\}$ does not violate outer 1-planarity and $\{u, v\}$ is plane for every o1p embedding of G.*

Proof. Let π be a P-node with separation pair u, v, that is connected by virtual edges only. According to the definition of SPQR-trees, every skeleton of a P-node has at least three edges. Hence, π is adjacent to at least three other nodes. Subsequently, at least two virtual edges must be refined by S-nodes and are embedded with a crossing. This results in a crossing of two non-virtual edges in G that are, by Lemma 3, incident to u and v, respectively. By Proposition 1, the edge $\{u, v\}$ can always be inserted and is plane. □

Again, Algorithm 1 can check these two conditions and augment the graph for a P-node in time $\mathcal{O}(1)$, which results in a running time of $\mathcal{O}(n)$ for ll. 4 – 12.

Consider a P-node π with vertices u, v. If skel(π) has at most two virtual edges, they can be embedded without a crossing and such that both completely lie on the outer face. Suppose skel(π) has at least three virtual edges. In consequence of Proposition 3, two of them must cross each other. In Fig. 2(b), this holds for π_1 and π_2. We say that a P-node π *claims* a non-virtual edge e, and express this by defining the mapping $\mathcal{C}(e) = \pi$, if e is crossed in every embedding of skel(π) that conforms with Lemma 3. Observe that $\mathcal{C}$ is uniquely defined, since G is *o1p* and thus, no edge may be crossed more than once. We say that an embedding of the skeleton of a P-node is *admissible* if it conforms with Lemma 3 and does not imply the crossing of non-virtual edges claimed by other P-nodes. In Fig. 2(b), e. g., π_1 has two admissible embeddings, but both imply crossing the edge $\{f, m\}$, either by $\{d, i\}$ or by $\{h, i\}$. Hence, π_1 claims $\{f, m\}$. Computing $\mathcal{C}$ involves checking the embeddings of all P-nodes. As every P-node has at most four virtual edges, there are at most $\binom{4}{2} \cdot 2 = 12$ embeddings. Hence, the total time needed for this step is in $\mathcal{O}(n)$.

If every admissible embedding of skel(π) yields the same set of edges that are crossed, then π is called *fixable*. Let e, e' be two virtual edges that are embedded crossing each other. Observe that in this case, two S-nodes, namely refn(e) and refn(e'), are "crossing". By Proposition 1, the crossing can be augmented to a K_4. The insertion of these additional edges transforms the crossing S-nodes into an R-node that represents the K_4. In Fig. 2(b), this happens to π_1, σ_1, and σ_2. If the skeleton of an S-node previously had exactly three vertices, it is now completely contained in the K_4. Otherwise, its number of vertices is reduced by exactly 1, i. e., the vertex u or v, respectively. Note that completing the K_4 may affect the number of admissible embeddings and hence the fixability of other P-nodes if there was an admissible embedding of their skeletons that implied crossing one of e or e'. Algorithm 2 checks whether the virtual edges may cross each other and fixes the embedding of π. The next lemma enables us to also proceed when there is no fixable P-node.

Lemma 5. *Let π be a non-fixable P-node. If $\mathcal{T}$ has no fixable P-nodes, then every admissible embedding of* skel(π) *maintains at least one admissible embedding for every other P-node.*

Proof. Consider the fixing procedure of an embedding for a P-node π and S-nodes σ' and σ''. Let e' and e'' be the non-virtual edges that are crossed thereby. This affects the number of admissible embeddings for the skeletons of at most two other P-nodes π' and π'', that are adjacent to σ' and σ'', respectively. Observe that $\pi' \neq \pi''$, as $\mathcal{T}$ is a tree, and that every non-virtual edge is represented in the skeleton of exactly one node of $\mathcal{T}$.

Consider π'. W. l. o. g., let e' be the non-virtual edge in skel(σ') that is crossed after the fixing. Then, the number of admissible embeddings of skel(π') is reduced by exactly those that implied crossing e', too. However, π' did not claim e', so there is at least one other admissible embedding of skel(π'). The same argument holds for π'' and e''. □

Algorithm 2 Fix Embedding of P-node with two crossing S-nodes

```
1: procedure FixCrossingAtPNode(G, T, P-Node π, S-Node σ1, S-Node σ2)
2:     Let u, v be the separation pair of π,
3:     Let (u, c1, ..., ck, v, u) be the cycle in skel(σ1).
4:     Let (u, d1, ..., dl, v, u) be the cycle in skel(σ2).
5:     if {ck, v} virtual or {u, d1} virtual then
6:         if {u, c1} virtual or {dl, v} virtual then return ⊥
7:         else swap roles of σ1, σ2
8:     Pd ← ∅                                   ▷ possibly affected P-nodes
9:     if k > 1 then insert edge {u, ck} in G, update T
10:        if {ck−1, ck} virtual then add its associated P-node to Pd
11:    else if {u, ck} virtual then add its associated P-node to Pd
12:    if l > 1 then insert edge {v, d1} in G, update T
13:        if {d1, d2} virtual then add its associated P-node to Pd
14:    else if {v, d1} virtual then add its associated P-node to Pd
15:    insert edge {ck, d1}, update T
16:    if π has two (other) virtual edges then add π to Pd
17:    return Pd
```

Hence, by applying Lemma 5, we can step by step fix all embeddings of the skeletons of P-nodes with at least three virtual edges. Afterwards, every P-node has exactly two virtual edges and one non-virtual (cf. Fig. 2(c)). In Algorithm 1, this corresponds to ll. 15 – 26. FixCrossingAtPNode takes $\mathcal{O}(1)$ time per call and there are embeddings of at most $\mathcal{O}(n)$ P-nodes to fix. Hence, the time for this part is $\mathcal{O}(n)$. The algorithm concludes by selecting an admissible embedding for all P- and R-nodes. All remaining S-nodes are embedded as plane cycles. The embedding for G is obtained via the 2-clique-sums of all skeleton embeddings (cf. Fig. 2(d)). Consequently, Algorithm 1 has a running time of $\mathcal{O}(n)$.

It remains to show that all conditions presented so far are also sufficient for a graph to be *o1p*. Every skeleton is, taken by itself, embedded *o1p*. Consider the 2-clique-sum of two skeleton embeddings. This operation glues both graphs together at two virtual edges. After the augmentation of Algorithm 1, every virtual edge is embedded such that it lies on the outer face. Hence, in the resulting embedding, every vertex still lies on the outer face and every edge is crossed at most once. With this, the outer 1-planarity of the whole embedding follows by structural induction.

Lemma 6. *A graph G is o1p if and only if it is a subgraph of a graph H with SPQR-tree $\mathcal{T}$ such that R-nodes and S-nodes are adjacent to P-nodes only, every skeleton of an R-node is a K_4, and every skeleton of a P-node has exactly one non-virtual and two virtual edges.*

This concludes the proof of Theorem 1. Additionally, if a graph is *o1p*, Algorithm 1 also provides an *o1p* embedding. With some extra effort, we can augment G to maximality. Consider the supergraph H constructed from G by Algorithm 1

and its SPQR-tree. It may have S-nodes with four or more vertices. As all remaining S-nodes are embedded plane, we can insert a plane edge between two non-adjacent vertices, which splits the S-node into two smaller S-nodes with an intermediate P-node. This procedure can be repeated until all S-nodes are triangles. Next, consider a P-node that is adjacent to exactly two S-nodes, e. g., π_4 in Fig. 2(c). Then, we can insert a crossing edge ($\{g, i\}$ in the example) that augments the subgraph to a K_4. As a result, the nodes π_4, σ_6, and σ_7 are replaced by a new R-node. We denote by H^+ this supergraph of H. Its SPQR-tree consists of R-nodes, of which each corresponds to a K_4 and S-nodes, of which each corresponds to a triangle. R- and S-nodes are only connected via P-nodes, which in turn have exactly two virtual edges and one non-virtual. Consider an embedding of H^+. It has a tree-like structure that consists of K_4s and triangles (K_3s) that share an edge if and only if their corresponding R- and S-nodes are connected via a P-node. As no P-node is adjacent to two S-nodes, triangles can only share an edge with K_4s. Suppose H^+ was not maximal. If we were able to insert an inner, plane edge, this would correspond to inserting a P-node into the SPQR-tree of H^+. However, no two P-nodes may be adjacent. Inserting an inner, crossed edge is equal to augmenting two triangles to a K_4, which is impossible, too, as no P-node is adjacent to two S-nodes. Finally, consider adding an edge to the outer face. As every crossing has been augmented to a K_4, the boundary of the outer face consists of a plane Hamiltonian cycle. Hence, every additional edge would shield at least one vertex from the outer face. Consequently, we can easily extend Algorithm 1 such that it maximizes the input graph. Additionally, we receive another characterization:

Lemma 7. *A graph G is maximal o1p if and only if the conditions for H in Lemma 6 hold and in its SPQR-tree, no P-node is adjacent to more than one S-node and the skeleton of every S-node is a cycle of length three.*

The argument above also implies that every embedded maximal *o1p* graph is maximal for all *o1p* embeddings.

Corollary 2. *A graph G is maximal o1p if it has a maximal o1p embedding.*

Note that the embedding of a maximal *o1p* graph is fixed if and only if that of the skeleton of every R-node is. This, in turn, is the case iff it contains at least two incident virtual edges.

Corollary 3. *The embedding of a maximal o1p graph is unique up to inversion if and only if the skeleton of every R-node of its SPQR-tree contains a vertex that is incident to exactly two virtual edges.*

Another consequence of Lemma 7 is, that every maximal *o1p* graph is composed of triangles and K_4s. Changing the embedding of the K_4s, we obtain:

Corollary 4. *Every o1p graph is planar and has treewidth at most three.*

Observe that if the step that augments a P-node with two adjacent triangle S-nodes to a K_4 is omitted, we obtain a plane-maximal *o1p* graph, i. e., every additional edge either violates outer 1-planarity or introduces a new crossing. Equivalently, we can also adjust Algorithm 1 to test (plane) *o1p* maximality.

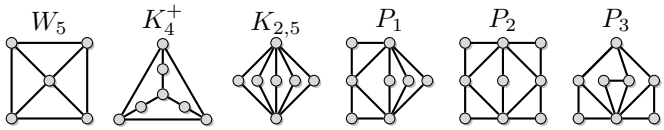

Fig. 3: Set M of minors of non-*o1p* graphs

Corollary 5. *There is a linear-time algorithm to test whether a graph is maximal (plane-maximal) o1p or to augment an o1p graph to a maximal (plane-maximal) o1p graph.*

From the recognition algorithm, we can immediately derive minors of non-*o1p* graphs: If the algorithm returns $\perp$, the graph at hand contains at least one of the *o1p* minors M as depicted in Fig. 3.

Theorem 2. *If a graph is not o1p, it contains at least one graph in M as a minor. Further, M is minimal and every graph in M is not o1p while removing or contracting an edge makes it o1p.*

Note that a graph might still be *o1p* even if it contains a graph in M as a minor, as outer 1-planar graphs are not closed under taking minors. The first minor W_5 is the wheel with five vertices, which is the smallest triconnected graph that is not *o1p* (Lemma 1). W_5 occurs in ll. 2 and 6 of Algorithm 1. If $\perp$ is returned in l. 2, then the graph contains a K_5 or $K_{3,3}$ as minor and W_5 is a minor of both. In l. 6, the first of the two checks implies W_5: If the R-node contains more than four vertices, $\perp$ is returned and the whole graph contains a planar triconnected component with at least four vertices, which always contains a W_5 as a minor. If the second condition in l. 6 is true, then the R-node at hand is a K_4 that contains a vertex v incident to three virtual edges. As the expansion graph of a virtual edge has a path with two edges as minor, we obtain K_4^+ in Fig. 3, where vertex v is in the center. If a P-node has at least five virtual edges (l. 11), then the $K_{2,5}$ is the minor. The remaining minors P_1, P_2, and P_3 can occur when fixing the embedding of a P-node with two crossing S-nodes. Consider l. 6 in Algorithm 2. If $\{u, d_1\}$ and $\{u, c_1\}$ are virtual, then u is incident to virtual edges in both S-nodes σ_1 and σ_2. If u is incident to at least one other virtual edge in π in whose expansion graph, u has at least degree two, then π has no admissible embedding and we obtain P_3 as minor. By a complete case differentiation, P_1 and P_2 can also be identified as minors.

4 Conclusion

We have designed a linear-time recognition algorithm for *o1p* that uses the SPQR-tree and returns a witness in terms of an *o1p* embedding or detects one of six minors, respectively.

Are there other classes of 1-planar graphs which can be recognized efficiently?

References

1. Alam, M.J., Brandenburg, F.J., Kobourov, S.G.: Straight-line drawings of 3-connected 1-planar graphs. In: Wismath, S., Wolff, A. (eds.) GD 2013. LNCS, vol. 00, pp. xx–xx. Springer, Heidelberg (2013)
2. Auer, C., Brandenburg, F.J., Gleißner, A., Reislhuber, J.: On 1-planar graphs with rotation systems. Tech. Rep. MIP 1207, University of Passau (2012)
3. Bannister, M.J., Cabello, S., Eppstein, D.: Parameterized complexity of 1-planarity. In: Dehne, F., Solis-Oba, R., Sack, J.R. (eds.) WADS 2013. pp. 97–108 (2013)
4. Brandenburg, F.J., Eppstein, D., Gleißner, A., Goodrich, M.T., Hanauer, K., Reislhuber, J.: On the density of maximal 1-planar graphs. In: Didimo, W., Patrignani, M. (eds.) GD 2012. LNCS, vol. 7704, pp. 327–338. Springer, Heidelberg (2013)
5. Cabello, S., Mohar, B.: Adding one edge to planar graphs makes crossing number and 1-planarity hard. Tech. Rep. arXiv:1203.5944 [cs.CG], Computing Research Repository (CoRR) (March 2012)
6. Dehkordi, H.R., Eades, P.: Every outer-1-plane graph has a right angle crossing drawing. Internat. J. Comput. Geom. Appl. 22(6), 543–558 (2012)
7. Di Battista, G., Tamassia, R.: On-line planarity testing. SIAM J. Comput. 25(5), 956–997 (1996)
8. 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. In: Didimo, W., Patrignani, M. (eds.) GD 2012. LNCS, vol. 7704, pp. 339–345. Springer, Heidelberg (2013)
9. Eades, P., Liotta, G.: Right angle crossing graphs and 1-planarity. In: van Kreveld, M.J., Speckmann, B. (eds.) GD 2011. LNCS, vol. 7034, pp. 148–153. Springer, Heidelberg (2012)
10. Eggleton, R.B.: Rectilinear drawings of graphs. Utilitas Math. 29, 149–172 (1986)
11. 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)
12. Hliněný, P.: Crossing number is hard for cubic graphs. J. Combin. Theory, Ser. B 96(4), 455–471 (2006)
13. Hong, S.H., Eades, P., Liotta, G., Poon, S.H.: Fáry's theorem for 1-planar graphs. In: Gudmundsson, J., Mestre, J., Viglas, T. (eds.) COCOON 2012. LNCS, vol. 7434, pp. 335–346. Springer, Heidelberg (2012)
14. Hong, S.H., Eades, P., Naoki, K., Liotta, G., Schweitzer, P., Suzuki, Y.: A linear-time algorithm for testing outer-1-planarity. In: Wismath, S., Wolff, A. (eds.) GD 2013. LNCS, vol. 8242, pp. 71–82. Springer, Heidelberg (2014)
15. Korzhik, V.P., Mohar, B.: Minimal obstructions for 1-immersion and hardness of 1-planarity testing. J. Graph Theor. 72, 30–71 (2013)
16. Mitchell, S.L.: Linear algorithms to recognize outerplanar and maximal outerplanar graphs. Inform. Process. Lett. 9(5), 229–232 (1979)
17. Ringel, G.: Ein Sechsfarbenproblem auf der Kugel. Abh. aus dem Math. Seminar der Univ. Hamburg 29, 107–117 (1965)
18. Thomassen, C.: Rectilinear drawings of graphs. J. Graph Theor. 12(3), 335–341 (1988)