

# Embedding-Preserving Rectangle Visibility Representations of Nonplanar Graphs

Therese Biedl<sup>1</sup> · Giuseppe Liotta<sup>2</sup> ·  
Fabrizio Montecchiani<sup>2</sup> 

Received: 2 September 2016 / Revised: 2 September 2017 / Accepted: 7 September 2017 /  
Published online: 4 October 2017  
© Springer Science+Business Media, LLC 2017

**Abstract** A (weak) rectangle visibility representation, or simply an RVR, of a graph consists of an assignment of axis-aligned rectangles to vertices such that for every edge there exists a horizontal or vertical line of sight between the rectangles assigned to its endpoints. Given a graph with a fixed embedding in the plane, we show that the problem of testing whether this graph has an embedding-preserving RVR can be solved in polynomial time for general embedded graphs and in linear time for 1-plane graphs, i.e., for embedded graphs having at most one crossing per edge. The linear time algorithm uses three forbidden configurations, which extend the set known for straight-line drawings of 1-plane graphs. The algorithm first checks for the presence of these forbidden configurations in the input graph, and then either an embedding-preserving RVR is computed (also in linear time) or a forbidden configuration is reported as a negative witness. Finally, we discuss extensions of our study to the case when the embedding is not fixed but the RVR can have at most one crossing per edge.

---

Editor in Charge: János Pach

---

An extended abstract of this paper appeared in the Proceedings of the 32nd Symposium on Computational Geometry, 2016, pp. 19:1–19:16. Research of Therese Biedl supported by NSERC. Research undertaken while Fabrizio Montecchiani was visiting the University of Waterloo supported by NSERC.

---

Therese Biedl  
biedl@uwaterloo.ca

Giuseppe Liotta  
giuseppe.liotta@unipg.it

Fabrizio Montecchiani  
fabrizio.montecchiani@unipg.it

<sup>1</sup> David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada

<sup>2</sup> Dipartimento di Ingegneria, Università degli Studi di Perugia, Perugia, Italy

**Keywords** Visibility representations · 1-Planarity · Fixed embedding · Forbidden configuration

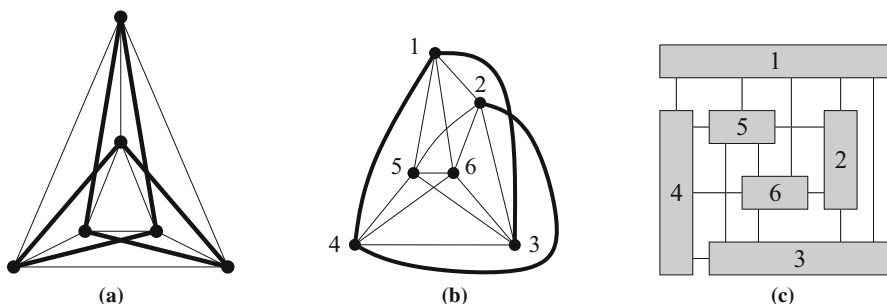
**Mathematics Subject Classification** 68U05 · 68R10 · 94C15

## 1 Introduction

A (weak) *rectangle visibility representation* (or *RVR*) is an appealing method of displaying graphs [15]. It consists of assigning axis-aligned rectangles to vertices and horizontal or vertical line segments (of width  $\varepsilon > 0$ ) to edges in such a way that line segments of edges begin and end at the rectangles representing their endpoints and intersect no rectangles in-between. Equivalently, each edge corresponds to a horizontal or vertical line-of-sight between its endpoints; hence the name “visibility”. Edge segments may cross each other, but any such crossing occurs at a right angle. Right-angle crossings and the absence of bends lead to a high readability of the visualization. See Fig. 1(c) for an illustration.

This kind of representation was introduced to the Computational Geometry community in 1985, when Wismath [47] showed that every planar graph has a (weak) *bar-visibility representation*, in which vertices are represented as horizontal bars and edges correspond to vertical visibilities. The same result was discovered independently multiple times [18, 35, 37, 42, 43]. Bar-visibility representations can exist only for planar graphs, so for representing nonplanar graphs one may use edge segments in both directions, as in rectangle visibility representations. Two necessary conditions for a graph  $G$  to have an RVR are the following, as shown in [25]. First,  $G$  must have thickness 2, i.e., it must be the union of two planar graphs. Second,  $G$  must have at most  $6n - 20$  edges (where  $n$  denotes the number of vertices of  $G$ ); this bound is tight for each  $n \geq 8$ . However, these two conditions are not sufficient. In fact, Shermer showed in 1996 that it is NP-hard to test whether a given graph has an RVR [38].

In this paper, we study the problem of testing whether a graph has a rectangle visibility representation if some aspects of it are fixed a priori. In previous work by Streinu and Whitesides [39], it was shown that testing whether a graph has an RVR can



**Fig. 1** (a, b) The two 1-planar embeddings of  $K_6$ . None of them can have an embedding-preserving RVR due to the presence of a forbidden configuration (bold edges). (c) An embedding-preserving RVR for the 1-planar embedding in (b) after removing edge (2, 4)

be solved in polynomial time if the following information is given as part of the input: For each edge it is known whether it is drawn horizontal or vertical, for each vertex  $v$  the fixed order of edges around  $v$  is known (this is also called a *rotation system*), and it is known which vertices form the outer face of the representation. Streinu and Whitesides give necessary and sufficient conditions for when such restrictions can be realized, and show that these can be tested in  $O(n^2)$  time, where  $n$  is the number of vertices of the graph.

We consider a slightly different scenario. Like Streinu and Whitesides, we assume that the rotation system and the outer face are fixed. However, we do not require to know the directions of the edges; instead we must know a priori which edges cross each other, and the order in which these crossings occur along each edge (if there is more than one crossing on an edge). In other words, the graph has a *fixed embedding* and we want an *embedding-preserving RVR* (see Sect. 2 for definitions). It turns out that a reduction to the topology-shape-metrics approach [41], combined with a recent breakthrough for the max  $st$ -flow problem [34], then allows to test in  $\tilde{O}((n+c)^{11/7})$  time<sup>1</sup> whether a graph with  $n$  vertices and  $c$  crossings has an embedding-preserving RVR, and to compute an RVR if it exists; we give the details in Sect. 2.

We then investigate *1-planar graphs*, i.e., those graphs that admit an embedding where each edge is crossed at most once (see, e.g., [30]). Previous attempts to represent 1-planar graphs had relaxed the restriction that edges must not cross non-incident vertices, and defined a (weak) *bar  $k$ -visibility representation* to be a bar-visibility representation where each line of sight can intersect at most  $k$  bars [14]. Brandenburg [7] and independently Evans et al. [20] proved that 1-planar graphs have a bar 1-visibility representation.

In contrast to these results, we consider in Sect. 3 rectangle visibility representations of 1-planar graphs. Furthermore, we consider *1-plane graphs*, i.e., embedded 1-planar graphs, and want to find an embedding-preserving RVR (also called *1-plane RVR*). It is easy to see that not all 1-plane graphs have a 1-plane RVR (see also Sect. 4). The main contribution of our paper is to show that for an  $n$ -vertex 1-plane graph it is possible to test in  $O(n)$  time whether it has a 1-plane RVR. The approach is entirely different from the topology-shape-metrics above and reveals much insight into the structure required for a 1-plane RVR to exist. Specifically, we consider three configurations, called B-configuration, W-configuration, and the newly defined T-configuration (illustrated in Fig. 4), and show that a 1-plane graph has an embedding-preserving RVR if and only if none of these three configurations occurs. This mirrors in a pleasing way the result by Thomassen [44], which shows that a 1-plane graph has an embedding-preserving straight-line drawing if and only if it contains no B-configuration or W-configuration. For example, the complete graph with six vertices admits the two 1-planar embeddings in Fig. 1 (a), (b). Both embeddings contain a forbidden configuration, highlighted with bold edges; Fig. 1 (c) shows a 1-plane RVR of the graph obtained from Fig. 1 (b) by removing edge (2, 4).

Our result provides an answer (in a special case) to Shermer's question of characterizing what graphs have RVRs [38]. Also, we prove that testing whether a 1-plane

<sup>1</sup> We recall that the notation  $\tilde{O}(f(n))$  means  $O(f(n)\text{poly}(\log f(n)))$ .

graph contains any of the three forbidden configurations can be done in  $O(n)$  time, and in the absence of them we can compute in  $O(n)$  time a 1-plane RVR which fits into a grid of  $O(n^2)$  size. We remark that embedding-preserving straight-line drawings of 1-plane graphs may require exponential area [24]. Furthermore, there are 1-planar graphs that do not admit any straight-line 1-planar drawing, due to the presence of a W-configuration in any possible 1-planar embedding [40]. However, every 3-connected 1-planar graph admits a 1-planar drawing in polynomial area where all edges are straight-line segments except at most one that has one bend [1].

The proof in Sect. 3 combines different techniques, namely: a novel technique to triangulate a 1-plane graph without changing its embedding; the 4-block tree to decompose the graph into its 4-connected components; transversal structures and rectangular dual representations to draw each component; and morphing tools such as zig-zag-slides to recursively combine computed subgraphs. From the algorithmic point-of-view, ingredients of the topology-shape-metrics approach are exploited to obtain linear time complexity and quadratic area for the resulting drawings.

Differently from planar graphs, 1-planar graphs may have an exponential number of different embeddings even if 3-connected or 4-connected (see, e.g., Fig. 14). Hence our algorithm does not automatically cover the case when the embedding is not fixed. We study 1-planar graphs without fixed embedding in Sect. 4. We show that there exist infinitely many 3-connected 1-planar graphs that contain linearly many edge-disjoint T-configurations in any possible 1-planar embedding. Hence, these graphs do not admit any 1-planar embedding with an embedding-preserving RVR, unless we remove at least one edge for each T-configuration. On the positive side, we show that 4-connected 1-planar graphs, which include the optimal 1-planar graphs, always admit an embedding-preserving RVR, after removing at most one edge. For example, Fig. 1(c) depicts a 1-planar embedding of  $K_6$  minus one edge.

Finally, in Sect. 5 we conclude with a discussion and open problems.

## 2 Definitions and Basic Results

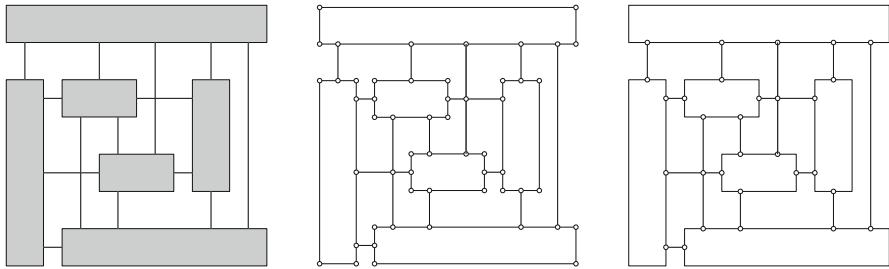
We consider *simple graphs*, which are graphs with neither loops nor multiple edges. A *drawing*  $\Gamma$  of a graph  $G$  maps each vertex to a distinct point of the plane and each edge to a simple Jordan arc between its two endpoints. We consider *simple drawings*, in which the arcs representing two edges have at most one point in common, which is either a common endpoint or a common interior point where the two arcs properly cross. A drawing is *planar* if no two arcs cross. A drawing divides the plane into topologically connected regions, called *faces*. The infinite region is called the *outer face*. For a planar drawing the boundary of a face consists of vertices and edges, while for a nonplanar drawing the boundary of a face may contain vertices, crossings, and edges (or parts of edges). An *inner face* (resp. edge, vertex) is a face (resp. edge, vertex) that is not part of the outer face. A *rotation system* of a graph  $G$  consists of assigning at each vertex a cyclic order of the edges incident to that vertex. A drawing is said to respect a rotation system if scanning around the vertex in clockwise order encounters the edges in the prescribed order. An *embedding* of a graph  $G$  is an equivalence class of drawings of  $G$  that define the same set of faces; in particular this determines

which pairs of edges cross each other. A graph with a *fixed embedding* comes with a rotation system, a cyclic order of edges around each crossing, and with one face indicated to be the outer face. A fixed embedding of a graph is sometimes called an *extended embedding* of the graph [32]. Given a graph  $G$  with a fixed embedding, the *planarization*  $G_p$  of  $G$  is the graph obtained by replacing each crossing of  $G$  with a *dummy-vertex*. Thus if edge  $(a, b)$  crosses  $(c, d)$ , then add a dummy-vertex  $x$  incident to all of  $a, b, c, d$  and remove the edges  $(a, b)$  and  $(c, d)$ . Vertex  $x$  uses as its cyclic order of edges the same order in which edges appeared at the crossing (which is given as part of the fixed embedding). The vertices of  $G$  in  $G_p$  are called *original vertices* to avoid confusion. The planarization  $G_p$  is a *plane graph*, i.e., a planar graph with a fixed embedding without crossings.

A *rectangle visibility representation* (or *RVR* for short) of a graph  $G$  is an assignment of disjoint axis-aligned rectangles to the vertices in such a way that for every edge there exists a horizontal or vertical line of sight between its endpoints (not all such lines of sight give rise to an edge). We follow a commonly adopted model where the lines of sight are *thick*, i.e., they have non-zero area (see e.g. [29, 38, 39, 42, 47]). Thus we may assume that each thick line of sight includes a (horizontal or vertical) line segment, attaching at each end at a point that is not a corner of the rectangle of that vertex. Note that our visibility model is *weak*, since two rectangles can be visible even if the two corresponding vertices are not adjacent. We remind the reader that there are other models of rectangle visibility representations (not studied further here): the  *$\varepsilon$ -visibility model* (refer to [38]) requires two rectangles in an RVR to be visible through a thick line of sight if and only the two corresponding vertices are adjacent in the depicted graph, and the *strong model* (refer to [42] in the context of bar-visibility representations) requires two rectangles in an RVR to be visible, possibly along a line of sight of zero width, if and only the two corresponding vertices are adjacent in the depicted graph.

In what follows, when this leads to no confusion, we shall use the same term *edge* to indicate an edge of a graph, the line of sight between two rectangles, and the line segment representing the line of sight, and we use the same term *vertex* for both the vertex of a graph and the rectangle that represents it. If we have an RVR, we can naturally extract a drawing from it as follows. Place a point for each vertex  $v$  inside the rectangle representing  $v$  and connect it to all the attachment points of incident edges of  $v$  inside the rectangle; this adds no crossing. We say that an RVR  $\Gamma$  of a graph  $G$  is *embedding-preserving*, or *respects* the embedding of  $G$ , if applying this operation to  $\Gamma$  results in the same embedding. An embedding-preserving RVR of a 1-plane graph is a *1-plane RVR*.

*Topology-Shape-Metrics.* The *topology-shape-metrics (TSM)* is a well-known method introduced by Tamassia [41] to compute *orthogonal drawings* of graphs, i.e., drawings where all vertices are points and edges are polylines with horizontal and vertical segments. The main idea is to express such a drawing as *orthogonal representations* (defined below). We now briefly review how any RVR gives rise to an orthogonal representation; this is relatively simple but has to our knowledge not been used before. We will use such orthogonal representations twice in our paper: once to determine whether



**Fig. 2** An RVR  $\Gamma$  (left), the PSLG (center) obtained from  $\Gamma$ , and the orthogonal drawing (right) obtained from  $\Gamma$ . The only difference between the latter two is whether the corners of rectangles become nodes or bends

a given embedded graph has an RVR, and once to store an RVR of a 4-connected 1-plane graph so that it can be manipulated efficiently (see Sect. 3.3).

A *planar straight-line graph* (PSLG for short) is a representation of a plane graph such that each vertex is a point of the plane and each edge is a straight-line segment connecting its two endpoints and intersecting no other edge or vertex. To avoid ambiguities, we use *nodes* and *arcs* for the vertices and edges of a PSLG. Consider an RVR  $\Gamma$  and convert it into a PSLG as follows (see also Fig. 2): replace every corner of a rectangle, every crossing, and every attachment point of an edge at a rectangle by a node of the PSLG, and connect two nodes with an arc if and only if they are connected by a segment. Then, each node of the PSLG has around it a sequence of angles that are all multiple of  $90^\circ$  and sum to  $360^\circ$ . Likewise, each face  $f$  of the PSLG has along it a sequence of angles that are all multiple of  $90^\circ$  and that sum to  $(a(f) - 2)180^\circ$  for inner faces and  $(a(f) + 2)180^\circ$  for the outer face, where  $a(f)$  is the number of angles inside  $f$  at the nodes of  $f$ .

The idea of an *orthogonal representation* of a plane graph  $G$  with maximum vertex degree 4 is that such sequences of angles (around a vertex or along a face) are enough to describe an orthogonal drawing of  $G$ , in the sense that suitable edge-lengths to realize them can always be found. To define this precisely, we follow the notation of [15]. We need that an edge  $(u, v)$  of  $G$  can be viewed as two *darts*, i.e., as the two distinct ordered pairs  $(u, v)$  and  $(v, u)$ .

**Definition 2.1** (based on [15]) An *orthogonal representation* of a plane graph  $G$  with maximum vertex degree 4 is a triple  $(G, \alpha, \beta)$  such that the following conditions are satisfied:

- $\alpha$  and  $\beta$  are two functions that associate a number  $\alpha(u, v) \in \{1, 2, 3, 4\}$  and a number  $\beta(u, v) \in \{0, 1, 2, \dots\}$  to every dart  $(u, v)$  of  $G$ , respectively;
- at every vertex  $u$ ,  $\sum_{v:(u,v) \in E} \alpha(u, v) = 4$ ;
- at every inner face  $f$ ,  $\sum_{(u,v) \in D(f)} \alpha(u, v) + \beta(v, u) - \beta(u, v) = 2a(f) - 4$ , where  $D(f)$  denotes the darts  $(u, v)$  for which  $f$  is the left face when walking from  $u$  to  $v$ ;
- at the outer face  $f$ ,  $\sum_{(u,v) \in D(f)} \alpha(u, v) + \beta(v, u) - \beta(u, v) = 2a(f) + 4$ .

We will call the last two conditions of Definition 2.1 the *sum-condition for faces*.

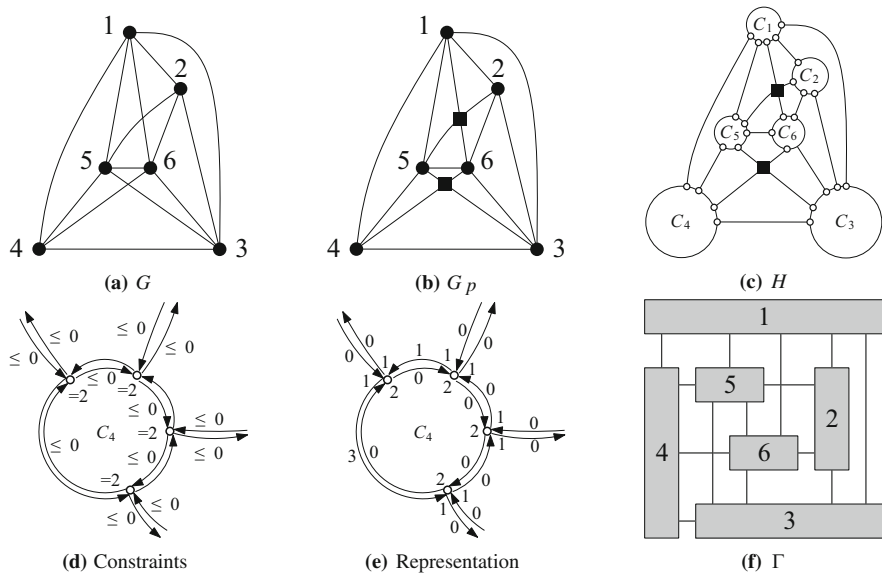
Any orthogonal drawing  $\Gamma$  of a planar graph  $G$  gives rise to an orthogonal representation  $(G, \alpha, \beta)$  as follows. For each dart  $(u, v)$ , let  $\alpha(u, v) \in \{1, 2, 3, 4\}$  be the angle (measured as multiples of  $90^\circ$ ) at vertex  $u$  between the incident segment of  $(u, v)$  and the incident segment of the counter-clockwise next edge at  $u$  in the orthogonal drawing  $\Gamma$ . Put differently, let  $f$  be the face at  $u$  between  $(u, v)$  and the counter-clockwise next edge at  $u$ ;  $\alpha(u, v)$  describes the *face-angle at  $u$  in  $f$* . Define further  $\beta(u, v)$  as follows. Let  $f$  be the face to the left when traversing dart  $(u, v)$  from  $u$  to  $v$ . Define  $\beta(u, v)$  to be the number of bends on  $(u, v)$  that have the  $90^\circ$  angle incident to face  $f$  in  $\Gamma$ ;  $\beta(u, v)$  is the *bend-count of  $(u, v)$  towards face  $f$* .

The following observation summarizes the above conversion of RVRs to PSLGs, which are orthogonal drawings without bends and hence yield orthogonal representations.

**Observation 2.2** *Assume that  $G$  has an RVR  $\Gamma$ , and construct a plane graph  $H$  by replacing every crossing, every edge attachment point and every corner of a vertex-rectangle of  $\Gamma$  with a node in  $H$ . Then there exists an orthogonal representation  $(H, \alpha, \beta)$  such that  $\beta(u, v) = 0$  for all darts  $(u, v)$ .*

Two orthogonal drawings  $\Gamma_1$  and  $\Gamma_2$  of  $G$  are *shape-equivalent* if they give rise to the same orthogonal representation. An orthogonal representation of  $G$  hence describes a class of shape-equivalent orthogonal drawings of  $G$ . Tamassia showed ([41], see also [15, Thm. 5.4]) that for any orthogonal representation  $(G, \alpha, \beta)$ , one can compute an orthogonal drawing  $\Gamma$  of  $G$  that belongs to the equivalence class of  $(G, \alpha, \beta)$  in  $O(n + b)$  time on a grid of size  $O((n + b)^2)$ , where  $n$  is the number of vertices of  $G$  and  $b = \sum_{(u,v) \in E} (\beta(u, v) + \beta(v, u))$  is the total number of bends in  $\Gamma$ .

An orthogonal representation of  $G$  with the minimum number of bends can be computed by means of a flow network  $\mathcal{N}$ : each unit of flow corresponds to a  $90^\circ$  angle, each vertex supplies 4 units of flow, and each face consumes an amount of flow proportional to its degree. Bends along edges correspond to unit of flows transferred across adjacent faces, and each bend has a unit cost in the network. More precisely, the flow network  $\mathcal{N}$  is constructed as follows. It has a *vertex-node* for each vertex of  $G$  and a *face-node* for each face of  $G$ . Each vertex-node  $v$  supplies  $\sigma(v) = 4$  units of flow, and each face-node  $f$  consumes  $\tau(f)$  units of flow, where  $\tau(f) = 2a(f) - 4$  if  $f$  is an internal face, and  $\tau(f) = 2a(f) + 4$  if  $f$  is the outer face. By Euler's formula,  $\sum_v \sigma(v) = \sum_f \tau(f)$ , i.e., the total supply is equal to the total consumption. For each dart  $(u, v)$  of  $G$ , with faces  $f$  and  $g$  on its left and right, respectively,  $\mathcal{N}$  has two *arcs*: (i) an arc  $(u, f)$  with lower bound  $\lambda(v, f) = 1$ , capacity  $\mu(v, f) = 4$ , and cost  $\chi(v, f) = 0$ , and (ii) an arc  $(f, g)$  with lower bound  $\lambda(v, f) = 0$ , capacity  $\mu(v, f) = +\infty$ , and cost  $\chi(v, f) = 1$ . The conservation of flow at the vertices expresses that the sum of the angles around a vertex must equal  $360^\circ$ . The conservation of flow at the faces expresses the sum-condition for the faces. Every feasible flow  $\Phi$  in network  $\mathcal{N}$  corresponds to an orthogonal representation for graph  $G$ , whose number of bends is equal to the cost of flow  $\Phi$ . More precisely, let  $\Phi$  be a flow of  $\mathcal{N}$  with cost  $b$ . Then, for each dart  $(u, v)$  whose associated arcs of  $\mathcal{N}$  are  $(u, f)$  and  $(f, g)$ , we set  $\alpha(u, v) = \Phi(u, f)$  and  $\beta(u, v) = \Phi(f, g)$ . On the other hand, by setting  $\Phi(u, f) = \alpha(u, v)$  and  $\Phi(f, g) = \beta(u, v)$ , an orthogonal representation  $H$  with at



**Fig. 3** Illustration of the procedure to compute an embedding-preserving RVR of a 1-planar embedding of  $K_6$  minus edge  $(2, 4)$  with the topology-shape metric approach. For readability we only show parts of the graph in (d, e). (a)  $G$ . (b)  $G_p$ . (c)  $H$ . (d) Constraints. (e) Representation. (f)  $\Gamma$

most  $b$  bends is transformed into a feasible flow  $\Phi$  of  $\mathcal{N}$  with cost  $b$ . The next theorem summarizes the above discussion.

**Theorem 2.3** (see e.g. [15]) *Let  $G$  be a plane graph with  $n$  vertices and maximum vertex degree 4. An orthogonal representation  $H$  of  $G$  with the minimum number of bends can be computed in  $O(T(n))$  time, where  $T(n)$  is the time for computing a min-cost flow of the flow network  $\mathcal{N}$  associated with  $G$ .*

As observed in [15], the flow network  $\mathcal{N}$  can be easily modified to handle constraints on face-angles and bend-counts. For example, to prescribe at most two bends on an edge with darts  $(u, v)$  and  $(v, u)$  we set the capacity of the arcs  $(f, g)$  and  $(g, f)$  on  $\mathcal{N}$  as  $\mu(f, g) = \mu(g, f) = 2$ . Given a constrained network  $\mathcal{N}$ , a feasible flow may not exist, but one can test the existence of an orthogonal representation that complies with the prescribed constraints by testing for a feasible flow.

*Testing Embedded Graphs for RVRs.* Assume that we are given a graph  $G$  with a fixed embedding. We will show how to test whether  $G$  has an RVR that respects its embedding using the TSM approach. Note that the embedding tells us nearly the entire topology of a (putative) RVR of  $G$ . The only thing unknown is the location of the corners of each rectangle of a vertex. We can now interpret these corners as bends (rather than as nodes) in the topology. Hence, define a *constrained plane graph*  $H$  as follows (see also Fig. 3): (i) Start with the planarization  $G_p$  of  $G$  (see, e.g., Fig. 3(a), (b)). (ii) For every original vertex  $v$  of  $G$ , define a cycle  $C_v$  with  $\deg(v)$  nodes in  $H$ , where  $\deg(v)$  is the degree of  $v$  in  $G$ . Attach the (parts of) edges incident to  $v$



to the nodes of cycle  $C_v$  in the specified order of the embedding, in such a way that the interior of the cycle forms a face. In other words, form a cycle that can represent the boundary of the rectangle of  $v$  (see, e.g., Fig. 3(c)). (iii) Impose constraints on orthogonal representations of  $H$  as follows (see, e.g., Fig. 3(d)). Every edge  $(u, v)$  of  $G$  has been replaced in  $H$  by a path that starts at  $C_u$ , ends at  $C_v$ , and possibly has dummy-vertices as inner vertices. For any dart  $(x, x')$  on such a path, impose an upper bound  $\beta(x, x') \leq 0$  on the bend-count. In any orthogonal representation we also have  $\beta(x, x') \geq 0$ , so effectively we do not allow any bends for the original edges. (iv) Every vertex  $v$  of  $G$  has been replaced by a cycle  $C_v$  in  $H$  that bounds a face  $f_v$ . If  $(x, x')$  is a dart of  $H$  that has  $f_v$  on its left side, then impose an upper bound  $\beta(x', x) \leq 0$ . In any orthogonal representation we must have  $\beta(x', x) \geq 0$ , so this implies  $\beta(x', x) = 0$ . Observe that  $\beta(x', x)$  counts the number of bends on edge  $(x', x)$  whose incident angle in  $f_v$  is  $270^\circ$ . Hence this constraint expresses that each bend on  $(x, x')$  must have its  $90^\circ$  angle towards face  $f_v$ . Furthermore, impose an upper bound  $\alpha(x, x') \geq 2$  and a lower bound  $\alpha(x, x') \leq 2$  onto the face-angle  $\alpha(x, x')$ . Since  $f_v$  is the face to the left of dart  $(x, x')$ , the angle at  $x$  inside  $f_v$  must be  $180^\circ$ .

**Theorem 2.4** *Let  $G$  be a graph with a fixed embedding,  $n$  vertices and  $c$  crossings. There exists an  $\tilde{O}((n+c)^{11/7})$ -time algorithm to test whether  $G$  admits an embedding-preserving RVR. In the positive case, the algorithm computes an RVR  $\Gamma$  that has  $O(n^2)$  area.*

*Proof* Build the constrained plane graph  $H$  as described above. We claim that  $H$  admits an orthogonal representation that satisfies the constraints of  $H$  if and only if  $G$  admits an RVR that respects its embedding.

Assume first that  $G$  admits an RVR  $\Gamma$  that respects its embedding. Consider the orthogonal drawing  $\Gamma'$  of the graph  $H'$  obtained from  $\Gamma$  by replacing every edge-vertex attachment point and every crossing with a node; see, e.g., Fig. 2. All edges of  $G$  correspond to straight-line segments in  $\Gamma'$  and all vertices of  $G$  correspond to rectangular faces such that no edge attaches at a corner in  $\Gamma'$ . Let  $(H', \alpha', \beta')$  be the orthogonal representation associated with  $\Gamma'$ . It is easy to verify that  $H'$  has the same set of vertices and edges as  $H$  and that both  $\alpha'$  and  $\beta'$  satisfy the constraints of  $H$ .

Now assume that  $H$  has an orthogonal representation that satisfies the constraints, see, e.g., Fig. 3(e). Compute an orthogonal drawing  $\Gamma_H$  that corresponds to this orthogonal representation. We claim that by dropping added nodes from  $\Gamma_H$ , we obtain an RVR of  $G$ , see, e.g., Fig. 3(f).

We first argue that for any vertex  $v$  of  $G$ , the cycle  $C_v$  must form a rectangle. Cycle  $C_v$  bounds an inner face  $f_v$  with  $a(f_v) = \deg(v)$  incident nodes. At each dart  $(u, v)$  of  $C_v$  that has  $f_v$  on the left side, we have  $\alpha(u, v) = 2$  and  $\beta(v, u) = 0$  due to our constraints. But  $\sum_{(u,v) \in D(f_v)} (\alpha(u, v) + \beta(v, u) - \beta(u, v)) = 2a(f_v) - 4$ , which implies  $\sum_{(u,v) \in D(f_v)} \beta(u, v) = 4$ . Since drawing  $\Gamma_H$  respects the orthogonal representation, cycle  $C_v$  has exactly four angles of  $90^\circ$ , none of which occur at a node. Hence  $C_v$  forms a rectangle  $R_v$ . We use  $R_v$  to represent  $v$  in the RVR, and observe that none of the corners of  $R_v$  are incident to an edge since  $\alpha(u, v) = 2$  for all  $(u, v) \in D(f_v)$ . Next consider any dummy-vertex  $x$ . It has degree 4, hence four incident faces. Let  $(x, y)$  be a dart incident to  $x$ . Since  $\alpha(x, y) \geq 1$  but  $\sum_{(x,y) \in E} \alpha(x, y) = 4$ , we must have

$\alpha(x, y) = 1$  for all such darts. In consequence, the four incident face-angles at any dummy-vertex  $x$  must be  $90^\circ$ . Now we argue that for any edge  $(u, v)$  of  $G$ , there exists a line of sight between the rectangles  $R_u$  and  $R_v$ . In  $H$ , the edge was (possibly) subdivided with dummy-vertices, so the edge is split into a path  $x_0, x_1, x_2, \dots, x_k, x_{k+1}$  for some  $k \geq 0$  with  $x_0 \in C_u$  and  $x_{k+1} \in C_v$ . We imposed the constraint  $\beta(x_i, x_{i+1}) = 0$  onto each dart of this path, which means that in  $\Gamma_H$  they are all drawn as line segments. Also, at  $x_i$  for  $1 \leq i \leq k$  the two incident arcs are not consecutive, and hence have angle  $180^\circ$  between them. Therefore the union of the arcs forms one line segment without bends connecting  $C_u$  to  $C_v$ . Thus  $R_u$  and  $R_v$  are connected by a line of sight, along this path, proving that we indeed obtained an RVR.

It remains to argue the time complexity and the area requirement. We initially check whether the number of edges of  $G$  is  $m \leq 6n - 20$ , as otherwise  $G$  has no RVR [25]. We then use the TSM flow approach to test if an orthogonal representation of  $H$  exists that satisfies all constraints. We construct a flow network  $\mathcal{N}$  starting from  $H$  as described earlier in this section. The following constraints are imposed: (i) Every arc  $(u, f_v)$  of  $\mathcal{N}$  from a vertex-node  $u$  to a face-node  $f_v$  has fixed flow 2 (i.e.,  $\lambda(u, f_v) = \mu(u, f_v) = 2$ ) if  $f_v$  corresponds to a cycle  $C_v$  (this implies a  $180^\circ$  angle inside the cycle); (ii) every arc  $(f, g)$  between two face-nodes such that neither  $f$  nor  $g$  corresponds to a cycle  $C_v$  of  $H$  is removed (to avoid bends on such an edge); (iii) every arc  $(g, f_v)$  between two face-nodes such that  $f_v$  corresponds to a cycle  $C_v$  of  $H$  is removed (to ensure that each cycle is represented as a rectangle); (iv) for each edge  $(f_v, g)$  between two face-nodes such that  $f_v$  corresponds to a cycle  $C_v$  of  $H$ , we set  $\mu(f_v, g) = 4$ , since each edge of a cycle can make at most four bends. In order to test for the existence of a feasible flow in  $\mathcal{N}$ , we reduce to the max  $st$ -flow problem on a network  $\mathcal{N}'$  constructed from  $\mathcal{N}$  as follows. We set  $\sigma(v) = 0$  for all vertex-nodes and  $\tau(f) = 0$  for all face-nodes. We add a super source  $s$  and a super sink  $t$ . We add an edge  $(s, v)$  for each vertex-node  $v$  of  $\mathcal{N}$  and we set  $\mu(s, v) = \sigma(v) = 4$ . We add an edge  $(f, t)$  for each face-node  $f$  of  $\mathcal{N}$  and we set  $\mu(f, t) = \tau(f) \in O(n + c)$ . Now we solve a max  $st$ -flow problem in  $\mathcal{N}'$ . Then the original network  $\mathcal{N}$  has a feasible flow  $\Phi$  if and only if the max  $st$ -flow  $\Phi'$  in  $\mathcal{N}'$  saturates all the arcs incident to  $s$ . For a flow network with  $M$  edges and bounded capacities, the max  $st$ -flow problem can be solved in  $\tilde{O}(M^{10/7}U^{1/7})$  time, where  $U$  is the largest (integer) capacity in the network [34]. Since  $\mathcal{N}$  has  $O(n + c)$  edges and  $U \in O(n + c)$ , it follows that we can test for the existence of a feasible flow in  $\mathcal{N}$ , and hence of an orthogonal representation satisfying all the desired constraints, in  $\tilde{O}((n + c)^{11/7})$  time. If a feasible flow exists, we replace all arcs of  $\mathcal{N}$  with capacity  $k > 1$  with  $k$  arcs having unit capacity. The unit-capacity min-cost flow problem can then be solved on the resulting flow network in  $\tilde{O}((n + c)^{10/7} \log W)$  time, where  $W$  is the maximum cost of an arc [12]. Thus, we can compute an orthogonal representation satisfying all the desired constraints in  $\tilde{O}((n + c)^{11/7})$  time, if it exists. Finally, in  $O(n + c)$  time, an orthogonal representation can be turned into an orthogonal drawing  $\Gamma_H$  of  $H$  (and hence into the RVR  $\Gamma$ ) that has  $O((n + c)^2)$  area [41]. We can improve the area-bound substantially.  $H$  has  $m_H = \sum_{v \in V} \deg(v) + m + 2c = 3m + 2c$  edges and  $\Gamma_H$  has  $B = 4n$  bends (four per vertex of  $G$ ). By the results of [4],  $\Gamma_H$  has half-perimeter at most  $B + 2n_H - m_H = 4n + m \in O(n)$ , so the area of  $\Gamma_H$  is  $O(n^2)$ .  $\square$

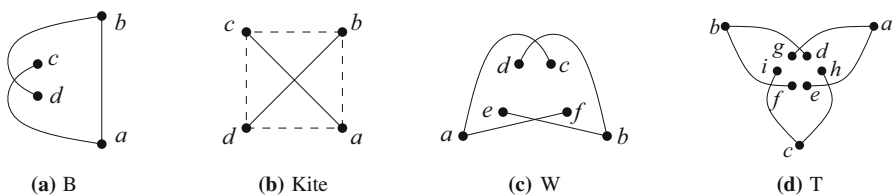
We remark that Theorem 2.4 contributes to answering a question posed by Streinu and Whitesides [39] about what topological information added to an input graph  $G$  makes the problem of testing whether  $G$  has an RVR polynomial.

### 3 Embedded 1-Planar Graphs

While the TSM approach used to prove Theorem 2.4 gives a polynomial way to test whether an embedded graph has an embedding-preserving RVR, its black-box approach naturally leads to more ambitious research questions about testing the representability of embedded graphs. In particular, in case of a negative answer, we obtain no good insights as to which parts of the graph prevented the existence of a representation. In this section, we therefore turn our attention to 1-planar graphs. We chose this graph class for two reasons. One is that they are known to have thickness 2 and at most  $4n - 8$  edges [6, 36], and so they are good candidates for always having an embedding-preserving RVR. (We will, however, see that this is not the case.) Secondly, 1-planar graphs have been studied widely in the graph theory and graph drawing communities (see e.g. [1, 13, 24, 30, 31, 36, 40, 44]), and visibility representations of these graphs are of interest [7, 16, 20, 33].

*Background.* A graph is 1-planar if it has a drawing with at most one crossing per edge. A 1-plane graph  $G$  is a 1-planar graph with a fixed embedding. Graph  $G$  is triangulated if all faces are triangles. In a 1-plane graph, no two crossings are adjacent and so a triangle consists of three vertices or two vertices and one crossing point.

Let  $e = (a, c)$  and  $e' = (b, d)$  be two edges of  $G$  that cross at a point  $p$ . We say that  $e, e'$  induce a *B-configuration* [43] if there exists an edge between their endpoints (say edge  $(a, b)$ ) such that (in the induced drawing of  $\{a, b, c, d\}$ ) the triangle with vertices  $a, b, p$ , denoted by  $\Delta(a, b, p)$ , contains vertices  $c$  and  $d$  inside; see also Fig. 4(a) for an illustration. If the crossing is not in a B-configuration, then there are two further possibilities for edge  $(a, b)$ . The triangle  $\Delta(a, b, p)$  may be an inner face, or it could have other vertices inside. If, for all pairs in  $\{a, c\} \times \{b, d\}$ , there either is no edge between the vertices or the triangle that it forms is a face, then we could add the edges that did not exist previously, routing them along the crossing, and then obtain a *kite*, i.e., a crossing with a 4-cycle among its endpoints such that removing the crossing turns the 4-cycle into a face; see also Fig. 4(b) for an illustration (the added edges are dotted). Let  $(a, c)$  and  $(b, d)$  be two edges of  $G$  that cross at a point  $p$ , and let  $(a, f)$



**Fig. 4** Possible crossing configurations in a 1-plane graph. (a) A B-configuration. (b) A (subgraph) of a kite. The dashed edges either do not exist or define faces. (c) A W-configuration. (d) A T-configuration

and  $(b, e)$  be two further edges of  $G$  that cross at a point  $q$ . The four edges induce a *W-configuration* [43] if vertices  $c, d, e, f$  lie inside the closed region delimited by the edge segments  $(a, p)$ ,  $(b, p)$ ,  $(a, q)$ , and  $(b, q)$ ; see also Fig. 4(c) for an illustration. For the results in this paper, we introduce a fourth configuration, called the *trillium configuration*, or *T-configuration* for short, which is illustrated in Fig. 4(d). Namely, let  $(a, c)$  and  $(b, d)$  be a pair of edges of  $G$  that cross at a point  $p$ . Let  $(a, e)$  and  $(c, h)$  be a second pair of edges that cross at a point  $q$ . Let  $(b, f)$  and  $(c, i)$  be a third pair of edges that cross at a point  $t$ . The six edges induce a T-configuration if they are all distinct and the vertices  $d, e, f, g, h, i$  lie inside the closed region delimited by the edge segments  $a - p - b - t - c - q - a$ . Vertices  $a, b, c$  are called the *outer vertices* of the T-configuration, while the remaining six vertices are the *inner vertices*.

In the following we will mostly work with the planarization  $G_p$  of  $G$  where crossings are replaced by dummy-vertices. We use *planarized* versions of a B-configuration, W-configuration and T-configuration for  $G_p$ , which are defined in exactly the same way except that “crossing” is replaced by “dummy-vertex” everywhere.

### 3.1 Main Result

In this section we give a characterization of those 1-plane graphs that admit a 1-plane RVR. Our characterization is summarized by the following theorem.

**Theorem 3.1** *A 1-plane graph admits a 1-plane RVR if and only if it contains no B-configuration, no W-configuration, and no T-configuration.*

Notably, Theorem 3.1 extends the characterization for straight-line drawability of 1-plane graphs given by Thomassen [43], adding the T-configuration to the set of forbidden subgraphs. The necessity of the condition is proved via an angle-counting-argument below. The sufficiency-proof is more elaborate and is given in Sect. 3.2. It comes with an efficient algorithm to construct a 1-plane RVR of a drawable graph. Area and time complexity issues are discussed in Sect. 3.3. Finally, in Sect. 3.4, we describe a testing procedure that follows from Theorem 3.1. The next theorem is obtained by combining our testing procedure and our drawing algorithm.

**Theorem 3.2** *Let  $G$  be a 1-plane graph with  $n$  vertices. There exists an  $O(n)$ -time algorithm to test whether  $G$  admits a 1-plane RVR. In the positive case the algorithm computes an RVR of  $G$  that has  $O(n^2)$  area.*

To prove the necessity of Theorem 3.1, we show a slightly stronger statement, which does not assume 1-planarity.

**Lemma 3.3** *Let  $G$  be a simple graph that admits an RVR  $\Gamma$ . If while walking along the outer face of  $\Gamma$  we encounter  $k$  vertices and  $c$  crossing points, then  $k \geq 3$  and  $c \leq 2k - 4$ .*

*Proof* Convert  $\Gamma$  into an orthogonal representation (Observation 2.2) and consider the sum of face-angles at the outer face. Since there are  $k$  vertices on the outer face, we have  $2k$  edge-attachment points, and each contributes  $90^\circ$  to the sum of angles. Each of the  $c$

crossings also contributes  $90^\circ$  to the sum. If  $b$  is the number of rectangle-corners on the outer face, then  $b \leq 4k$ , and each contributes  $270^\circ$  to the sum. In total the outer face has  $2k + c + b$  nodes and the sum of angles must be  $180^\circ(2k + c + b + 2)$ . But we also know that the angles sum to  $90^\circ(2k + c) + 270^\circ b = 180^\circ(2k + c + b + 2) + 90^\circ(b - 2k - c - 4)$ . So  $b - 2k - c - 4 = 0$ , hence  $4k \geq b = 2k + c + 4$ , and  $c \leq 2k - 4$ . This implies  $k \geq 2$ , and in fact  $k \geq 3$  is required, else  $c = 0$  and the outer face would consist of a double edge.  $\square$

Since a B-configuration (W-configuration, T-configuration) has as outer face a cycle with  $k = 2$  and  $c = 1$  ( $k = 2$  and  $c = 2$ ,  $k = 3$  and  $c = 3$ ), the following holds.

**Corollary 3.4** *Let  $G$  be a 1-plane graph that admits a 1-plane RVR. Then  $G$  contains no B-configuration, no W-configuration, and no T-configuration.*

### 3.2 Proof of Sufficient Condition

Let  $G$  be a 1-plane graph with no B-configuration, no W-configuration, and no T-configuration as a subgraph, see also Fig. 5(a) for an example. We give a drawing algorithm, 1P-RVDRAWER, which computes a 1-plane RVR  $\Gamma$  of  $G$ ; a high-level description of 1P-RVDRAWER is given in Algorithm 1.

---

**Algorithm 1:** Pseudocode for algorithm 1P-RVDRAWER.

---

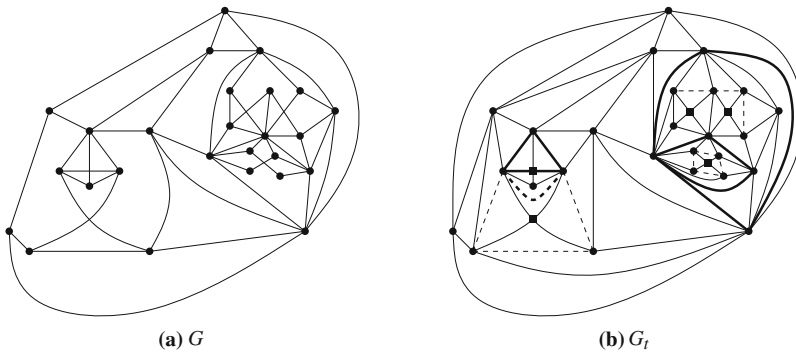
```

1 1P-RVDRAWER ( $G$ )
   input : A 1-plane graph  $G$  with no B-configuration, no W-configuration, and no
         T-configuration.
   output: A 1-plane RVR  $\Gamma$  of  $G$ .
2   $G_p \leftarrow \text{Planarize}(G)$ 
3   $G_p^+ \leftarrow \text{ADDKITEEDGES}(G_p)$  // See Algorithm 2
4   $G_t \leftarrow \text{Triangulate}(G_p^+)$ 
5  while  $G_t$  has multiple edges, say at vertex-pair  $\{v, w\}$  do
6      $G_t \leftarrow \text{FLIPMULTIEDGES}(G_t, v, w)$  // See Algorithm 4
7  end
8   $G^+ \leftarrow G_t$ 
9   $\mathcal{T} \leftarrow \text{Compute4BlockTree}(G^+)$ 
10  $\rho \leftarrow \text{Root}(\mathcal{T})$ 
11 foreach  $\mathcal{C}_v$  in  $\text{TopDownTraversal}(\mathcal{T})$  do
12      $e \leftarrow \text{CHOOSESURROUNDEGE}(\mathcal{C}_v)$  // See Algorithm 3
13      $\gamma_v \leftarrow \text{ComputeRVR}(\mathcal{C}_v, e)$ 
14 end
15 foreach  $\gamma_v$  in  $\text{BottomUpTraversal}(\mathcal{T})$  do
16      $\Gamma_v \leftarrow \text{MergeChildren}(\gamma_v, \Gamma_{\mu_1}, \dots, \Gamma_{\mu_h})$ 
17 end
18  $\Gamma_p \leftarrow \Gamma_\rho$  with all edges in  $G^+ - G_p$  removed
19  $\Gamma \leftarrow \text{Unplanarize}(\Gamma_p)$ 

```

---

*Triangulating and Planarizing.* The first few steps of the algorithm aim to triangulate  $G$  without introducing multiple edges. This is a well-known operation for 3-connected 1-plane graphs, see [1, 24]. However, these algorithms all modify the given 1-planar



**Fig. 5** (a) A 1-plane graph  $G$ . (b) The triangulated plane graph  $G_t$  obtained from  $G$ . Dummy-vertices are black squares. Dashed edges were added by ADDKITEEDGES

embedding, even if there is no B-configuration or W-configuration. In fact, this may be required since every crossing must form a kite in a triangulation of  $G$ , and so if an edge between endpoints of a crossing already existed, but did not form a face, then it must be re-routed. We do not want to change the embedding, and hence cannot triangulate the 1-plane graph per se. Instead, we planarize first, and then triangulate it in a careful way. The planarization is done in the standard way (see Sect. 2), by replacing crossings by dummy-vertices that inherit the embedding. The rest of the algorithm will only work on the planarization  $G_p$  (with the exception of the last step), and so from now when using the terms B-configuration, W-configuration, T-configuration, we mean the planarized version that uses “dummy-vertex” in place of “crossing” as explained earlier.

Now we triangulate  $G_p$  and do this in such a way that all dummy-vertices have degree 4. To ensure this, we first add edges between the four neighbors of each dummy-vertex  $x$  so that the four faces of  $x$  are all triangles. Put differently, we ensure that the corresponding crossing in  $G$  is surrounded by kite-edges. See also Algorithm 2. In the resulting graph  $G_p^+$ , all faces of degree 4 or more then only contain original vertices. We triangulate such faces by adding edges within them in the standard manner; it is well-known that this can be done in linear time while ensuring that no multiple edges are added (see, e.g., [27]). For example, Fig. 5(a) and shows a 1-plane graph  $G$  and the plane triangulated graph  $G_t$  obtained with 1P-RVDRAWER, respectively.

Note that Algorithm ADDKITEEDGES may add multiple edges. In particular, if  $(x, v)$  and  $(x, w)$  are consecutive edges at a dummy-vertex  $x$ , then edge  $(v, w)$  gets added to create a triangular face  $\Delta(x, v, w)$ , even if edge  $(v, w)$  exists already at some other location of the graph. However, this is the only situation in which the resulting graph  $G_p^+$  may have multiple edges. When triangulating  $G_p^+$  to obtain  $G_t$ , we add no more multiple edges, and hence have the following property:

**Observation 3.5** *Graph  $G_t$  may have multiple edges, but if  $(v, w)$  is such a multiple edge, then  $v$  and  $w$  are original vertices, and for all except at most one copy of  $(v, w)$  there exists a dummy-vertex  $x$  that forms a triangular face  $\Delta(x, v, w)$  in  $G_t$ .*

We observe some further properties of  $G_t$  that will be needed later.

---

**Algorithm 2:** Pseudo-code for adding edges around dummy-vertices.

---

```

1 ADDKITEEDGES ( $G_p$ )
2   foreach dummy-vertex  $x$  do
3     foreach pair of consecutive edges  $(x, v), (x, w)$  at  $x$  do
4        $f \leftarrow$  face incident to  $(x, v)$  and  $(x, w)$ 
5       if  $f$  is the outer face, or  $f$  is not a triangle then
6         Insert edge  $(v, w)$  // Even if the edge existed already in  $G$ 
7         Fix the embedding such that  $\Delta(x, v, w)$  is an inner face
8       end
9     end
10  end

```

---

**Observation 3.6** *In graph  $G_t$ , all dummy-vertices have degree 4 and are inner. Furthermore, if  $G$  has no  $W$ -configuration or  $T$ -configuration then  $G_t$  has no planarized  $W$ -configuration or  $T$ -configuration.*

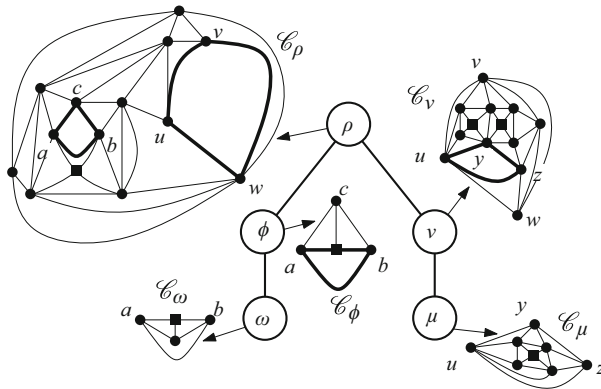
*Proof* The first claim is immediately since ADDKITEEDGES surrounds each dummy-vertex  $x$  by a kite. Hence, all faces incident to  $x$  are triangles that are inner faces, and so  $x$  is an inner vertex and will receive no further edges when triangulating.

For the second claim, observe that both ADDKITEEDGES and the triangulation-step only add edges for which both endpoints are original vertices. Such edges can never create a planarized  $W$ -configuration or  $T$ -configuration since all edges in these configurations are incident to dummy-vertices. □

Unfortunately,  $G_t$  may have a planarized  $B$ -configuration even if  $G$  had none, and  $G_t$  need not be simple. Dealing with this requires a fairly elaborate detour and another procedure FLIPMULTIEDGES that removes such multiple edges and  $B$ -configurations by flipping. To get to the heart of the algorithm more quickly, we defer this detour to later. Specifically, let us assume for now that  $G$  was 3-connected. We can then argue (Lemma 3.7) that  $G_t$  has no  $B$ -configuration and is simple.

**Lemma 3.7** *Let  $G$  be a simple 1-plane graph that is 3-connected and has no  $B$ -configuration or  $W$ -configuration. Then graph  $G_t$  obtained after line 4 of 1P-RVDRAWER is simple and has no planarized  $B$ -configuration.*

*Proof*  $G_t$  has no loops since  $G$  has none and none of the steps adds loops. Assume for contradiction that there is some double edge  $(u, v)$  in  $G_t$ . This must have been added by ADDKITEEDGES, since  $G$  is simple and the triangulation-step adds no multiple edges. It is not possible that the two copies of  $(u, v)$  form an inner face, because  $G$  (and hence  $G_p$ ) was simple, and we add an edge with ADDKITEEDGES only if it does not exist already at this place in the planar embedding. So the two copies either form the outer face or form no face at all. In the latter case,  $\{u, v\}$  is a separation pair of the planar graph  $G_t$ , and (as one easily shows) therefore also of  $G$ . This contradicts 3-connectivity. So multiple edges can only exist if the two copies form the outer face. At least one of the copies did not exist in the planarization  $G_p$ , and hence was added due to some dummy-vertex  $x$ . If the other copy already existed in  $G_p$ , then the crossing at  $x$  together with this copy forms a  $B$ -configuration in  $G$ . If the other copy did not



**Fig. 6** The 4-block tree  $\mathcal{T}$  of the triangulated plane graph  $G^+$  of Fig. 5 (b)

exist in  $G$ , then it was added due to some dummy-vertex  $y$ , and the crossings at  $x$  and  $y$  form a W-configuration. Both contradict the assumption.

So we know that  $G_t$  is simple. Now finally assume that  $G_t$  has a B-configuration, say at dummy-vertex  $x$  with neighbors  $a, b, c, d$  and with edge  $(a, b)$  such that triangle  $\Delta(a, b, x)$  contains  $c$  and  $d$  inside and hence is not a face. We also surrounded  $x$  by a kite, which adds an edge  $(a, b)$  for which  $\Delta(a, b, x)$  is a face. Hence there are two copies of edge  $(a, b)$  and  $G_t$  is not simple, a contradiction.  $\square$

*The Decomposition Technique.* Assume that we have reached line 8 of Algorithm 1, where graph  $G^+$  is defined. We claim that  $G^+$  is a triangulated simple plane graph that has the same vertices as  $G_p$ , contains all edges of  $G_p$ , and has no planarized B-configuration, W-configuration, and T-configuration. Furthermore, every dummy-vertex is inner and has degree at least 4. It is easy to see that this is true if  $G$  is 3-connected. Namely,  $G_t$  then has no multiple edges by Lemma 3.7, and so lines 5–7 were not executed, that is,  $G^+ = G_t$ . Moreover, we know that  $G_t$  satisfies the above conditions by Observation 3.6 and Lemma 3.7. We will discuss later (in Lemma 3.18) why this claim on  $G^+$  is true even if  $G$  is not 3-connected (and thus lines 5–7 are executed).

As next step, Algorithm 1P-RVDRAWER computes a decomposition of  $G^+$  into its 4-connected components. To explain this, we need a few definitions. Recall that  $G^+$  is a triangulated plane graph and therefore has a unique planar embedding. A *separating triangle* of  $G^+$  is a triangle  $T$  that has vertices both inside and outside  $T$ . Such a separating triangle naturally gives rise to two triangulated plane graphs; one consisting of all vertices inside  $T$  together with  $T$  and one consisting of all vertices outside  $T$  together with  $T$ . Repeat this operation in both resulting subgraphs until there are no separating triangles left. The resulting set of subgraphs are called the *4-connected components*. They are naturally organized in the the *4-block tree* of  $G^+$ , which is a tree  $\mathcal{T}$  defined as follows (see also [26] and Fig. 6). Every 4-connected component  $\mathcal{C}_v$  of  $G^+$  is represented by a node  $v$  in  $\mathcal{T}$ . There is an edge between two nodes  $\nu$  and  $\mu$  in  $\mathcal{T}$ , if there is a separating triangle that belongs to both  $\mathcal{C}_\nu$  and  $\mathcal{C}_\mu$ .



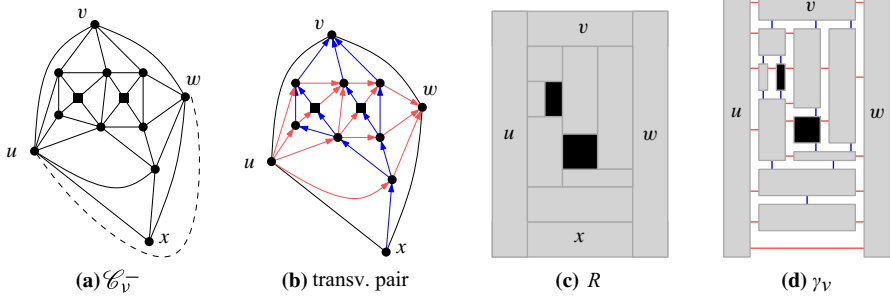
We root  $\mathcal{T}$  at the node  $\rho$  with the 4-connected component that contains the vertices and edges of the outer face (line 10 in Algorithm 1). Then for any parent  $\nu$  and child  $\mu$ , the separating triangle common to  $\mathcal{C}_\nu$  and  $\mathcal{C}_\mu$  is an inner face in  $\mathcal{C}_\nu$  and the outer face of  $\mathcal{C}_\mu$ . For example, in Fig. 6 the node  $\nu$  is the parent of  $\mu$ , since  $\Delta(u, y, z)$  is an inner face of  $\nu$  and the outer face of  $\mu$ . The 4-block tree  $\mathcal{T}$  can be computed in  $O(n)$  time [26].

We give here an outline of the remaining steps; details will be given in the next few pages. Algorithm 1P-RVDRAWER will use the 4-block tree  $\mathcal{T}$  as follows. It first traverses  $\mathcal{T}$  top-down and determines for each 4-connected component  $\mathcal{C}_\nu$  a special edge, called *surround-edge* (line 12 in Algorithm 1). It also computes an RVR  $\gamma_\nu$  of  $\mathcal{C}_\nu$  using a so-called *rectangular dual representation* [23, 28, 46] (line 13 in Algorithm 1). Next, the algorithm traverses  $\mathcal{T}$  bottom-up. Let  $\nu$  be a node of  $\mathcal{T}$  with  $h \geq 1$  children  $\mu_1, \dots, \mu_h$ . Denote by  $G_\nu$  the graph whose 4-block tree is the subtree of  $\mathcal{T}$  rooted at  $\nu$ . The already computed representations  $\Gamma_{\mu_1}, \dots, \Gamma_{\mu_h}$  of  $G_{\mu_1}, \dots, G_{\mu_h}$  are suitably merged into  $\gamma_\nu$  and an RVR  $\Gamma_\nu$  of  $G_\nu$  is obtained (line 16 in Algorithm 1). At the root  $\rho$  we obtain an RVR of  $G_\rho = G^+$ . We delete from it all edges that were added to triangulate the graph, resulting in an RVR of the planarization  $G_\rho$  of  $G$  (line 18 in Algorithm 1). In order to turn this in an RVR of  $G$ , we must un-planarize, i.e., replace every dummy-vertex by a crossing (line 19 in Algorithm 1). To this aim, we need each dummy-vertex to be drawn in a special way: its rectangle needs to have exactly one incident edge on each side. We call this the *4-sides-condition* and will examine throughout the drawing and merging steps whether it holds.

*Transversal Pairs of Bipolar Orientations.* We need a brief detour to define some concepts we need later. An *internally 4-connected plane graph*, also known as *irreducible triangulation* [22], is a plane graph where all inner faces are triangles and that has no separating triangle. Let  $G$  be an internally 4-connected planar graph for which the outer face is a 4-cycle. A *transversal pair of bipolar orientations* of  $G$ , also known as *regular edge labeling* [28] or simply *transversal structure* [21], assigns to each edge one of two colors<sup>2</sup>, say *red* and *blue*, and an orientation such that: (i) for each inner vertex  $v$ , the clockwise order of edges around  $v$  contains outgoing blue edges, then outgoing red edges, then incoming blue edges, then incoming red edges, and none of these sets is empty; (ii) any inner face has at least one red and at least one blue edge; (iii) let  $v_N, v_E, v_S, v_W$  be the four vertices on the outer face of  $G$ , in clockwise order, then all inner edges incident to  $v_N$  are blue incoming, all inner edges incident to  $v_E$  are red incoming, all inner edges incident to  $v_S$  are blue outgoing, and all inner edges incident to  $v_W$  are red outgoing; (iv) both the red and the blue graph are acyclic. See e.g. Fig. 7 (b), where  $v_N = v, v_E = w, v_S = x, v_W = u$ .

Such a transversal pair of bipolar orientations always exists (even if we arbitrarily pre-color the edges on the outer face) and can be computed in linear time [22, 28], presuming as before that the graph is internally 4-connected with a 4-cycle as outer face. The same papers also show how to convert, in linear time, a transversal pair of bipolar orientations into a *rectangular dual representation*. This is a representation that

<sup>2</sup> In all figures of the paper red edges appear in lighter gray than blue edges when printed b/w.



**Fig. 7** (a) The graph  $\mathcal{C}_v^-$  obtained from  $\mathcal{C}_v$  in Fig. 6. (b) A transversal pair of bipolar orientations for  $\mathcal{C}_v^-$ . (c) The corresponding rectangular dual representation (dummy-vertices are black squares). (d) Retracting the rectangles and adding the surround-edge

assigns to each vertex  $v$  of the graph  $G$  an axis-aligned rectangle  $R_v$ , such that every two rectangles are interior-disjoint, and the union of these rectangles is a rectangle without holes. Furthermore, for each edge  $(v, w)$ , the rectangles  $R_v$  and  $R_w$  share a positive-length part of their boundaries. Moreover, if the edge is directed  $v \rightarrow w$ , then the rectangle  $R_v$  is left (below) the rectangle  $R_w$  if and only if the edge  $(v, w)$  is red (blue). See Fig. 7 (c) for an illustration. We remark that the existence of such a representation was known much earlier already, see [46].

*Surround-Edges.* Consider a 4-connected component  $\mathcal{C}_v$  of  $G^+$ . We want to find an RVR of  $\mathcal{C}_v$ . To be able to merge later, we need to restrict this RVR further. We pick a so-called *surround-edge*  $e$  on the outer face with CHOOSESURROUNDEDGE (see Algorithm 3), and we want an RVR such that all edges incident to an endpoint of  $e$  are drawn horizontally.

**Algorithm 3:** Pseudocode for choosing the surround-edge.

```

1 CHOOSESURROUNDEDGE ( $\mathcal{C}_v$ )
2   let the outer face of  $\mathcal{C}_v$  be  $\Delta(u, v, w)$ 
3   if  $v$  has a parent  $\pi$  and the surround-edge  $e$  of  $\mathcal{C}_\pi$  is one of  $\{(u, v), (v, w), (u, w)\}$  then
4     let the surround-edge of  $\mathcal{C}_v$  be  $e$ 
5   end
6   else
7     foreach edge  $e$  in triangle  $\Delta(u, v, w)$  do
8       let  $f_e$  be the face of  $G^+$  that is incident to  $e$  and inside  $\Delta(u, v, w)$ 
9       let  $x_e$  be the vertex of  $f_e$  that is not an endpoint of  $e$ 
10      if  $x_e$  is an original vertex then
11        let the surround-edge of  $\mathcal{C}_v$  be  $e$  and break
12      end
13    end
14  end
    
```

We illustrate CHOOSESURROUNDEDGE on the example of Fig. 6. Consider graph  $\mathcal{C}_\phi$  whose outer face is  $\Delta(a, b, c)$ . The if-case does not apply since the surround-edge of the parent  $\mathcal{C}_\rho$  is on the outer face of  $\mathcal{C}_\rho$  and hence not in  $\{(a, b), (b, c), (c, a)\}$ . So we reach the else-case, say we consider first edge  $e = (a, c)$ . Its vertex  $x_e$  is the interior

vertex of  $\mathcal{C}_\phi$ , which is a dummy-vertex, hence we do not use  $(a, c)$  as surround-edge. Similarly we do not use  $(b, c)$ . For  $e' = (a, b)$ , the vertex  $x_{e'}$  is the interior vertex of  $\mathcal{C}_\omega$  (recall that face  $f_e$  is defined to be a face of  $G^+$ , *not* of the 4-connected component under consideration). This vertex is an original vertex and we hence choose  $(a, b)$  as surround-edge for  $\mathcal{C}_\phi$ . Now we proceed to the child  $\omega$ . Here the if-case applies:  $\omega$  has parent  $\phi$  and the surround-edge  $(a, b)$  of  $\mathcal{C}_\phi$  belongs to the outer face of  $\mathcal{C}_\omega$ . Therefore  $(a, b)$  becomes the surround-edge of  $\mathcal{C}_\omega$  as well.

**Lemma 3.8** *If  $G$  contains no T-configuration and no B-configuration, then for any 4-connected component  $\mathcal{C}_v$ , algorithm CHOOSESURROUNDEDGE finds a surround-edge  $e$ . Furthermore, if  $f_e$  is the face of  $G^+$  incident to  $e$  that is inside the triangle formed by the outer face of  $\mathcal{C}_v$ , then the vertex of  $f_e$  that is not an endpoint of  $e$  is an original vertex.*

*Proof* Assume first that CHOOSESURROUNDEDGE reaches the if-case (line 3 of Algorithm 3), i.e.,  $v$  has a parent  $\pi$ , and the surround-edge  $e$  of  $\mathcal{C}_\pi$  is one of the edges on the outer face  $\Delta(u, v, w)$  of  $\mathcal{C}_v$ , say  $e = (u, v)$ . We chose  $e$  as the surround-edge of  $\mathcal{C}_v$  in this case. We may assume (by induction on the depth of  $\mathcal{C}_v$  in the 4-block tree) that the lemma holds for  $\mathcal{C}_\pi$ . Let  $f_e$  be the face of  $G^+$  that is incident to  $e$  and inside the triangle that is the outer face of  $\mathcal{C}_\pi$ . Let  $x_e$  be the vertex  $\neq u, v$  on  $f_e$ ; we know that  $x_e$  is an original vertex since the claim holds for  $\mathcal{C}_\pi$ . Let  $f'_e$  be the face of  $G^+$  that is incident to  $e$  and inside the triangle that is the outer face of  $\mathcal{C}_v$ , and let  $x'_e \neq u, v$  be the third vertex of  $f'_e$ . The crucial insight is that all of  $\mathcal{C}_v$  is on or inside the outer face of  $\mathcal{C}_\pi$ . Since  $f_e$  and  $f'_e$  are faces of  $G^+$ , rather than faces of  $\mathcal{C}_\pi$  or  $\mathcal{C}_v$ , we therefore have  $f'_e = f_e$  and  $x'_e = x_e$ . Thus  $x'_e$  is an original vertex as desired.

Now assume that CHOOSESURROUNDEDGE reaches the else-case (line 6 of Algorithm 3). Let  $x_{uv}, x_{vw}$  and  $x_{uw}$  be the three vertices that serve as  $x_e$  for the three edges  $e \in \{(u, v), (v, w), (u, w)\}$ . We fail to choose a surround-edge only if all three of  $x_{uv}, x_{vw}$  and  $x_{uw}$  are dummy-vertices. We show that this is not possible. If they are all distinct and all dummy-vertices, then they would form a T-configuration. If they are all dummy-vertices and exactly two of them coincide, (say)  $x_{uv} = x_{vw}$  is distinct from  $x_{uw}$ , then in  $G$  the crossing at  $x_{uw}$  would form a B-configuration with the copy of  $(u, w)$  in  $G$  that must be involved in the crossing  $x_{uv} = x_{vw}$ . Not all three can coincide, else  $x_{uv} = x_{vw} = x_{uw}$  would be the only vertex in the graph rooted at  $v$  and have degree 3 and not be a dummy-vertex. So there exists at least one edge  $e$  of triangle  $\Delta(u, v, w)$  where  $x_e$  is an original vertex. Thus CHOOSESURROUNDEDGE chooses  $e$  as a surround-edge and  $x_e$  is an original vertex as required.  $\square$

*Drawing a 4-Connected Component.* We are now ready to show how to find an RVR of  $\mathcal{C}_v$  such that the surround-edge  $e$  and all edges of  $\mathcal{C}_v$  that are incident to one of its endpoints are drawn horizontally. If  $\mathcal{C}_v$  is  $K_4$ , then such an RVR is easily obtained (see also Fig. 10(b)), so we assume that  $\mathcal{C}_v$  has at least 5 vertices. This implies that any inner vertex of  $\mathcal{C}_v$  has degree at least 4, else its neighbors would form a separating triangle. Remove the surround-edge from  $\mathcal{C}_v$  to obtain graph  $\mathcal{C}_v^-$ ; see also Fig. 7(a). Since  $\mathcal{C}_v$  is 4-connected,  $\mathcal{C}_v^-$  is internally 4-connected. We now obtain an RVR  $\gamma_v$  of  $\mathcal{C}_v$  using a rectangular dual representation as follows:

**Lemma 3.9**  $\mathcal{C}_v^-$  has an RVR such that all edges incident to an endpoint of the surround-edge of  $\mathcal{C}_v$  are drawn horizontally and the 4-sides-condition holds at every inner vertex of degree 4.

*Proof* Since  $\mathcal{C}_v^-$  is internally 4-connected, it admits a transversal pair of bipolar orientations, and with it, a rectangular dual representation  $\mathcal{R}$  where blue and red edges correspond to shared vertical and horizontal (parts of) sides [22, 28]. We choose  $v_W$  and  $v_E$  to be the endpoints of the surround-edge, and color the outer edges red. As an illustration, Fig. 7 (b) shows the transversal pair of bipolar orientations of the planarization of the graph in Fig. 7 (a), and Fig. 7 (c) gives the corresponding rectangular dual representation.

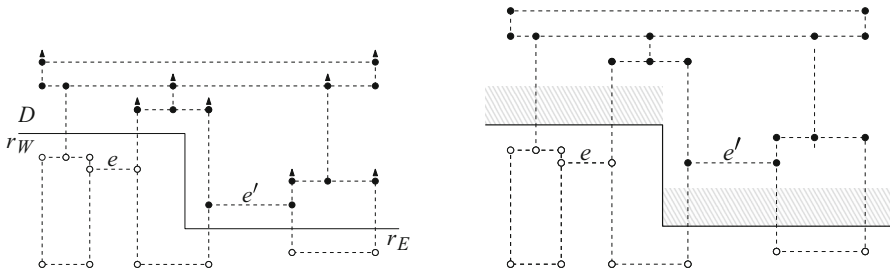
Now retract the rectangles a bit as follows. Let  $\varepsilon > 0$  be so small that all rectangles have width and height at least  $\varepsilon$  and the length of any shared side is at least  $\varepsilon$ . Replace each rectangle  $[x, x'] \times [y, y']$  by a slightly smaller rectangle  $[x + \frac{\varepsilon}{3}, x' - \frac{\varepsilon}{3}] \times [y + \frac{\varepsilon}{3}, y' - \frac{\varepsilon}{3}]$ . By choice of  $\varepsilon$  each rectangle still has positive width and height. For each edge of  $G$ , the shared side of length at least  $\varepsilon$  is replaced by a region between its two endpoints, not containing other vertices, that has dimension  $2\frac{\varepsilon}{3} \times \frac{\varepsilon}{3}$  or more. Hence for each edge we obtain a line of sight between the rectangles of its endpoints, and hence an RVR with thick lines of sight. Note that the line of sight is horizontal if and only if the shared rectangle-sides were vertical, so the horizontal and vertical lines of sight correspond to the red and blue edges of the transversal pair of bipolar orientations, respectively. See also Fig. 7 (d).

It remains to prove the claim on the 4-sides-condition. Let  $z$  be an inner vertex of degree 4, and let  $R_z$  be the rectangle of  $z$  in the rectangular dual. By properties of the transversal pair of bipolar orientations, the clockwise order of edges around  $z$  consists of incoming red edges, incoming blue edges, outgoing red edges, and outgoing blue edges. None of these sets is empty for an inner vertex, and, since  $\deg(z) = 4$ , there exists exactly one edge of each kind. In consequence, in the rectangular dual representation there is exactly one edge incident to each side of  $R_z$ , as desired.  $\square$

This gives an RVR of  $\mathcal{C}_v^-$ , to which we need to add the surround-edge. Say the vertices of the outer face of  $\mathcal{C}_v^-$  are  $u, v, w, x$  in clockwise order, with surround-edge  $(u, w)$  and  $x$  the vertex incident to the inner face at the surround-edge. The construction draws all edges of  $\mathcal{C}_v$  incident to  $u$  or  $w$  horizontally. After rotation, assume that  $u$  is left of  $w$ . We can now extend both  $u$  and  $w$  upwards or downwards beyond the drawing of  $x$ , and insert here a horizontal segment for the surround-edge such that  $\Delta(u, v, x)$  becomes an inner face of the drawing. This gives an embedding-preserving RVR  $\gamma_v$  of  $\mathcal{C}_v$ . See also Fig. 7 (d) for an illustration.

**Lemma 3.10** In the RVR of  $\mathcal{C}_v$ , the 4-sides-condition holds for every dummy-vertex that is an inner vertex of  $\mathcal{C}_v$  and has degree 4 in  $\mathcal{C}_v$ .

*Proof* The claim holds vacuously if  $\mathcal{C}_v$  is  $K_4$ , so assume it is not. Let  $z$  be a dummy-vertex that is an inner vertex of  $\mathcal{C}_v$  with degree 4. Deleting the surround-edge does not change the degree of  $z$ , therefore  $\deg(z) = 4$  even in  $\mathcal{C}_v^-$ . If  $z$  is an inner vertex of  $\mathcal{C}_v^-$ , then the construction for  $\mathcal{C}_v^-$  ensures that all four sides of  $z$  receive an edge. The only vertex that is an inner vertex of  $\mathcal{C}_v$ , but not of  $\mathcal{C}_v^-$ , is the vertex  $x$  incident to the inner



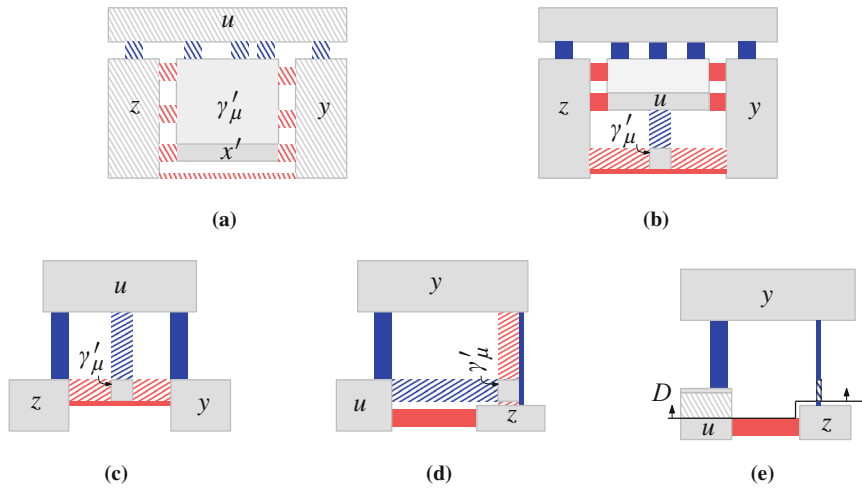
**Fig. 8** A zig-zag-slide (adapted from [5]). Black points move upward while white points remain stationary. With this amount of sliding, edges  $e$  and  $e'$  become aligned

face at the surround-edge. However, by Lemma 3.8 vertex  $x$  is not a dummy-vertex, so  $z \neq x$ . □

*Zig-Zag-Slides.* Both for the merging step and for undoing the planarization, we need a method of modifying the drawing such that some items are moved while others are stationary. This can be achieved with what was called *creating a channel* in [45] and *zig-zag-bend-elimination-slide* in [5], we call it a *zig-zag-slide* for short. We briefly review this here. Assume that  $\Gamma$  is an RVR and we have a dividing curve  $D$  as follows:  $D$  consists of some vertical line segment  $s$  that intersects no horizontal element of the drawing, i.e., neither a horizontal edge nor a horizontal boundary of a rectangle. At the top end of  $s$ , attach a leftward horizontal ray  $r_W$ . At the bottom end of  $s$ , attach a rightward horizontal ray  $r_E$ . No conditions are being put onto  $r_W$  and  $r_E$ . Define the *region above  $D$*  to be the set of all points that are in the  $x$ -range of  $r_W$  and strictly above  $r_W$ , and all points in the  $x$ -range of  $r_E$  and on or above  $r_E$ .

Now slide all points in the region above  $D$  upwards by some amount  $\delta > 0$ . This maintains a rectangle visibility representation, after lengthening vertical edges and rectangle borders suitable, because any horizontal segment either stays stationary or moves in its entirety (recall that no horizontal segments cross  $s$ ). In particular, for any point  $p$  above  $r_E$  and any point  $q$  below  $r_W$  but with a larger  $y$ -coordinate, a shift by  $y(q) - y(p)$  achieves that these points become horizontally aligned. See Fig. 8. We will use such zig-zag-slides for this shape of  $D$  as well as for the three other shapes achieved by flipping the picture horizontally and/or rotating  $90^\circ$ .

*Merging Components.* Let  $\mu$  be a child of  $v$  in  $\mathcal{T}$ , and recall that we already obtained drawings  $\gamma_v$  of  $\mathcal{C}_v$  and (recursively)  $\gamma_\mu$  of the graph  $G_\mu$  consisting of  $\mathcal{C}_\mu$  and the components at its descendants. We now want to create a drawing of  $G_\mu \cup \mathcal{C}_v$ , i.e., the graph containing all vertices and edges from  $G_\mu$  and  $\mathcal{C}_v$ . The common vertices  $u, y, z$  of  $G_\mu$  and  $\mathcal{C}_v$  form the outer face of  $G_\mu$  and an inner face of  $\mathcal{C}_v$ . See Fig. 6. The outer face of  $\mathcal{C}_\mu^-$ , consists of  $\{u, y, z\}$  as well as a fourth vertex, say  $x'$ ; after possible renaming of  $\{u, y, z\}$  we may assume that the clockwise order around the outer face of  $\mathcal{C}_\mu^-$  is  $u, y, x', z$ . In particular, drawing  $\gamma_\mu$  contains (up to rotation) node  $u$  on the top, node  $y$  on the right, the surround-edge  $(y, z)$  and node  $x'$  at the bottom, and node  $z$  on the left. Let  $\gamma'_\mu$  be the drawing obtained from  $\gamma_\mu$  by deleting  $u, y, z$ . It suffices



**Fig. 9** (a) Illustration of  $\gamma_\mu$  and  $\gamma'_\mu$ . (b) Illustration for the proof of Lemma 3.11. (c–e) Illustration for the proof of Lemma 3.12. (c)  $u$  has two vertical edges. (d)  $y$  has two vertical edges,  $u$  has a higher top than  $z$ . (e) Slide so that  $u$  has a higher top than  $z$

to merge  $\gamma'_\mu$ , since  $u, y, z$  and the edges between them are represented in  $\gamma_\nu$  already. See Fig. 9(a). Triangle  $\Delta(u, y, z)$  is an inner face of  $\mathcal{C}_\nu$ . The challenge is now to find a region in  $\gamma_\nu$  into which  $\gamma'_\mu$  will “fit”. Put differently, we want to find a region within  $\gamma_\nu$  that is inside the face  $\Delta(u, y, z)$  and that is (after possible rotation) below  $u$ , to the left of  $y$  and to the right of  $z$ . We call such a region, a *feasible region* for  $\gamma_\mu$  in  $\gamma_\nu$ . As we have no control over how triangle  $\Delta(u, y, z)$  is drawn in  $\gamma_\nu$ , the existence of a feasible region is non-trivial, and may in fact require modifying  $\gamma_\nu$  slightly. We start with a simple case:

**Lemma 3.11** *Assume that the surround-edge of  $\mathcal{C}_\nu$  also belongs to  $\mathcal{C}_\mu$ . Then there exists a feasible region for  $\gamma_\mu$  in  $\gamma_\nu$ .*

*Proof* If the surround-edge  $e$  of  $\mathcal{C}_\nu$  belongs to  $\mathcal{C}_\mu$ , then CHOOSESURROUNDEDGE chose it to be the surround-edge for  $\mathcal{C}_\nu$  as well. Say edge  $(y, z)$  is the surround-edge for both  $\mathcal{C}_\mu$  and  $\mathcal{C}_\nu$ . In  $\gamma_\nu$ ,  $(y, z)$  was drawn (after possible rotation) bottommost, with  $u$  above it. A feasible region is now found in the small strip above the drawing of edge  $(y, z)$ . See Fig. 9(b) for an illustration.  $\square$

We now show how to find a feasible region for  $\gamma_\mu$  in  $\gamma_\nu$  in the general case.

**Lemma 3.12** *A feasible region for  $\gamma_\mu$  in  $\gamma_\nu$  always exists, possibly after modifying  $\gamma_\nu$  without changing angles or incidences.*

*Proof* We already argued this if  $\Delta(u, y, z)$  includes the surround-edge  $e$  of  $\mathcal{C}_\nu$ , so assume that this is not the case. Then face  $\Delta(u, y, z)$  of  $\mathcal{C}_\nu$  is also a face of  $\mathcal{C}_\nu^- = \mathcal{C}_\nu - e$ . The drawing of  $\mathcal{C}_\nu^-$  was obtained via the transversal pair of bipolar orientations of  $\mathcal{C}_\nu^-$ , with red and blue edges corresponding to horizontal and vertical edges, respectively.

Since any inner face in a transversal pair of bipolar orientations has at least one red and at least one blue edge,  $\Delta(u, y, z)$  is drawn with at least one horizontal and at least one vertical edge. Say two edges are vertical and one is horizontal (the other case is similar). Now consider which vertex of  $u, y, z$  is the one where both incident edges of triangle  $\Delta(u, y, z)$  are vertical.

If vertex  $u$  has two incident vertical edges  $(u, y)$  and  $(u, z)$ , then  $(y, z)$  is drawn horizontally. We can then find a feasible region within the thick line of sight of edge  $(y, z)$ . See Fig. 9(c) for an illustration.

Now assume that  $y$  has two incident vertical edges  $(y, u)$  and  $(y, z)$  (the case of vertex  $z$  is symmetric). For ease of description, assume that  $y$  is above  $u$  and  $z$ , and  $u$  is to the left of  $z$ . If the top side of  $u$  is strictly above the top side of  $z$ , then it is easy to find a feasible region within the thick line of sight of edge  $(y, z)$  into which we can merge  $\gamma'_v$  after rotating it; see Fig. 9(d). If the top side of  $u$  is below (or at the same height as) the top side of  $z$ , then we modify the drawing to create a suitable region using a zig-zag-slide. The dividing curve  $D$  is defined as follows. Insert a vertical edge just left of  $z$ , ranging from just above  $(u, z)$  (hence below the top of  $u$  since we have thick lines of visibility) to just above the top of  $z$ . Expand it with a horizontal ray leftward from the lower end and a horizontal ray rightward from the upper end. By sliding upward sufficiently far, we hence achieve that the top of  $u$  is above the one of  $z$ , which allows us to find a feasible region as previously. See Fig. 9(e) for an illustration. □

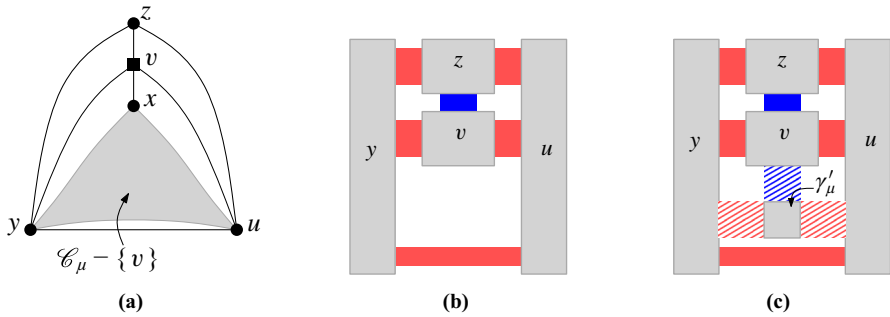
By Lemma 3.12, for every child  $\mu$  of  $v$  we can find a feasible region into which we merge (after suitable scaling) the drawing  $\gamma_\mu - \{u, y, z\}$ . This gives the RVR of  $\mathcal{C}_v \cup G_\mu$ . It remains to prove the 4-sides-condition for the inner vertices of  $\mathcal{C}_v \cup G_\mu$ .

**Observation 3.13** *In the RVR of  $\mathcal{C}_v \cup G_\mu$ , the 4-sides-condition holds for every dummy-vertex that is an inner vertex of  $\mathcal{C}_v \cup G_\mu$  and has degree 4.*

*Proof* Let  $v$  be an inner dummy-vertex of  $\mathcal{C}_v \cup G_\mu$ . If  $v$  is an inner vertex of either  $\mathcal{C}_v$  or  $G_\mu$ , then the statement follows by Lemma 3.10 (or induction on the height of  $\mathcal{T}$ ). Else,  $v$  must be on the outer face of  $\mathcal{C}_\mu$ , and hence be part of a separating triangle  $\Delta(u, y, v)$  which is an inner face of  $\mathcal{C}_v$ .

Vertex  $v$  must have neighbors both inside and outside triangle  $\Delta(u, y, z)$  by 3-connectivity of the triangulated plane graph  $G^+$ . Since dummy-vertices have degree 4, it therefore has exactly one neighbor, say  $x$ , in  $\mathcal{C}_\mu - \{u, y, v\}$  and exactly one neighbor, say  $z$ , in  $\mathcal{C}_v - \{u, y, v\}$ . Further, these neighbors appear in clockwise order  $z, u, x, y$  around  $v$  and form a kite (see Fig. 10(a)). It follows that  $\Delta(u, y, z)$  is also a separating triangle (or the outer face of  $G^+$ ) and that  $\Delta(v, z, u)$  and  $\Delta(v, y, z)$  are faces due to the kite. Therefore  $\mathcal{C}_v$  is  $K_4$ , with  $\mu$  the only child of  $v$ . Similarly,  $\Delta(u, x, y)$  is either a separating triangle or an inner face of  $G^+$ , and  $\mathcal{C}_v$  is also a  $K_4$  with at most one child (call it  $\omega$  if it exists).

When choosing the surround-edge of  $\mathcal{C}_v$ , we cannot use  $(u, z)$  or  $(y, z)$ , because the inner faces at them contain the dummy-vertex  $v$ . Therefore the surround-edge of  $\mathcal{C}_v$  is  $(y, u)$ , and by our choice of surround-edge therefore  $(y, u)$  also becomes the surround-edge of  $\mathcal{C}_\mu$  and (if it exists)  $\mathcal{C}_\omega$ . So Lemma 3.11 is used to find the feasible region for merging  $G_\mu$  into  $\mathcal{C}_v$ . Also, the drawing of  $\mathcal{C}_v$  is fixed since it is  $K_4$  (see



**Fig. 10** Illustration for the proof of Observation 3.13

also Fig. 10(b)). Inspecting the construction (see also Fig. 10(c)) shows that during this merge  $v$  obtains an incident edge on each side as desired.  $\square$

After repeating the process for the entire 4-block tree we obtain the desired RVR  $\Gamma^+$  of  $G^+$ . After every merging step inner dummy-vertices of degree 4 satisfy the 4-sides-condition. Once we have merged all subgraphs, all inner dummy-vertices have degree 4, and we get the following result:

**Observation 3.14** *In the RVR of  $G^+$ , the 4-sides-condition holds for every dummy-vertex that is an inner vertex of  $G^+$  and has degree 4.*

*Undoing the Planarization.* Once all merging is completed, the drawing  $\Gamma_\rho$  at the root gives an RVR  $\Gamma^+$  of  $G^+$ . To turn this into an RVR of  $G$ , we must undo the planarization. To do so, first remove from  $\Gamma^+$  all edges that belonged to  $G^+$  but not to the planarization  $G_p$ . Let  $\Gamma_p$  be the resulting RVR of  $G_p$ .

The next step is to re-insert the crossings in place of the dummy-vertices. Consider any dummy-vertex  $z$  and let the *cross-edges* at  $z$  be the four edges incident to  $z$  in  $G_p$  that replaced the crossing edges. To replace  $z$  with a crossing, we need that the four cross-edges of  $z$  attach to the four sides of the rectangle of  $z$ . This is not immediately obvious (even in the light of Observation 3.14), because  $z$  may have degree exceeding 4 in  $G^+$  if  $G$  is not 3-connected. We say that an RVR of  $G^+$  satisfies the *extended 4-sides-condition* if for every dummy-vertex  $z$  exactly one cross-edge of  $z$  attaches on each side of the rectangle  $R_z$  of  $z$ . If  $G$  is 3-connected, then we use  $G^+ = G_t$ , so every dummy-vertex  $z$  has degree 4 in  $G^+$  by Observation 3.6. By Observation 3.14 then the four cross-edges of  $z$  are on the four sides of  $R_z$ , and therefore we have:

**Observation 3.15** *If  $G$  is 3-connected, then the RVR  $\Gamma^+$  of  $G^+$  satisfies the extended 4-sides-condition.*

If  $G$  is not 3-connected, then we will ensure later directly that the RVR of  $\Gamma^+$  satisfies the extended 4-sides-condition. Thus after deleting added edges we can now unplanarize the resulting RVR  $\Gamma_p$  of  $G_p$ , because for any dummy-vertex  $z$  the four cross-edges attach on the four sides of the rectangle  $R_z$  of  $z$ .



Let  $e_W, e_N, e_E$  and  $e_S$  be the four edges incident at the west, north, east and south side of  $R_z$ , respectively. If the edges  $e_W$  and  $e_E$  have the same  $y$ -coordinate, and the edges  $e_N$  and  $e_S$  have the same  $x$ -coordinate, then we can simply remove  $R_z$ , extend the edges, and obtain the desired crossing.

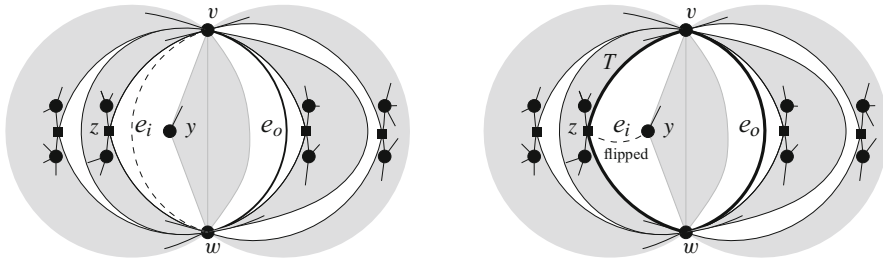
So assume that the edges  $e_W$  and  $e_E$  have different  $y$ -coordinates, with (say) the  $y$ -coordinate  $y_W$  of  $e_W$  larger than the  $y$ -coordinate  $y_E$  of  $e_E$ . Construct a dividing curve  $D$  by starting with a vertical line segment inside  $R_z$ , ranging from  $y$ -coordinate  $y_W + \varepsilon$  to  $y$ -coordinate  $y_E - \varepsilon$ , where  $\varepsilon > 0$  is chosen small enough that the segment is inside  $R_z$ . Attach a leftward horizontal ray at the top and a rightward horizontal ray at the bottom. Now apply a zig-zag-slide of length  $y_W - y_E$ . This moves  $e_E$  upward while keeping  $e_W$  stationary, and hence aligns the two edges. See also Fig. 8, where this is illustrated for  $e_W = e$  and  $e_E = e'$ . With another zig-zag-slide (with the dividing curve using a horizontal line segment and vertical rays), we similarly can align  $e_N$  and  $e_S$ , if needed. After this, remove the dummy-vertex to obtain the crossing. Repeating this at all dummy-vertices, and finally deleting all added edges, gives the RVR of  $G$ , as no crossings are created except where dummy-vertices were removed. This ends the description of the last step of algorithm 1P-RVDRAWER.

*General 1-Plane Graphs.* We previously assumed that the input graph  $G$  is 3-connected, chiefly so that ADDKITEEDGES does not create multiple edges and B-configurations. We now explain how to handle an arbitrary input graph, and in particular, how to flip edges that are multiple copies. This operation of flipping edges is the only change that is required; all other steps operate exactly the same. However, there are two non-trivial things to show to argue that this is feasible. First, we must argue that after flipping edges, the resulting graph satisfies the condition on  $G^+$  required after completion of line 8 of Algorithm 1. Second, we must argue that the rectangle visibility representation obtained for  $G^+$  satisfies the extended 4-sides-condition.

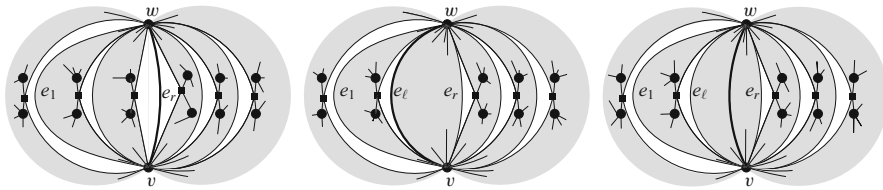
So let  $G$  be a simple (not necessarily 3-connected) 1-plane graph with no B-configuration, no W-configuration, and no T-configuration. We planarize and triangulate  $G$  as before to obtain  $G_t$ . Note that neither ADDKITEEDGES nor the triangulation-step adds any loop, hence  $G_t$  is a triangulated graph without loops and consequently 2-connected. For any pair of vertices in  $G_t$  with multiple edges between them, we execute FLIPMULTIEDGES (explained below) which removes multiple edges without creating new ones. We first clarify this flipping-operation. Consider an edge  $e = (v, w)$  in a triangulated planar graph, and let the two vertices *facing*  $e$  be the two vertices  $y, z$  that are not endpoints of  $e$  but are on a face incident to  $e$ . The operation of *flipping*  $e$  consists of removing  $e$  from the graph and replacing it with an edge  $(y, z)$ , routed within the degree-4 face that resulted from removing  $e$ . See edge  $e_t$  in Fig. 11. Note that the resulting graph is again a triangulation.

**Observation 3.16** *Let  $e = (v, w)$  be a multiple edge of  $G_t$ . Then flipping  $e$  does not create a new multiple edge, and does not create a B-configuration, W-configuration or T-configuration.*

*Proof* We know from Observation 3.5 that  $v$  and  $w$  are original vertices. Let  $y, z$  be the vertices facing  $e$ . Let  $e'$  be another copy of edge  $(v, w)$ . Since  $G_t$  is triangulated,



**Fig. 11** Flipping edge  $e_i = (v, w)$ . Note that  $\Delta(v, w, z)$  forms a separating triangle for the facing vertex  $z$



**Fig. 12** Three examples of multiple edges, in the three possible configurations  $r = \ell$ ,  $r = \ell + 1$  and  $r = \ell + 2$ . Edge  $e_o$  is bold

$e$  and  $e'$  did not bound one face, and so the 2-cycle  $e, e'$  enclosed exactly one of  $y$  and  $z$ . In consequence, by planarity there can be no connection in  $G_p$  from  $y$  to  $z$  except through  $v$  or  $w$ . Therefore no edge  $(y, z)$  can exist in  $G_t$ , which means that flipping  $e$  does not create a multiple edge. Also no dummy-vertex can be adjacent to both  $y$  and  $z$ , which means that the new edge  $(y, z)$  cannot create a B-configuration. Finally the new edge  $(y, z)$  is non-crossed, which means that the resulting graph has no new W-configuration or T-configuration.  $\square$

We now explain how Algorithm FLIPMULTIEDGES (Algorithm 4) chooses which copies of  $(v, w)$  to flip. At most one copy of  $(v, w)$  may have existed in  $G$  as well; we call this the *original copy* and all other copies *added copies*. By Observation 3.5 (and because the triangulation step does not add multiple edges) any added copy has at least one dummy-vertex as facing vertex.

Enumerate the copies of  $(v, w)$  as  $e_1, \dots, e_s$  in clockwise order around  $v$  in such a way that the 2-cycle  $e_1, e_s$  encloses all other copies. We say that  $e_i$  has a *dummy on the left (right) side* if the vertex that faces  $e_i$  and is before (after)  $e_i$  in the order around  $v$  is a dummy-vertex. Set  $\ell$  to be the maximal index where  $e_\ell$  has a dummy on the left side (set  $\ell = 0$  if there is no such copy). Set  $r$  to be the minimal index where  $e_r$  has a dummy on the right side (set  $r = s + 1$  if there is no such copy). See Fig. 12 for an illustration. Set  $o$  to be the index of the original copy if there is one. If there is no original copy, then set  $o = \ell$  if  $\ell \geq 1$  and  $o = r$  otherwise. Algorithm FLIPMULTIEDGES simply flips all edges  $(v, w)$  except  $e_o$ .

We need an observation that will be crucial later to argue that 1P-RVDRAWER can handle the graph obtained from flipping all edges.

**Observation 3.17** *Let  $z$  be a dummy-vertex facing a copy  $e_i$  of  $(v, w)$  that was flipped, resulting in an edge  $e'_i$  incident to  $z$ . Let  $e_o$  be the copy of  $(v, w)$  that was not flipped.*

---

**Algorithm 4:** Pseudocode for flipping edges.

---

```

1 FLIPMULTIEDGES ( $v, w$ )
2   if there exists an original copy of ( $v, w$ ) then
3     | let  $e_o$  be this copy
4   end
5   else
6     | choose  $e_o$  among the added copies of ( $v, w$ ) as explained in the text
7   end
8   foreach copy  $e$  of ( $v, w$ ) that is not  $e_o$  do
9     | let  $y, z$  be the two vertices facing  $e$ 
10    | remove edge  $e$  and insert edge ( $y, z$ ) in the resulting degree-4 face
11  end

```

---

Then the cross-edges  $(z, v)$  and  $(z, w)$ , together with edge  $e_o$ , form a separating triangle  $T$  with  $e'_i$  inside  $T$  and all cross-edges of  $z$  on or outside  $T$ .

*Proof* Since  $z$  faced  $e_i = (v, w)$ , the three edges  $(z, v)$ ,  $e_i$ ,  $(z, w)$  form a face of  $G_t$ . Thus after flipping  $e_i$  the edges  $(z, v)$ ,  $e'_i$ ,  $(z, w)$  are consecutive at  $z$ . Let  $T$  be the triangle with edges  $(z, v)$ ,  $e_o$ ,  $(w, z)$ . Triangle  $T$  is separating since  $e'_i$  is on one side of  $T$  and all cross-edges of  $z$  are on  $T$  or on the other side of  $T$ , due to the order of edges at  $z$ .

It remains to show that  $e'$  is inside  $T$ . To argue this, we need some observations. Enumerate the copies of  $(v, w)$  as  $e_1, \dots, e_s$  and define  $\ell, r$  as explained above.

- We have  $\ell \leq r$ . This trivially holds if  $\ell = 0$  or  $r = s + 1$ , so assume that edges  $e_\ell$  and  $e_r$  exist. If  $\ell > r$ , then the dummy-vertex to the right of  $e_r$ , together with the dummy-vertex to the left of  $e_\ell$ , form a W-configuration, a contradiction. So  $\ell \leq r$ .
- We have  $\ell \leq o \leq r$ . Assume for contradiction that  $o < \ell$ , which means that  $e_o$  exists in  $G$ . Then  $e_o$ , together with the dummy-vertex to the left of edge  $e_\ell$ , forms a B-configuration, a contradiction. Similarly  $o > r$  leads to a B-configuration and hence a contradiction.
- We have  $r \leq \ell + 2$ . For if  $r \geq \ell + 3$ , then both  $e_{\ell+1}$  and  $e_{\ell+2}$  have no dummy-vertex facing on either side. But then neither of them can be an added copy, which contradicts simplicity of  $G$ .

There are hence only three possibilities for  $r - \ell$  (see also Fig. 12). We may have  $r = \ell$ . In this case,  $e_o = e_\ell = e_r$ . We may have  $r = \ell + 1$ . In this case,  $e_o = e_\ell$  or  $e_o = e_r$ . We may have  $r = \ell + 2$ . In this case, edge  $e_o = e_{\ell+1}$  since  $e_{\ell+1}$  has no facing incident dummy-vertex.

Now consider again edge  $e_i$ . Since  $e_i$  was flipped while  $e_o$  was not, we have  $i \neq o$ , say  $i < o$  (the other case is symmetric). In all three situations discussed above,  $i < o$  implies  $i < r$ , therefore  $e_i$  has no dummy-vertex on its right side and  $z$  is on the left side of  $e_i$ . By  $i < o$  therefore the clockwise order at  $v$  contains  $(v, z)$  before  $e_i$  before  $e_o$ . Therefore triangle  $T$  has  $e_i$  (and hence also  $e'_i$ ) on the inside.  $\square$

As in line 8 of 1P-RVDRAWER, let  $G^+$  be the graph that results from  $G_t$  by applying FLIPMULTIEDGES at all multiple edges. We claim that this graph  $G^+$  satisfies all the required conditions.

**Lemma 3.18** *Let  $G$  be a 1-plane graph with no B-configuration, no W-configuration, and no T-configuration. The graph  $G^+$  obtained after line 8 of Algorithm 1 is a triangulated simple plane graph that has the same vertices as  $G_p$ , contains all edges of  $G_p$ , and has no planarized B-configuration, W-configuration, and T-configuration. Furthermore, every dummy-vertex is inner and has degree at least 4.*

*Proof* Clearly  $G^+$  is triangulated and plane and obtained from  $G_p$  by adding edges, since  $G_t$  satisfies these conditions and flipping edges does not violate any of them. Also no edge to be flipped can be on the outer face, and all edges to be flipped connect original vertices, so after applying the flipping operations all dummy-vertices are still inner vertices and their degrees only increased or remained the same. We flip all multiple edges creating neither new ones nor loops, so  $G^+$  is simple. Finally  $G^+$  has no W-configuration and no T-configuration since  $G_t$  had none by Observation 3.6 and flipping edges does not create any by Observation 3.16.

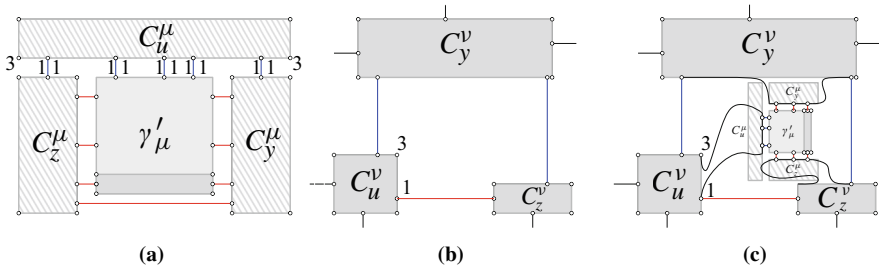
Thus the only remaining claim is that  $G^+$  has no planarized B-configuration. Assume for contradiction that it had one, consisting of an edge  $e_j = (v, w)$  and a dummy-vertex  $z$ . Edge  $e_j$  cannot be original, else this would give a B-configuration in  $G$ . Also edge  $e_j$  cannot have  $z$  as facing vertex, else  $e_j$  and  $z$  would not form a B-configuration. Hence during ADDKITEEDGES, we added another copy  $e_i$  of  $(v, w)$  such that  $z$  becomes a facing vertex of  $(v, w)$ . In particular,  $(v, w)$  was a multiple edge in  $G_t$ , and we flipped  $e_i$  while  $e_j$  was not flipped. By Observation 3.17, the triangle  $T$  formed by  $(z, v)$ ,  $(z, w)$  and edge  $e_j$  is such that the cross-edges of  $z$  are on or outside  $T$ . But the definition of a B-configuration says that the cross-edges of  $z$  must be on or inside  $T$ . Only two of the four cross-edges can be on  $T$ , so this is a contradiction and  $G^+$  has no planarized B-configuration.  $\square$

Thus lines 9-17 of 1P-RVDRAWER can proceed exactly as in the case where  $G$  is 3-connected, and create an RVR  $\Gamma^+$  of  $G^+$ . It remains to show that we can unplanarize  $\Gamma^+$ , for which the following suffices.

**Lemma 3.19** *The RVR  $\Gamma^+$  of  $G^+$  obtained after line 17 of 1P-RVDRAWER satisfies the extended 4-sides-condition at all dummy-vertices.*

*Proof* Consider a dummy-vertex  $z$  of  $G^+$ . If  $z$  has degree 4 in  $G^+$  then its four edges in  $G^+$  are the four cross-edges and are on the four sides by Observation 3.14, and so the extended 4-sides-condition holds at  $z$ . So now assume  $\deg(z) > 4$ , which implies that some multiple edge  $(v, w)$  had  $z$  as a facing vertex and was flipped to become an edge  $e'$  incident to  $z$ . By Observation 3.17, then there exists a separating triangle  $T$  with  $e'$  inside  $T$  and all cross-edges of  $z$  on or outside  $T$ . 1P-RVDRAWER creates the RVR  $\Gamma^+$  by splitting the graph at  $T$ . By induction, the RVR  $\Gamma_v$  of the graph  $G_v$  outside  $T$  satisfies the extended 4-sides-condition. In particular, in  $\Gamma_v$  all four sides of the rectangle  $R_z$  of  $z$  have exactly one incident cross-edge. The algorithm then merges into  $\Gamma_v$  the RVR of the graph inside  $T$ . This does not change edge-directions, so the resulting RVR of  $G^+$  still has exactly one incident cross-edge on each side of  $R_z$ , and the extended 4-sides-condition holds at  $z$  in  $\Gamma^+$ .  $\square$

Thus we can unplanarize as before, and obtain the desired RVR. We summarize:



**Fig. 13** (a)  $\gamma_\mu$  of Fig. 9(a) as an orthogonal representation. Only selected face-angles are listed. (b)  $\gamma_\nu$  of Fig. 9(d) as orthogonal representation. (c) Combining the two orthogonal representations into one

**Lemma 3.20** *Let  $G$  be a 1-plane graph with no B-configuration, no W-configuration, and no T-configuration as a subgraph. Then algorithm 1P-RVDRAWER computes a 1-plane RVR of  $G$ .*

The reader may be curious where exactly we used that  $G$  has no B-configuration, W-configuration, or T-configuration as a subgraph. The absence of T-configurations and B-configurations was used when choosing a surround-edge. This was done after the graph had been triangulated and planarized, so we must be careful to triangulate such that no T-configuration or B-configuration is introduced. This used that we had no W-configuration (e.g. in Lemma 3.7).

Corollary 3.4 and Lemma 3.20 imply Theorem 3.1. The next two sections prove Theorem 3.2.

### 3.3 Area and Time Complexity

Algorithm 1P-RVDRAWER, as described, does not have linear time complexity, since we repeatedly change the entire drawing, especially when applying zig-zag-slides. Also, coordinates may get very small when merging, resulting (after re-scaling to be integral) in a very large area. To avoid both these issues, we modify 1P-RVDRAWER and store RVRs implicitly. This requires the following changes:

- We obtain the RVR  $\gamma_\nu$  of each 4-connected component  $\mathcal{C}_\nu$  exactly as before, using explicit coordinates.
- We then convert  $\gamma_\nu$  into an orthogonal representation  $(H_\nu, \alpha_\nu, \beta_\nu)$  using Observation 2.2. Recall that in this orthogonal representation the graph  $H_\nu$  is obtained from  $\mathcal{C}_\nu$  by replacing every vertex  $u$  by a cycle  $C_u^\nu$ .
- Consider one merging step, say 1P-RVDRAWER wants to merge drawing  $\gamma_\mu$  into drawing  $\gamma_\nu$  at the separating triangle  $\Delta(u, y, z)$ . In the two orthogonal representations corresponding to  $\gamma_\mu$  and  $\gamma_\nu$ , we have cycles  $C_u^\mu$  and  $C_u^\nu$  representing vertex  $u$ . To do the merge, all that is required is to split the two cycles and to combine them into one cycle while maintaining the face-angle values. See the illustration in Fig. 13. In fact, the place to split the cycles is easily found from the orthogonal representation: both cycles must be split at the place where  $(u, z)$  and  $(u, y)$  attach, or one edge further if there is a face-angle of 3 (corresponding to a corner of the

- rectangle of  $u$ ). Similarly we combine the cycles for  $y$  and  $z$ . This gives exactly the orthogonal representation of the drawing that we would have obtained with 1P-RVDRAWER, and can be done in constant time per merge.
- We hence obtain an assignment of face-angles to the graph obtained from  $\mathcal{C}_v \cup \mathcal{C}_\mu$  by replacing vertices by cycles. These face angles are exactly the same as the face-angles that we would obtain if we used the RVR that algorithm 1P-RVDRAWER would create for  $\mathcal{C}_v \cup \mathcal{C}_\mu$ . Hence this assignment of face-angles is again an orthogonal representation, i.e., satisfies the sum-condition for the faces.
  - We want to convert the orthogonal representation of  $G^+$  into an orthogonal representation  $G_p$ . Recall that  $G^+$  is obtained from  $G_p$  by adding edges, so we must delete these edges. Deleting an edge can easily be done in an orthogonal representation without affecting its realizability; all that is required is to update at each end of the deleted edge the face-angles by adding the two face-angles at that edge.
  - So now we have an orthogonal representation of  $G_p$ . We now explain how to do the un-planarization without zig-zag-slide, and for this, need to analyze the structure at a dummy-vertex  $z$ .

The face-angles that we have computed mirror exactly the face-angles that would occur in the drawing created by 1P-RVDRAWER. Therefore the extended 4-sides-condition holds for  $z$ , and we have exactly one incident edge on each side of the rectangle of  $z$ . In terms of the orthogonal representation of  $G_p$ , this means that the face-angles at the cycle  $C_z$  representing  $z$  satisfy the following: First we have a degree-2 vertex with a face-angle of 3 (corresponding to  $270^\circ$ ) at the outside of  $C_z$ ; this represents one of the corners. Then we have a number (possibly 0) of degree-2 vertices with a face-angle of 2 (corresponding to  $180^\circ$ ), which may have been added if in  $G^+$  vertex  $z$  had more than four incident edges. Then we have one vertex of degree 3, corresponding to where one cross-edge of  $z$  attaches at the rectangle of  $z$ . Here hence there are two face-angles 1 outside  $C_z$ , one on each side of the cross-edge. Then we have again a number of degree-2 vertices with face-angle 2. This pattern repeats three more times, once for each side of the rectangle of  $z$ .

Now consider a face  $f$  that is incident to  $z$ , i.e., lies between two consecutive cross-edges at  $z$ . The face-angles at  $C_z$  that belong to  $f$  hence form a sub-sequence  $12^*32^*1$ , where  $2^*$  indicates that the 2 may repeat arbitrarily often. Note that if we replace the entire sequence with just one face-angle of value 1, then the sum-condition continues to hold for  $f$ .

Thus, in the orthogonal representation, replace cycle  $C_z$  by a single vertex, adjacent to the four cross-edges. Set the four face-angles at this vertex to be 1. By the above discussion, these face-angles satisfy the sum-condition for the faces, and clearly the four face-angles at the vertex sum to 4, so we have an orthogonal representation.

- Now convert this orthogonal representation into a planar orthogonal drawing  $\Gamma$  that respects the face-angles (and the bend-numbers, which are 0 everywhere). For every original vertex  $u$  of  $G$ , the corresponding cycle  $C_u$  is mapped into a rectangle. Every dummy-vertex is mapped into a point with the four cross-edges on the four sides; remove all these points and re-combine the cross edges to obtain two crossing edges. Since all bend-numbers are 0, all edges of  $G$  are drawn as

horizontal or vertical straight-line segments, and we have obtained the desired RVR of  $G$ .

To summarize, we used 1P-RVDRAWER (with it the techniques of scaling down graphs and apply zig-zag-slide) only to argue that a suitable RVR exists. The actual algorithm to find such an RVR only uses coordinates for 4-connected components; all other steps are done with orthogonal representations. Then merging and unplanarization can be done in constant time per operation and linear time overall. All other parts of the algorithm (such as finding separating triangles, computing 4-connected components, finding a transversal pair of bipolar orientations, and finding the rectangular dual representation) take linear time as well.

The conversion of the final orthogonal representation into the orthogonal drawing  $\Gamma$  also takes linear time, and gives linear integral coordinates. Here “linear” measures the size of the  $\Gamma$ , which in our case is proportional to the number of vertices, edges, and crossings. For an  $n$ -vertex 1-plane graph  $G$  this is  $O(n)$  since  $G$  has at most  $4n - 8$  edges and at most  $n - 2$  crossings [13]. Therefore the final RVR has linear coordinates and the area is  $O(n^2)$ . We conclude:

**Lemma 3.21** *Let  $G$  be a 1-plane graph  $G$  with  $n$  vertices and with no B-configuration, no W-configuration, and no T-configuration as a subgraph. Then there exists an  $O(n)$ -time algorithm that computes a 1-plane RVR  $\Gamma$  of  $G$  that has  $O(n^2)$  area.*

### 3.4 Testing Algorithm

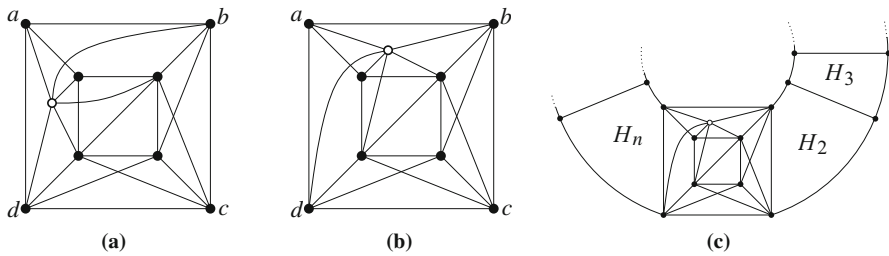
We now show how to test whether a given 1-plane graph  $G$  with  $n$  vertices contains any B-configuration, any W-configuration, or any T-configuration. If  $G$  contains none of them, we can apply algorithm 1P-RVDRAWER to produce the desired representation. Hence, Lemma 3.21 and Lemma 3.22 imply Theorem 3.2.

**Lemma 3.22** *Let  $G$  be a 1-plane graph with  $n$  vertices. There exists an  $O(n)$ -time algorithm to test whether  $G$  contains any B-configuration, any W-configuration, and any T-configuration.*

*Proof* Hong et al. [24] gave an algorithm to detect every possible B-configuration and W-configuration in  $O(n)$  time. Thus, in what follows we describe how to detect a T-configuration, assuming that  $G$  contains no B-configuration and no W-configuration.

We claim that it suffices to test whether  $G^+$  contains a planarized T-configuration. Clearly any T-configuration of  $G$  becomes a planarized T-configuration in  $G^+$ . On the other hand, if  $G^+$  has a planarized T-configuration, then by Observation 3.16 graph  $G_t$  had a planarized T-configuration, which by Observation 3.6 means that  $G_p$  had a planarized T-configuration or B-configuration or W-configuration, which means (by our assumption) that  $G$  has a T-configuration.

Suppose  $G^+$  contains a T-configuration  $t$  with outer vertices  $a, b, c$  and inner vertices  $d, e, f, g, h, i$ , as in Fig. 4(d). Since we added a kite around each dummy-vertex, vertices  $a, b, c$  induce a triangle in  $G^+$ . Thus to detect a T-configuration in  $G^+$ , we first list all triangles of  $G^+$ ; this can be done in linear time since  $G^+$  is planar and simple [11]. For each listed triangle  $\Delta(u, v, w)$ , consider the faces of  $G^+$  that are



**Fig. 14** (a, b) show two different 1-planar embeddings of the same 4-connected 1-planar graph  $H$ . Attaching  $n$  copies of  $H$  as in (c) we obtain a 4-connected 1-planar graph with  $\Theta(n)$  vertices and  $\Omega(2^n)$  different 1-planar embeddings

incident to an edge of  $\Delta(u, v, w)$  and lie in the interior of  $\Delta(u, v, w)$ . Vertices  $u, v$ , and  $w$  are the three outer vertices of a T-configuration if and only if the above faces are three distinct faces and all have a dummy-vertex as third vertex. This condition can be tested in  $O(1)$  time per triangle.  $\square$

## 4 Variable Embedding Setting

Motivated by our characterization, in this section we investigate the variable embedding setting. We remark that, differently from planar graphs, a 1-planar graph  $G$  can have exponentially many different embeddings, even if 3-connected or 4-connected (see e.g. Fig. 14). Thus, an exhaustive analysis may not be feasible. Nevertheless, one may wonder whether it is always possible to compute a 1-planar embedding of  $G$  that has no forbidden configurations. In fact, Alam et al. [1] proved that if  $G$  is 3-connected, then it always admits a 1-planar embedding with no B-configurations and no W-configurations but at most one. We show now that this is not true for T-configurations. Moreover, it might be worth mentioning that there are 2-connected 1-planar graphs such that any 1-planar embedding contains linearly many W-configurations (see, e.g., [8, 16]).

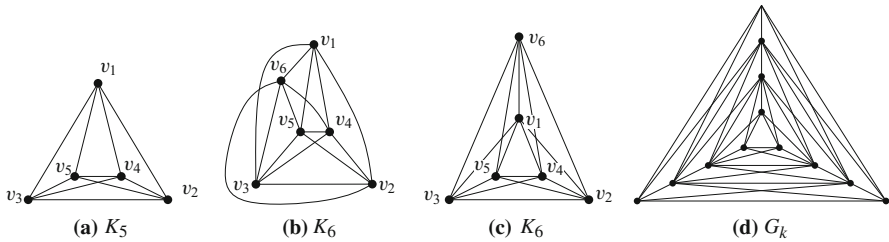
**Theorem 4.1** *Let  $k \geq 1$  be an integer. There exists a 3-connected 1-planar graph  $G_k$  with  $n = 6k$  vertices that contains at least  $k - 1$  T-configurations in every 1-planar embedding.*

To prove Theorem 4.1, the idea is to first show that  $K_6$  has only two embeddings (up to renaming), which differ for choice of the outer face and such that both of them contain a forbidden subgraph.

**Lemma 4.2** *The complete graph  $K_6$  has only two 1-planar embeddings, up to renaming of the vertices. In each of these two embeddings, the outer face forms a B-configuration or contains the outer vertex of a T-configuration.*

*Proof* The complete graph  $K_5$  has a unique 1-planar embedding up to renaming and outer face [31, 32], which is shown in Fig. 15(a). Observe that every face of such an embedding is a triangle, either incident to two vertices and a crossing point or to three





**Fig. 15** (a)  $K_5$  in its unique (up to renaming and outer face) 1-planar embedding. (b, c)  $K_6$  in its unique (up to renaming) 1-planar embedding and two possible choices of the outer face. (d) Multiple copies of  $K_6$  combined

vertices. In order to realize a 1-planar embedding of  $K_6$ , a sixth vertex can only be added to one of the four faces of  $K_5$  that are incident to three vertices. Two of these four cases are illustrated in Fig. 15 (b), (c); the other two cases are symmetric and yield the same embedding (up to renaming of the vertices). In every one of these two cases the statement holds.  $\square$

The next corollary is an immediate consequence of Corollary 3.4 and Lemma 4.2.

**Corollary 4.3**  $K_6$  has no 1-planar embedding that can be represented as a 1-plane RVR.

For any  $k > 0$  we can create a 3-connected 1-planar graph  $G_k$  with  $6k$  vertices that consists of  $k$  vertex-disjoint copies of  $K_6$ . Figure 15(d) shows one possible such construction for  $k = 2$ . As shown by Grigni et al. [10], since  $G_k$  is a 3-connected triangulated 1-planar graph, the edges of any separating triangle of  $G_k$  are crossing-free in any 1-planar embedding of the graph. Hence, all copies of  $K_6$  in  $G_k$ , except for at most one (which is part of the outer face), must be embedded as in Fig. 15(c). Thus,  $G_k$  contains at least  $k - 1$  T-configurations. This concludes the proof of the Theorem 4.1.

Moreover, if some spanning subgraph  $H$  of  $G$  had a 1-plane RVR  $\Gamma$ , then for each of the  $k$  copies of  $K_6$ , at least one edge cannot exist in  $\Gamma$ , else  $\Gamma$  would induce an RVR of  $K_6$ . Hence at least  $k = n/6$  edges of  $G$  must be removed from  $H$  in order to be able to compute  $\Gamma$ . This proves the following stronger corollary.

**Corollary 4.4** Let  $k \geq 1$  be an integer. There exists a 3-connected 1-planar graph  $G_k$  with  $n = 6k$  vertices that does not admit any 1-planar embedding that can be realized as a 1-plane RVR, unless we remove at least  $k = n/6$  edges.

On the positive side, a 4-connected 1-planar graph  $G$  admits a 1-planar embedding  $G'$  with no B-configurations and W-configurations except that the outer face may consist of a crossing with an edge between two of its endpoints [1]. In any T-configuration in a 1-plane graph, the three outer vertices are either the outer face or form a separating triple, so any embedding of a 4-connected 1-planar graph has at most one T-configuration. This means that in  $G'$  there is at most one T-configuration, and if there is one then there is no B-configuration or W-configuration. This leads to the following theorem.

**Theorem 4.5** *Every 4-connected 1-planar graph  $G$  admits a 1-planar embedding that can be realized as a 1-plane RVR, after removing at most one edge.*

An *optimal 1-planar graph* is one that has the maximum number  $4n - 8$  of possible edges. Optimal 1-planar graphs are always 4-connected, and in any 1-planar embedding all crossings are in kites, with the exception of one crossing on the outer face [40]. It follows that  $G$  does not contain any T-configuration, and with the exception of the crossing at the outer face, it contains no B-configuration or W-configuration.

**Corollary 4.6** *Every optimal 1-plane graph  $G$  admits a 1-plane RVR, after removing one edge.*

## 5 Conclusions and Open Problems

We studied embedding-preserving rectangle visibility representations of embedded nonplanar graphs. We showed that we can test in polynomial time whether an embedded graph has such a representation. Of special interest to us were 1-plane graphs, for which we described a linear-time algorithm to test the existence of embedding-preserving rectangle visibility representations. Moreover, in case of a negative answer the algorithm provides a witness in form of either a B-configuration, W-configuration, or T-configuration. We also studied the setting in which the graph is not embedded, and we aim to find a 1-planar embedding that can be realized as a 1-plane RVR. We showed that not all 3-connected 1-planar graphs admit such an embedding, but all 4-connected 1-planar graphs do after deleting (at most) one edge.

The most pressing open problem is whether we can restrict the drawings less and still test for the existence of rectangle visibility representations? Most importantly, if we fix the rotation system and outer face, but *not* the cyclic order of the edges around the crossings (and perhaps not even which edges cross), is it possible to test whether an RVR respecting the rotation system and outer face exists? The NP-hardness proof of Shermer [38] does not hold if the rotation system is fixed, since it uses a reduction from linear-arboricity-2, and fixing the rotation system would severely restrict the possible ways of splitting a graph into two linear forests. The orthogonal representation approach utterly fails if the order of crossings is not fixed, since it requires planarization as a first step. Also, it is NP-complete to decide if a graph with a fixed rotation system admits a 1-planar embedding in which the rotation system is preserved [2].

Finally, it is immediate to observe that any RVR can be transformed into a drawing such that vertices are points, edges are polylines with at most 2 bends, and crossings occur only at right angles. Hence, Lemma 3.21 can be used to compute *RAC-drawings* (see e.g. [17]) with at most 2 bends per edge in linear time and quadratic area. It is known that not all 1-plane (straight-line drawable) graphs admit a bendless RAC-drawing [19] (and when they do they may require exponential area [9]), and that every 1-planar graph admits a 1-planar RAC-drawing with at most 1 bend per edge [3] (the drawing algorithm may produce drawings that take exponential area), we ask whether all 1-plane graphs admit an embedding-preserving RAC-drawing with at most 2 bends per edge in polynomial area?

**Acknowledgements** We thank the anonymous referees of this paper for their valuable suggestions.

## References

1. Alam, M.J., Brandenburg, F.J., Kobourov, S.G.: Straight-line grid drawings of 3-connected 1-planar graphs. In: Graph Drawing. Lecture Notes in Computer Science, vol. 8242, pp. 83–94. Springer, Cham (2013)
2. Auer, C., Brandenburg, F.J., Gleißner, A., Reislhuber, J.: 1-Planarity of graphs with a rotation system. *J. Graph Algorithms Appl.* **19**(1), 67–86 (2015)
3. Bekos, M.A., Didimo, W., Liotta, G., Mehrabi, S., Montecchiani, F.: On RAC drawings of 1-planar graphs. *Theor. Comput. Sci.* **689**, 48–57 (2017)
4. Biedl, T.: Relating bends and size in orthogonal graph drawings. *Inform. Process. Lett.* **65**(2), 111–115 (1998)
5. Biedl, T., Lubiw, A., Petrick, M., Spriggs, M.: Morphing orthogonal planar graph drawings. *ACM Trans. Algorithms* **9**(4), Art. No. 29 (2013)
6. Bodendiek, R., Schumacher, H., Wagner, K.: Bemerkungen zu einem Sechsfarbenproblem von G. Ringel. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* **53**(1), 41–52 (1983)
7. Brandenburg, F.J.: 1-Visibility representations of 1-planar graphs. *J. Graph Algorithms Appl.* **18**(3), 421–438 (2014)
8. 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.) Graph Drawing. Lecture Notes in Computer Science, vol. 7704, pp. 327–338. Springer, Cham (2013)
9. Brandenburg, F.J., Didimo, W., Evans, W.S., Kindermann, P., Liotta, G., Montecchiani, F.: Recognizing and drawing IC-planar graphs. *Theor. Comput. Sci.* **636**, 1–16 (2016)
10. Chen, Z.Z., Grigni, M., Papadimitriou, C.H.: Recognizing hole-free 4-map graphs in cubic time. *Algorithmica* **45**(2), 227–262 (2006)
11. Chiba, N., Nishizeki, T.: Arboricity and subgraph listing algorithms. *SIAM J. Comput.* **14**(1), 210–223 (1985)
12. Cohen, M.B., Mądry, A., Sankowski, P., Vladu, A.: Negative-weight shortest paths and unit capacity minimum cost flow in  $\tilde{O}(m^{10/7} \log W)$  time (extended abstract). In: Klein, P.N. (ed.) Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17), pp. 752–771. SIAM, Philadelphia (2017)
13. Czap, J., Hudák, D.: On drawings and decompositions of 1-planar graphs. *Electron. J. Comb.* **20**(2), Art. No. 54 (2013)
14. Dean, A.M., Evans, W., Gethner, E., Laison, J.D., Safari, M.A., Trotter, W.T.: Bar  $k$ -visibility graphs. *J. Graph Algorithms Appl.* **11**(1), 45–59 (2007)
15. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice-Hall, Upper Saddle River (1999)
16. Di Giacomo, E., Didimo, W., Evans, W.S., Liotta, G., Meijer, H., Montecchiani, F., Wismath, S.K.: Ortho-polygon visibility representations of embedded graphs. *Algorithmica* (2017). doi:[10.1007/s00453-017-0324-2](https://doi.org/10.1007/s00453-017-0324-2)
17. Didimo, W., Liotta, G.: The crossing-angle resolution in graph drawing. In: Pach, J. (ed.) Thirty Essays on Geometric Graph Theory, pp. 167–184. Springer, New York (2013)
18. Duchet, P., Hamidoune, Y., Vergnas, M.L., Meyniel, H.: Representing a planar graph by vertical lines joining different levels. *Discrete Math.* **46**(3), 319–321 (1983)
19. Eades, P., Liotta, G.: Right angle crossing graphs and 1-planarity. *Discrete Appl. Math.* **161**(7–8), 961–969 (2013)
20. Evans, W., Kaufmann, M., Lenhart, W., Mchedlidze, T., Wismath, S.: Bar 1-visibility graphs and their relation to other nearly planar graphs. *J. Graph Algorithms Appl.* **18**(5), 721–739 (2014)
21. Felsner, S.: Rectangle and square representations of planar graphs. In: Pach, J. (ed.) Thirty Essays on Geometric Graph Theory, pp. 213–248. Springer, New York (2013)
22. Fusy, É.: Transversal structures on triangulations: a combinatorial study and straight-line drawings. *Discrete Math.* **309**(7), 1870–1894 (2009)
23. He, X.: On finding the rectangular duals of planar triangular graphs. *SIAM J. Comput.* **22**(6), 1218–1226 (1993)

24. 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.) *Computing and Combinatorics*. Lecture Notes in Computer Science, vol. 7434, pp. 335–346. Springer, Berlin (2012)
25. Hutchinson, J.P., Shermer, T.C., Vince, A.: On representations of some thickness-two graphs. *Comput. Geom.* **13**(3), 161–171 (1999)
26. Kant, G.: A more compact visibility representation. *Int. J. Comput. Geom. Appl.* **7**(3), 197–210 (1997)
27. Kant, G., Bodlaender, H.L.: Triangulating planar graphs while minimizing the maximum degree. *Inf. Comput.* **135**(1), 1–14 (1997)
28. Kant, G., He, X.: Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems. *Theor. Comput. Sci.* **172**(1–2), 175–193 (1997)
29. Kant, G., Liotta, G., Tamassia, R., Tollis, I.G.: Area requirement of visibility representations of trees. *Inf. Process. Lett.* **62**(2), 81–88 (1997)
30. Kobourov, S.G., Liotta, G., Montecchiani, F.: An annotated bibliography on 1-planarity. *Comput. Sci. Rev.* doi:[10.1016/j.cosrev.2017.06.002](https://doi.org/10.1016/j.cosrev.2017.06.002)
31. Korzhik, V.P.: Minimal non-1-planar graphs. *Discrete Math.* **308**(7), 1319–1327 (2008)
32. Kynčl, J.: Enumeration of simple complete topological graphs. *Eur. J. Comb.* **30**(7), 1676–1685 (2009)
33. Liotta, G., Montecchiani, F.: L-visibility drawings of IC-planar graphs. *Inf. Process. Lett.* **116**(3), 217–222 (2016)
34. Madry, A.: Computing maximum flow with augmenting electrical flows. In: Dinur, I. (ed.) *IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS'16)*, pp. 593–602. IEEE (2016)
35. Otten, R.H.J.M., Wijk, J.G.V.: Graph representations in interactive layout design. In: *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCSS'78)*, pp. 914–918. IEEE (1978)
36. Pach, J., Tóth, G.: Graphs drawn with few crossings per edge. *Combinatorica* **17**(3), 427–439 (1997)
37. Rosenstiehl, P., Tarjan, R.E.: Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete Comput. Geom.* **1**(4), 343–353 (1986)
38. Shermer, T.C.: On rectangle visibility graphs. III. External visibility and complexity. In: *Proceedings of the 8th Canadian Conference on Computational Geometry (CCCG'96)*, pp. 234–239. Carleton University Press, Ottawa (1996)
39. Streinu, I., Whitesides, S.: Rectangle visibility graphs: Characterization, construction, and compaction. In: Alt, H., Habib, M. (eds.) *STACS 2003*. Lecture Notes in Computer Science, vol. 2607, pp. 26–37. Springer, Berlin (2003)
40. Suzuki, Y.: Re-embeddings of maximum 1-planar graphs. *SIAM J. Discrete Math.* **24**(4), 1527–1540 (2010)
41. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.* **16**(3), 421–444 (1987)
42. Tamassia, R., Tollis, I.G.: A unified approach to visibility representations of planar graphs. *Discrete Comput. Geom.* **1**(4), 321–341 (1986)
43. Thomassen, C.: Plane representations of graphs. In: Bondy, J.A., Murty, U.S.R. (eds.) *Progress in Graph Theory*, pp. 43–69. Academic Press, New York (1984)
44. Thomassen, C.: Rectilinear drawings of graphs. *J. Graph Theory* **12**(3), 335–341 (1988)
45. Ullman, J.D.: *Computational Aspects of VLSI*. Computer Science Press, Rockville (1984)
46. Ungar, P.: On diagrams representing maps. *J. Lond. Math. Soc.* **s1-28**(3), 336–342 (1953)
47. Wismath, S.K.: Characterizing bar line-of-sight graphs. In: O'Rourke, J. (ed.) *Proceedings of the 1st Annual Symposium on Computational Geometry (SoCG'85)*, pp. 147–152. ACM, New York (1985)