

Ortho-Polygon Visibility Representations of Embedded Graphs

Emilio Di Giacomo¹, Walter Didimo¹, William S. Evans², Giuseppe Liotta¹,
Henk Meijer³, Fabrizio Montecchiani^{1(✉)}, and Stephen K. Wismath⁴

¹ Università Degli Studi di Perugia, Perugia, Italy
`fabrizio.montecchiani@unipg.it`

² University of British Columbia, Vancouver, Canada

³ University College Roosevelt, Middelburg, The Netherlands

⁴ University of Lethbridge, Lethbridge, Canada

Abstract. An ortho-polygon visibility representation of an n -vertex embedded graph G (OPVR of G) is an embedding preserving drawing of G that maps every vertex to a distinct orthogonal polygon and each edge to a vertical or horizontal visibility between its end-vertices. The vertex complexity of an OPVR of G is the minimum k such that every polygon has at most k reflex corners. We present polynomial time algorithms that test whether G has an OPVR and, if so, compute one of minimum vertex complexity. We argue that the existence and the vertex complexity of an OPVR of G are related to its number of crossings per edge and to its connectivity. Namely, we prove that if G is 1-plane (i.e., it has at most one crossing per edge) an OPVR of G always exists while this may not be the case if two crossings per edge are allowed. Also, if G is a 3-connected 1-plane graph, we can compute in $O(n)$ time an OPVR of G whose vertex complexity is bounded by a constant. However, if G is a 2-connected 1-plane graph, the vertex complexity of any OPVR of G may be $\Omega(n)$. In contrast, we describe a family of 2-connected 1-plane graphs for which an embedding that guarantees constant vertex complexity can be computed. Finally, we present the results of an experimental study on the vertex complexity of OPVRs of 1-plane graphs.

1 Introduction

Visibility representations are among the oldest and most studied methods to display graphs. The first papers appeared between the late 70s and the mid 80s, mostly motivated by VLSI applications (see, e.g., [15, 24, 25, 31, 32, 34]). These papers were devoted to *bar visibility representations (BVR)* of planar graphs where the vertices are modeled as non-overlapping horizontal segments, called *bars*, and the edges correspond to vertical visibilities, i.e. vertical segments that do not intersect any bar other than at their end points. The study of visibility

Research of EDG, WD, GL and FM supported in part by the MIUR project AMANDA, prot. 2012C4E3KT_001. NSERC funding is gratefully acknowledged for WE and SW.

representations of non-planar graphs started about ten years later when *rectangle visibility representations (RVR)* were introduced in the computational geometry and graph drawing communities (see e.g., [11, 20, 21, 27]). Every vertex is represented as an axis-aligned rectangle and two vertices are connected by an edge using either horizontal or vertical visibilities. Figure 1(a) is an example of a RVR of the complete graph K_5 . RVRs are an attractive way to draw a non-planar graph: Edges are easy to follow because they do not bend and can have only one of two possible slopes, edge crossings are perpendicular, textual labels associated with the vertices can be inserted in the rectangles. Motivated by the NP-hardness of recognizing whether a graph admits an RVR [27], Streinu and Whitesides [28] initiated the study of RVRs that must respect a set of topological constraints. They proved that if a graph G is given together with the cyclic order of the edges around each vertex, the outer face, and a horizontal/vertical direction for each edge, then there exists a polynomial-time algorithm to test whether G admits an RVR that respects these constraints. Biedl *et al.* [5] have shown that testing the representability of G is polynomial also with a different set of constraints, namely when G is given with an embedding that must be preserved in the RVR. In these settings, however, even structurally simple “almost planar” graphs may not admit an RVR. For example, the embedded graph of Fig. 1(b) is 1-plane (i.e., it has at most one crossing per edge), and it does not have an embedding-preserving RVR [5].

In this paper we introduce a generalization of RVRs. We study to what extent such a generalization enlarges the family of graphs that are representable, and we describe testing and drawing algorithms. Let G be an embedded graph. An *ortho-polygon visibility representation (OPVR)* of G is an embedding-preserving drawing of G that maps each vertex to a distinct orthogonal polygon and each edge to a vertical or horizontal visibility between its end-vertices. For example, Fig. 1(c) is an embedding-preserving OPVR of the graph of Fig. 1(b). In Fig. 1(c) all vertices except two are rectangles: The non-rectangular vertices have a reflex corner each; intuitively, each of them is “away from a rectangle” by one reflex corner. We say that the OPVR of Fig. 1(c) has *vertex complexity* one. More generally, we say that an OPVR has vertex complexity k , if k is the minimum

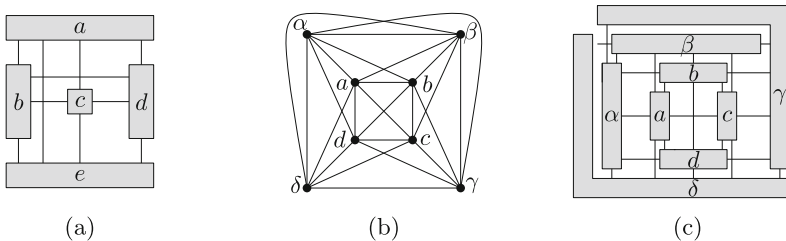


Fig. 1. (a) An RVR of K_5 . (b) An embedded graph G that does not admit an embedding preserving RVR. (c) An embedding preserving OPVR of G with vertex complexity one.

integer such that any polygon representing a vertex has at most k reflex corners. We are not only interested in characterizing and testing what graphs admit an OPVR, but we also aim at computing representations of minimum vertex complexity (RVRs if possible). The main results in this paper are as follows.

- In Sect. 3 we present a combinatorial characterization of the graphs that admit an OPVR. This leads to an $O(n^2)$ -time algorithm that tests whether an embedded graph G with n vertices admits an embedding-preserving OPVR. If so, an embedding-preserving OPVR of G with minimum vertex complexity is computed in $O(n^{\frac{5}{2}} \log^{\frac{3}{2}} n)$ time. An implication of this characterization is that any 1-plane graph admits an embedding-preserving OPVR. We remark that 1-planar graphs have been widely studied in recent years (see, e.g., [2, 4–6, 8, 16–18, 22, 29, 33]).
- In Sect. 4 we prove that every 3-connected 1-plane graph admits an OPVR whose vertex complexity is bounded by a constant and that this representation can be computed in $O(n)$ time. This implies an $O(n^{\frac{7}{4}} \sqrt{\log n})$ -time algorithm to compute OPVRs of minimum vertex complexity for these graphs. Biedl *et al.* [5] proved that not every 3-connected 1-plane graph has a representation with zero vertex complexity, and we show a lower bound of two for infinitely many graphs of this family.
- In Sect. 4 we also study 2-connected 1-plane graphs. Not every 2-connected 1-plane graph can be augmented to become 3-connected (and 1-plane). This has a strong impact on the vertex complexity of the corresponding OPVRs. We prove that an embedding-preserving OPVR of a 2-connected 1-plane graph may require $\Omega(n)$ vertex complexity. Also, we show a sufficient condition that allows to compute an embedding that guarantees constant vertex complexity in $O(n)$ time.
- In Sect. 5 we discuss the results of an experimental study whose aim is to estimate both the vertex complexity of these drawings in practice and the percentage of vertices that are not represented as rectangles.

Some proofs and technicalities are omitted and can be found in [14].

2 Preliminaries

We assume familiarity with basic terminology of graph drawing [13]. We only consider *simple* drawings of graphs, i.e., drawings where two edges have at most one point in common (either a common endpoint or a common interior point where the two edges properly cross each other). A graph is *planar* if it admits a crossing free drawing. Such a drawing subdivides the plane into topologically connected regions, called *faces*. The infinite region is the *outer face*. A *planar embedding* of a graph is an equivalence class of planar drawings that define the same set of faces. A *plane graph* is a planar graph with a given planar embedding. Let f be a face of a plane graph G . The number of vertices encountered in the closed walk along the boundary of f is the *degree* of f , denoted as $\deg(f)$. If G is not 2-connected a vertex may be encountered more than once, thus contributing

more than one unit to the degree of the face. The concept of planar embedding can be extended to non-planar drawings. Given a non-planar drawing, replace each crossing with a dummy vertex. The resulting planarized drawing has a planar embedding. An *embedding* of a graph G is an equivalence class of drawings of G whose planarized versions have the same planar embedding. An *embedded graph* G is a graph with a given embedding: An *embedding-preserving* drawing Γ of G is a drawing of G whose embedding coincides with that of G .

A bar visibility representation (BVR) is *strong* if each visibility between two bars corresponds to an edge of the graph, while it is *weak* when visibilities between non adjacent bars may occur. An *orthogonal polygon* is a polygon whose sides are axis-aligned. A *corner* of an orthogonal polygon is a point of the polygon where a horizontal and a vertical side meet. A corner is a *reflex corner* if it forms a $\frac{3\pi}{2}$ angle inside the polygon. An *ortho-polygon visibility representation* (OPVR) of a graph G maps each vertex v of G to a distinct orthogonal polygon $P(v)$ and each edge (u, v) of G to a vertical or horizontal visibility connecting $P(u)$ and $P(v)$ and not intersecting any other polygon $P(w)$, for $w \notin \{u, v\}$. The intersection points between visibilities and polygons are the *attachment points*. We adopt the ϵ -visibility model [21, 28, 31, 34], where the segments representing the edges can be replaced by strips of non-zero width; this implies that an attaching point never coincides with a corner of a polygon. An OPVR is on an *integer grid* if all its corners and attachment points have integer coordinates. Given an OPVR, we can extract a drawing from it as follows. For each vertex v , place a point inside polygon $P(v)$ and connect it to all the attachment points of the boundary of $P(v)$; this can be done without creating any crossing and preserving the circular order of the edges around the vertices. Thus, we refer to an OPVR as a drawing and we extend to OPVRs all the definitions given for drawings. An OPVR γ of an embedded graph is *embedding preserving* if the drawing extracted from γ is embedding preserving. The *vertex complexity* of an OPVR is the maximum number of reflex corners in any polygon representing a vertex. An *optimal OPVR* is an OPVR with minimum vertex complexity.

3 Test and Optimization for Embedded Graphs

Any embedded graph G that admits an OPVR is biplanar, i.e., its edge set can be bicolored so that each color class induces a plane subgraph (use red for the horizontal and blue for the vertical edges of an OPVR of G). However, a biplanar graph G may not have an embedding preserving OPVR. An example is given in Fig. 2 (thin and bold edges define the two colors). The boundary of face f in the figure contains six edge crossings and no vertex. In any OPVR, each crossing forms a $\frac{\pi}{2}$ angle inside f , thus the orthogonal polygon representing f would have six $\frac{\pi}{2}$ corners and no $\frac{3\pi}{2}$ corners in its interior, which is impossible.

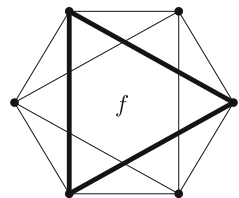


Fig. 2. An embedded graph with no embedding-preserving OPVR.

In the following we first describe an algorithm that, given an embedded graph G that admits an embedding preserving OPVR, computes an optimal OPVR of G (Lemma 2). Then, we describe a characterization of the embedded graphs that admit an embedding preserving OPVR (Lemma 3). This leads to an efficient testing algorithm and it implies that 1-plane graphs always admit an embedding preserving OPVR. Both results extend the *topology-shape-metrics (TSM)* framework to handle OPVRs. The TSM approach, briefly recalled below, was introduced by Tamassia [30] to compute *orthogonal drawings* (see also [13, 19]).

The TSM Framework. In an orthogonal drawing of a degree-4 graph each edge is a polyline of horizontal and vertical segments. An angle formed by two consecutive segments incident to the same vertex is a *vertex-angle*; an angle at a bend is a *bend-angle*. The following basic property holds.

Property 1. Let f be a face of an orthogonal drawing and let $N_\alpha(f)$ be the number of vertex-angles of value α inside f , with $\alpha \in \{\frac{\pi}{2}, \frac{3\pi}{2}, 2\pi\}$. Then: $N_{\frac{\pi}{2}}(f) - N_{\frac{3\pi}{2}}(f) - 2N_{2\pi}(f) = 4$ if f is an internal face and $N_{\frac{\pi}{2}}(f) - N_{\frac{3\pi}{2}}(f) - 2N_{2\pi}(f) = -4$ if f is the outer face.

Given a degree-4 graph G , the TSM computes, in three steps, an orthogonal drawing Γ of G with minimum number of bends (see also [13]). The first step, *planarization*, computes an embedding of G and replaces crossing points with dummy vertices. The resulting plane graph G' has $n + c$ vertices, where n and c are the number of vertices and crossings of G , respectively. The second step, *orthogonalization*, computes an *orthogonal representation* H of G' , which specifies the values of all vertex-angles and the sequence of bends along each edge. H is computed by means of a flow network N , where each unit of flow corresponds to a $\frac{\pi}{2}$ angle. Each *vertex-node* in N corresponds to a vertex of G' and supplies 4 units of flow; each *face-node* in N corresponds to a face of G' and demands an amount of flow proportional to its degree. Bends along edges correspond to units of flow transferred across adjacent faces of G' through the corresponding arcs of N , and each bend has a unit cost in N . Network N is constructed in $O(n + c)$ time since it has $O(n + c)$ nodes and arcs. Also, it always admits a feasible flow. A feasible flow Φ of cost b of N defines an orthogonal representation H of G' with b bends, and *vice versa*. The third step, *compaction*, computes in $O(n + c + b)$ time an orthogonal drawing preserving the shape of H on an integer grid of size $O(n + c + b) \times O(n + c + b)$.

Our Approach. To exploit the TSM framework, we define a new plane graph \overline{G} obtained from the input embedded graph G as follows (refer to Figs. 3(a) and (b)). Replace each vertex v with a cycle $C(v)$ of $d = \text{deg}(v)$ vertices, so that each of these vertices is incident to one of the edges formerly incident to v , preserving the circular order of the edges around v . If $d = 1$ or $d = 2$, $C(v)$ is a self-loop or a pair of parallel edges, respectively. $C(v)$ is the *expansion cycle* of v ; the vertices and the edges of $C(v)$ are the *expansion vertices* and the *expansion edges*, respectively. Also, replace crossings with *dummy vertices*. \overline{G} is called the

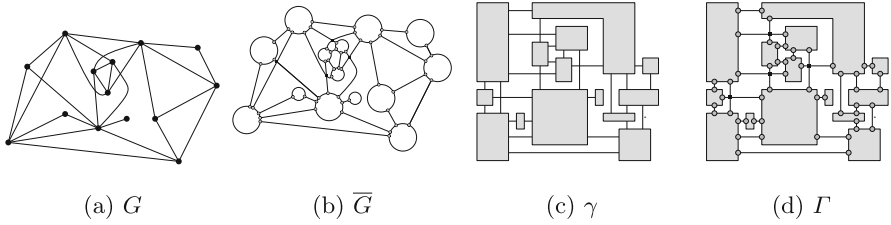


Fig. 3. (a) An embedded graph G and (b) its planarized expansion \overline{G} . (c) An OPVR γ of G and (d) the orthogonal drawing Γ obtained from γ .

planarized expansion of G . The edges of \overline{G} that are not expansion edges are the original edges. Each expansion vertex has degree 3 and each dummy vertex has degree 4. The next lemma and properties follow (see also Figs. 3(c) and (d)).

Lemma 1. *An embedded graph G admits an embedding preserving OPVR if and only if \overline{G} admits an orthogonal representation with the following properties: **P1.** Each vertex-angle inside an expansion cycle has value π . **P2.** Each original edge has no bend.*

Property 2. If G is biplanar, for each face f of \overline{G} that is not an expansion cycle, $\deg(f) \geq 4$.

Property 3. If G admits an embedding preserving OPVR, then for every internal face f of \overline{G} consisting only of dummy vertices, $\deg(f) = 4$.

Lemma 2. *Let G be an n -vertex embedded graph that admits an embedding preserving OPVR. There exists an $O(n^{\frac{5}{2}} \log^{\frac{3}{2}} n)$ -time algorithm that computes an embedding preserving optimal OPVR γ of G . Also, γ has the minimum number of total reflex corners among all embedding preserving optimal OPVRs of G .*

Proof. Since G admits an embedding preserving OPVR, it is biplanar. Hence it has $m \leq 6n - 12$ edges. By Lemma 1, an OPVR of G can be found by computing an orthogonal representation that satisfies **P1** and **P2**. This can be done by computing a feasible flow in the Tamassia’s flow network N associated with \overline{G} , subject to these constraints: (i) Every arc of N from a vertex-node to a face-node has fixed flow 2 if the face-node corresponds to an expansion cycle (which implies a π angle inside the cycle), and fixed flow 1 otherwise (which implies a $\frac{\pi}{2}$ angle inside the face); (ii) Arcs from two face-nodes such that none of them corresponds to an expansion cycle of \overline{G} are removed (to avoid bends on the original edges). A feasible flow for N may not correspond to an optimal OPVR. To minimize the vertex complexity we construct a different flow network as follows. The amount of flow moved from a vertex-node to an adjacent face-node is fixed a priori, and thus we can construct from N an equivalent flow network N' , such that all vertex-nodes are removed and their supplies are transferred onto the supply of the adjacent face-nodes. Namely, each face-node v_f corresponding

to an expansion cycle f receives $2 \deg(f)$ units of flow, while its demand is $2 \deg(f) - 4$ by definition. This is equivalent to saying that v_f will supply 4 units of flow in N' . Similarly, each face-node v_f corresponding to a face f that is not an expansion cycle receives $\deg(f)$ units of flow, while its demand is $2 \deg(f) - 4$ (or $2 \deg(f) + 4$ if f is the outer face). This is equivalent to saying that v_f will demand flow $\deg(f) - 4$ ($\deg(f) + 4$ if f is the outer face) in N' . By Property 2, $\deg(f) \geq 4$ and therefore $\deg(f) - 4 \geq 0$. We now consider every face f of \overline{G} having dummy vertices only (if any), and the corresponding face-node v_f in N' . Note that v_f is an isolated node of N' . Since G admits an embedding preserving OPVR, by Property 3, $\deg(f) = 4$; hence, we can remove v_f from N' and conclude that f must be drawn as a rectangle. Thus, every face-node in N' corresponds to a face of \overline{G} with at least one expansion vertex on its boundary. Since every expansion vertex belongs to at most three faces of \overline{G} and there are $O(n)$ expansion vertices, then N' has $O(n)$ nodes and arcs. We also add gadgets to the network N' in order to impose an upper bound h on the number of reflex corners inside the polygons representing the expansion cycles. Namely, let v_f be a node of N' corresponding to an expansion cycle f . We replace v_f with two face-nodes: a node v_f^{in} , with zero supply and demand; and a node v_f^{out} , with the same supply as v_f (which is 4). The incoming edges of v_f become incoming edges of v_f^{in} , while the outgoing edges of v_f become outgoing edges of v_f^{out} . Finally, we add an edge (v_f^{in}, v_f^{out}) with capacity h . Let N'' be the flow network resulting by applying this transformation to all nodes of N' corresponding to expansion cycles. Since each unit of flow entering in v_f (now in v_f^{in}) corresponds to a $\frac{3\pi}{2}$ angle inside f , a feasible flow of N'' defines an orthogonal representation where each expansion cycle is a polygon with at most h reflex corners, i.e., such a feasible flow defines an OPVR having vertex complexity at most h . N'' is computed in $O(n)$ time and has $O(n)$ nodes and arcs, as N' . In order to guarantee that the OPVR has the minimum number of reflex corners among those with vertex complexity at most h , we compute a feasible flow of minimum cost. Namely, we apply the min-cost flow algorithm of Garg and Tamassia [19], whose complexity is $O(\chi^{\frac{3}{4}} m'' \sqrt{\log n''})$, where n'' and m'' are the number of nodes and arcs of N'' , respectively, and χ is the cost of the flow¹. As already observed, both n'' and m'' are $O(n)$. Also, since the value of the flow is $O(n)$ and in a min-cost flow each unit of flow moved along an augmenting path can traverse each face-node at most once, we have $\chi = O(n^2)$. Hence, a min-cost flow of N'' (if any) is computed in $O(n^{\frac{5}{2}} \sqrt{\log n})$ time.

The supplied flow in N'' is $4n$ (four units for each expansion cycle) and each unit of a min-cost flow can traverse a face-node at most once. Thus, the vertex complexity of an embedding preserving optimal OPVR of G is $k \leq 4n$. We can find the value of k by performing a binary search in the range $[0, 4n]$, testing, for each considered value h , if an OPVR with vertex complexity at most h exists. The number of tests is $O(\log n)$ and each test takes $O(n^{\frac{5}{2}} \sqrt{\log n})$ time, with the algorithm described above. Thus, computing an orthogonal representation H

¹ Since N'' may not be planar, we cannot use the faster min-cost flow algorithm in [9].

corresponding to an OPVR with vertex complexity k takes $O(n^{\frac{5}{2}} \log^{\frac{3}{2}} n)$ time. A drawing of H is computed with the compaction step of the TSM. Since H has at most $k \cdot n$ bends, this can be done in $O((k + 1)n + c) = O(n^2)$ time. \square

We now introduce a new plane graph associated with the planarized expansion \overline{G} of G . Namely, let \overline{G}^* be the dual graph of \overline{G} where the dual edges associated with the original edges are removed. \overline{G}^* has a vertex for each face of \overline{G} and an edge between two vertices for every edge of an expansion cycle shared by the two corresponding faces. We call \overline{G}^* the *simplified dual* of \overline{G} . Given a connected component \mathcal{C} of \overline{G}^* , denote by $F_{\mathcal{C}}$ the set of faces of \overline{G} corresponding to the vertices of \mathcal{C} , by $F_{\mathcal{C}}^{ex}$ the subset of $F_{\mathcal{C}}$ corresponding to the expansion cycles, and by $F_{\mathcal{C}}^{n_{ex}}$ the set $F_{\mathcal{C}} \setminus F_{\mathcal{C}}^{ex}$. Finally, let f_{out} be the outer face of \overline{G} . We give the following characterization.

Lemma 3. *An embedded graph G admits an embedding preserving OPVR if and only if for each connected component \mathcal{C} of \overline{G}^* we have $\sum_{f \in F_{\mathcal{C}}^{n_{ex}}} \deg(f) = 4|F_{\mathcal{C}}| - 8 \cdot \beta$, where $\beta = 1$ if $f_{out} \in F_{\mathcal{C}}$ and $\beta = 0$ otherwise.*

Lemma 3 leads to an $O(n+c)$ -time algorithm that tests whether an embedded graph G with n vertices and c crossings admits an embedding preserving OPVR. Indeed, the size of \overline{G}^* is $O(n + c)$ and thus the condition of Lemma 3 can be checked in $O(n + c)$ time. If G is biplanar it has at most $6n - 12$ edges, and $O(n+c) = O(n^2)$. The next theorem summarizes the contribution of this section.

Theorem 1. *Let G be an n -vertex embedded graph. There exists an $O(n^2)$ -time algorithm that tests if G admits an embedding preserving OPVR and, if so, it computes an embedding preserving optimal OPVR γ in $O(n^{\frac{5}{2}} \log^{\frac{3}{2}} n)$ time. Also, γ has the minimum number of reflex corners among all embedding preserving optimal OPVRs of G .*

We remark that an alternative algorithm to test whether G admits an embedding preserving OPVR can be derived from the result in [3]. Namely, Alam et al. [3] showed an algorithm to test whether an n -vertex biconnected plane graph G admits an orthogonal drawing such that edges have no bends, and each face f has most k_f reflex corners. The time complexity of this algorithm is $O((nk)^{\frac{3}{2}})$ -time, where $k = \max_{f \in G} k_f$. Thus, one can compute \overline{G} and split each expansion edge of \overline{G} with $4n$ subdivision vertices (the maximum number of reflex corners that a face can have). The resulting graph \overline{G}' has $O(n^2)$ vertices. Then one can apply the algorithm by Alam et al. on \overline{G}' with $k_f = 4n$ for every face f of G . However, this would lead to a time complexity $O(n^{\frac{9}{2}})$. We conclude this section by observing that the number of crossings per edge is a critical parameter for the ortho-polygon representability of an embedded graph: Even two crossings per edge may give rise to a graph that cannot be represented (see Fig. 2). On the positive side, the following theorem can be proved by applying Lemma 3.

Theorem 2. *Every 1-plane graph admits an embedding-preserving OPVR.*

4 Bounds and Optimization for 1-Plane Graphs

Motivated by Theorem 2, in this section we study upper and lower bounds on the vertex complexity of 1-plane graphs. We present a result about partitioning the edges of a 3-connected 1-plane graph so that each partition set induces a plane graph and one of these plane graphs has maximum vertex degree six, which is a tight bound. This result may be of independent interest since it contributes to recent combinatorial studies about partitioning the edge set of 1-plane graphs into two plane subgraphs having special properties (see e.g. [1, 10, 23]). Next, we use this result to show an upper bound of 12 and a lower bound of 2 on the vertex complexity of 3-connected 1-plane graphs. Finally, we argue that the vertex complexity of OPVRs of 1-plane graphs strongly depends on their connectivity properties; namely, we show that if an n -vertex 1-plane graph G is 2-connected and it can be augmented to become 3-connected only at the expenses of loosing its 1-planarity, then the vertex complexity of any OPVR of G may be $\Omega(n)$. Also, for these graphs we show that a 1-planar embedding that guarantees constant vertex complexity can be computed in $O(n)$ time under the assumption that they do not have a certain type of crossing configuration.

We shall distinguish between the crossing configurations depicted in Fig. 4. Figure 4(a) is a *B-configuration* if the dotted edges are missing, and it is an *augmented B-configuration* otherwise. The crossing configurations of Figs. 4(b), (c), and (d) are a *kite*, a *W-configuration*, and a *T-configuration*, respectively [5, 33]. Figure 4(e) depicts an *augmented T-configuration*. In the following we shall also refer to *crossing augmented 1-plane graphs* [7]. A 1-plane graph G is crossing augmented, when for each pair of crossing edges (u, v) and (w, z) , the subgraph of G induced by $\{u, v, w, z\}$ is a K_4 . We call *cycle edges* of (u, v) and (w, z) the four edges of the K_4 different from (u, v) and (w, z) (they form a 4-cycle). Note that a 1-plane graph can always be made crossing augmented in $O(n)$ time, by adding the missing cycle edges without introducing any new crossings [2, 7, 29].

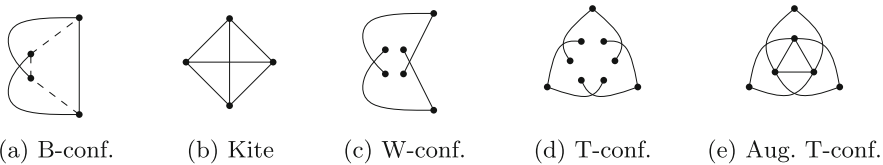


Fig. 4. Crossing configurations of 1-plane graphs.

Edge Partitions. An *edge partition* of a 1-plane graph G is a coloring of its edges with one of two colors, *red* and *blue*, such that both the red graph G_R induced by the red edges and the blue graph G_B induced by the blue edges are plane.

Theorem 3. *Let G be a 3-connected 1-plane graph with n vertices. There is an edge partition of G such that the red graph has maximum vertex degree six and this bound is worst case optimal. Also, such an edge edge partition can be computed in $O(n)$ time.*

Proof Sketch: We assume that G is crossing augmented. The proof relies on claims that describe properties of the cycle edges of G which make it possible to construct the desired partition of the edges of G .

Claim 1. *There are no two cycle edges of G that cross each other.*

Claim 2. *Any edge of G is the cycle edge of at most two pairs of crossing edges.*

Let G_p be the plane graph obtained from G by removing an edge for each pair of crossing edges. We can arbitrarily choose what edges to remove, provided that we never remove a cycle edge. Claim 1 ensures that this choice is always feasible. Let G_p^+ be a plane graph obtained by edge-augmenting G_p so to become a plane triangulation. We apply a *Schnyder trees decomposition* to G_p^+ , so to find an orientation of its internal edges such that each internal vertex has exactly three outgoing edges and the vertices of the outer face have no outgoing edge. Finally, we arbitrarily orient the edges of the outer face of G_p^+ .

Claim 3. *Let (u, v) and (w, z) be two crossing edges of G . Then either $\{u, v\}$ or $\{w, z\}$ have both an outgoing edge in G_p^+ , that is a cycle edge of (u, v) and (w, z) .*

We use Claim 3 to partition the edge set of G as follows. For each pair of crossing edges (u, v) and (w, z) of G we color with the red color the edge connecting the pair, $\{u, v\}$ or $\{w, z\}$, for which Claim 3 holds. By this choice, each end-vertex of a red edge has one outgoing edge among the cycle edges of (u, v) and (w, z) . Since every vertex is incident to at most three outgoing edges in G_p^+ , and since each edge is the cycle edge of at most two pairs of crossing edges (Claim 2), by this procedure at most six edges for each vertex get the red color. The proof that this bound on the vertex degree of G_R is tight uses a graph constructed as follows: Start with a sufficiently large plane triangulation G_p , and insert an augmented T-configuration inside every face of G_p . The tightness of the bound can then be derived by a counting argument based on Euler’s formula. The linear time complexity follows from the fact that G has $O(n)$ edges (see e.g. [29]) and that Schnyder trees can be constructed in $O(n)$ time [26]. \square

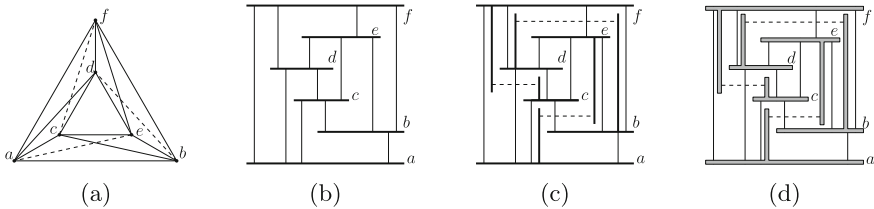


Fig. 5. (a) An edge partition of a 1-plane graph G ; red (blue) edges are dashed (solid). (b) A strong BVR γ_B of G_B . (c) Insertion of the red edges into γ_B . (d) An OPVR of G .

Vertex Complexity Bounds for 3-Connected 1-Plane Graphs. Theorem 3 can be used to construct an OPVR of a 3-connected 1-plane graph whose vertex complexity does not depend on the input size. The idea for this construction is as follows. Let G_B and G_R be the plane graphs defined by the edge partition of Theorem 3; see e.g. Fig. 5(a). Under the assumption that G is crossing augmented, it can be proved that G_B is 2-connected, which implies that it admits a strong BVR γ_B (this can be computed in $O(n)$ time [31]); see e.g. Fig. 5(b). Assume that two vertices u and v are connected by a red edge and let $\gamma_B(u)$ and $\gamma_B(v)$ be the horizontal bars representing them. We attach a vertical bar to $\gamma_B(u)$ and a vertical bar to $\gamma_B(v)$ such that each vertical bar shares an end-vertex with the horizontal bar and the two vertical bars can see each other horizontally. This makes it possible to draw the horizontal red edge (u, v) ; see e.g. Fig. 5(c). Once all red edges have been added to γ_B , every vertex v is represented as a “rake”-shaped object consisting of one horizontal bar and at most six vertical bars (we have a vertical bar for each red edge incident to v and there are at most six such edges). This “rake”-shaped object can then be used as the skeleton of an orthogonal polygon that has two reflex corners per vertical bar; see e.g. Fig. 5(d).

Theorem 4. *Let G be a 3-connected 1-plane graph with n vertices. There exists an $O(n)$ -time algorithm that computes an embedding-preserving OPVR of G with vertex complexity at most 12, on an integer grid of size $O(n) \times O(n)$.*

Based on Theorem 4, we can significantly improve the time complexity of an algorithm that computes an optimal OPVR.

Theorem 5. *Let G be a 3-connected 1-plane graph with n vertices. There exists an $O(n^{\frac{7}{4}}\sqrt{\log n})$ -time algorithm that computes an embedding-preserving optimal OPVR γ of G , on an integer grid of size $O(n) \times O(n)$. Also, γ has the minimum number of total reflex corners among all the embedding preserving optimal OPVRs of G .*

The following lower bound can be proved.

Theorem 6. *There is an infinite family \mathcal{G} of 3-connected 1-plane graphs such that for any graph G of \mathcal{G} , any embedding preserving OPVR has vertex complexity at least two.*

2-Connected 1-Plane Graphs. The next theorem shows a lower bound on the vertex complexity of 2-connected 1-planar graphs (and that cannot be augmented to become 3-connected without losing 1-planarity).

Theorem 7. *For every positive integer n , there exists a 2-connected 1-planar graph G with $O(n)$ vertices such that, for every 1-planar embedding of G , any embedding preserving OPVR of G has vertex complexity $\Omega(n)$.*

Proof Sketch: We prove the claim for a fixed 1-planar embedding (the proof can be easily extended to all 1-planar embeddings of G). Consider the 1-plane

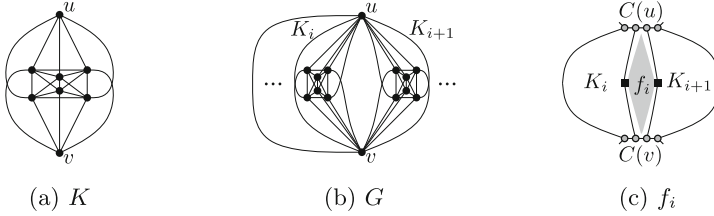


Fig. 6. Illustration for the proof of Theorem 7.

graph K in Fig. 6(a). It has 2 vertices on its outer face, u and v , plus 6 inner vertices. We now construct G as follows. Attach $n + 1$ copies K_1, \dots, K_{n+1} of K all sharing u and v . The copies are attached in parallel without introducing any further crossing, as shown in Fig. 6(b). Also connect u and v with an edge on the outer face. The resulting graph G has $8(n + 1) - 2n = 6n + 8 = O(n)$ vertices. Also, G is 2-connected and 1-plane by construction, hence it admits an OPVR by Theorem 2. Consider now an embedding preserving OPVR of G and the corresponding orthogonal drawing Γ . Between any two consecutive copies K_i and K_{i+1} ($i = 1, \dots, n$), there is a face f_i of G having two expansion vertices of $C(u)$ (the expansion cycle of u) and two expansion vertices of $C(v)$ on its boundary, together with two dummy vertices; see Fig. 6(c). Each dummy vertex forms one $\frac{\pi}{2}$ angle inside f_i . Each expansion vertex forms one $\frac{\pi}{2}$ angle inside f_i . Hence, there are at least six $\frac{\pi}{2}$ angles inside f_i . Since the original edges of f_i have no bends, by Property 1 the two expansion edges of f_i must form (at least) two $\frac{3\pi}{2}$ angles inside f_i . In Γ there are n of such faces requiring two angles of $\frac{3\pi}{2}$ each from an expansion edge. If every vertex of G is represented by a polygon with vertex complexity at most k , the edges of each expansion cycle form at most $4 + k$ angles of $\frac{3\pi}{2}$ inside their incident faces (that are not expansion cycles). At least ten of these angles are inside the outer face of Γ (Property 1), and hence it must be $(4 + k)2 - 10 \geq 2n$, that is $k \geq n + 1$. \square

The graphs used to prove Theorem 7 contain several W-configurations. For a contrast, we can show that the absence of W-configurations suffices to find a 1-planar embedding that admits an OPVR with constant vertex complexity.

Theorem 8. *Let G be a 2-connected 1-plane graph with n vertices and no W-configurations. A 1-planar OPVR of G with vertex complexity at most 22 on an integer grid of size $O(n) \times O(n)$ can be computed in $O(n)$ time.*

5 Experiments and Open Problems

We implemented the optimization algorithm of Theorem 1 using the GDTToolkit library [12]. To evaluate the performance of the algorithm in practice, we tested it on a large set of 1-plane graphs, which always admit an OPVR (Theorem 2). In addition, we have the following two objectives: (i) Measure the vertex complexity of the computed OPVRs; in particular, for 3-connected 1-plane graphs

we expect values close to the lower bound of 2. (ii) Establishing “how much” the computed drawings look like RVRs. For every computed OPVR with vertex complexity k , we measure the percentage of polygons with i reflex corners ($i \in [0, \dots, k]$). Since our optimization algorithm computes the optimal solution having the minimum number of reflex corners (see Theorem 1), we expect a high number of rectangles. We generated three different subsets of (simple) 1-plane graphs, which we call **GEN**, **BIC**, and **TRIC**, respectively. Each subset consists of 170 graph. The number of vertices of each graph ranges from 20 to 100. The graphs in **GEN** are general 1-plane graphs, while those in **BIC** and in **TRIC** are 2-connected and 3-connected, respectively. All graphs are maximal (no further edges can be added in their embedding while preserving 1-planarity). The experiments confirmed both our expectations. The optimization algorithm took less than 15 s for all instances up to 60 vertices, and about 41 s on the largest instance with 100 vertices on a common laptop. The optimal solutions of all **GEN** graphs required vertex complexity 1, except two of them with vertex complexity 0. The average percentage of rectangles is around 90%, and never below 80% in any instance. Hence, most of the drawing looks like an RVR. The running times for **BIC** and **TRIC** reflect the behavior observed for **GEN** (with some more demanding large instances). For every graph of **TRIC** we found a drawing with vertex complexity either 1 or 2. Most of the **BIC** graphs required vertex complexity 2, some required vertex complexity 3, and only one graph required vertex complexity 4. The percentage of vertices drawn as rectangles is very high also for **BIC** and **TRIC** (around 80% for **BIC** and around 75% for **TRIC**).

The results in this paper naturally raise several interesting open problems. Among them: (1) Close the gap between the upper bound and the lower bound on the vertex complexity of OPVRs of 3-connected 1-plane graphs (see Theorems 5 and 6). (2) We find it interesting to study the problem of computing OPVRs that maximize the number of rectangular vertices, even at the expenses of sub-optimal vertex complexity. (3) Theorem 8 constructs 1-planar embeddings that guarantee constant vertex complexity. What 2-connected 1-plane graphs admit a 1-planar OPVR with constant vertex complexity?

References

1. Ackerman, E.: A note on 1-planar graphs. *Discrete Appl. Math.* **175**, 104–108 (2014)
2. Alam, M.J., Brandenburg, F.J., Kobourov, S.G.: Straight-line grid drawings of 3-connected 1-planar graphs. In: Wismath, S., Wolff, A. (eds.) *GD 2013*. LNCS, vol. 8242, pp. 83–94. Springer, Heidelberg (2013). doi:[10.1007/978-3-319-03841-4_8](https://doi.org/10.1007/978-3-319-03841-4_8)
3. Alam, M.J., Kobourov, S.G., Mondal, D.: Orthogonal layout with optimal face complexity. In: Freivalds, R.M., Engels, G., Catania, B. (eds.) *SOFSEM 2016*. LNCS, vol. 9587, pp. 121–133. Springer, Heidelberg (2016). doi:[10.1007/978-3-662-49192-8_10](https://doi.org/10.1007/978-3-662-49192-8_10)
4. Bannister, M.J., Cabello, S., Eppstein, D.: Parameterized complexity of 1-planarity. In: Dehne, F., Solis-Oba, R., Sack, J.-R. (eds.) *WADS 2013*. LNCS, vol. 8037, pp. 97–108. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-40104-6_9](https://doi.org/10.1007/978-3-642-40104-6_9)

5. Biedl, T.C., Liotta, G., Montecchiani, F.: On visibility representations of non-planar graphs. In: Fekete, S.P., Lubiw, A. (eds) SoCG 2016, LIPIcs, vol. 51, pp. 19:1–19:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016). <http://www.dagstuhl.de/dagpub/978-3-95977-009-5>
6. Brandenburg, F.J.: 1-Visibility representations of 1-planar graphs. *J. Graph Algorithms Appl.* **18**(3), 421–438 (2014)
7. Brandenburg, F.J.: On 4-map graphs, 1-planar graphs, their recognition problem. CoRR, abs/1509.03447 (2015). <http://arxiv.org/abs/1509.03447>
8. Cabello, S., Mohar, B.: Adding one edge to planar graphs makes crossing number and 1-planarity hard. *SIAM J. Comput.* **42**(5), 1803–1829 (2013)
9. Cornelsen, S., Karrenbauer, A.: Accelerated bend minimization. *J. Graph Algorithms Appl.* **16**(3), 635–650 (2012)
10. Czap, J., Hudák, D.: On drawings and decompositions of 1-planar graphs. *Electr. J. Comb.* **20**(2), P54 (2013)
11. Dean, A.M., Hutchinson, J.P.: Rectangle-visibility representations of bipartite graphs. *Discrete Appl. Math.* **75**(1), 9–25 (1997)
12. Di Battista, G., Didimo, W.: GDrawTool. In: Tamassia, R. (ed.) *Handbook of Graph Drawing and Visualization*, pp. 571–597. CRC Press, Boca Raton (2013)
13. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: *Graph Drawing*. Prentice-Hall, Upper Saddle River (1999)
14. Di Giacomo, E., Didimo, W., Evans, W.S., Liotta, G., Meijer, H., Montecchiani, F., Wismath, S.K.: Ortho-polygon visibility representations of embedded graphs. ArXiv e-prints, abs/1604.08797v2 (2016). <http://arxiv.org/abs/1604.08797v2>
15. Duchet, P., Hamidoune, Y., Las Vergnas, M., Meyniel, H.: Representing a planar graph by vertical lines joining different levels. *Discrete Math.* **46**(3), 319–321 (1983)
16. Eades, P., Hong, S.-H., Katoh, N., Liotta, G., Schweitzer, P., Suzuki, Y.: A linear time algorithm for testing maximal 1-planarity of graphs with a rotation system. *Theor. Comput. Sci.* **513**, 65–76 (2013)
17. Eades, P., Liotta, G.: Right angle crossing graphs and 1-planarity. *Discrete Appl. Math.* **161**(7–8), 961–969 (2013)
18. Evans, W.S., Kaufmann, M., Lenhart, W., Mchedlidze, T., Wismath, S.K.: Bar 1-visibility graphs vs. other nearly planar graphs. *J. Graph Algorithms Appl.* **18**(5), 721–739 (2014)
19. Garg, A., Tamassia, R.: A new minimum cost flow algorithm with applications to graph drawing. In: North, S. (ed.) *GD 1996*. LNCS, vol. 1190, pp. 201–216. Springer, Heidelberg (1997). doi:10.1007/3-540-62495-3_49
20. Hutchinson, J.P., Shermer, T.C., Vince, A.: On representations of some thickness-two graphs. *Comput. Geom.* **13**(3), 161–171 (1999)
21. Kant, G., Liotta, G., Tamassia, R., Tollis, I.G.: Area requirement of visibility representations of trees. *Inf. Process. Lett.* **62**(2), 81–88 (1997)
22. Korzhik, V.P., Mohar, B.: Minimal obstructions for 1-immersions and hardness of 1-planarity testing. *J. Graph Theory* **72**(1), 30–71 (2013)
23. Lenhart, W.J., Liotta, G., Montecchiani, F.: On partitioning the edges of 1-planar graphs. CoRR, abs/1511.07303 (2015). <http://arxiv.org/abs/1511.07303>
24. Otten, R.H.J.M., Van Wijk, J.G.: Graph representations in interactive layout design. In: *IEEE ISCSS*, pp. 914–918. IEEE (1978)
25. Rosenstiehl, P., Tarjan, R.E.: Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete Comput. Geom.* **1**, 343–353 (1986)
26. Schnyder, W.: Embedding planar graphs on the grid. In: Johnson, D.S. (ed.), *SODA 1990*, pp. 138–148. SIAM (1990)

27. Shermer, T.C.: On rectangle visibility graphs III. External visibility and complexity. In: Fiala, F., Kranakis, E., Sack, J.-R., (eds.) CCCG 1996, pp. 234–239. Carleton University Press (1996)
28. Streinu, I., Whitesides, S.: Rectangle visibility graphs: characterization, construction, and compaction. In: Alt, H., Habib, M. (eds.) STACS 2003. LNCS, vol. 2607, pp. 26–37. Springer, Heidelberg (2003). doi:[10.1007/3-540-36494-3_4](https://doi.org/10.1007/3-540-36494-3_4)
29. Suzuki, Y.: Re-embeddings of maximum 1-planar graphs. *SIAM J. Discrete Math.* **24**(4), 1527–1540 (2010)
30. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comp.* **16**(3), 421–444 (1987)
31. Tamassia, R., Tollis, I.G.: A unified approach to visibility representations of planar graphs. *Discrete Comput. Geom.* **1**(1), 321–341 (1986)
32. Thomassen, C.: Plane representations of graphs. In: *Progress in Graph Theory*, pp. 43–69. AP (1984)
33. Thomassen, C.: Rectilinear drawings of graphs. *J. Graph Theory* **12**(3), 335–341 (1988)
34. Wismath, S.K.: Characterizing bar line-of-sight graphs. In: Rourke, J.O. (ed), *SoCG 1985*, pp. 147–152. ACM (1985)