Bend-Minimum Orthogonal Drawings in Quadratic Time

Walter Didimo[1(✉)], Giuseppe Liotta[1], and Maurizio Patrignani[2]

[1] Università degli Studi di Perugia, Perugia, Italy
{walter.didimo,giuseppe.liotta}@unipg.it
[2] Università Roma Tre, Rome, Italy
maurizio.patrignani@uniroma3.it

Abstract. Let G be a planar 3-graph (i.e., a planar graph with vertex degree at most three) with n vertices. We present the first $O(n^2)$-time algorithm that computes a planar orthogonal drawing of G with the minimum number of bends in the variable embedding setting. If either a distinguished edge or a distinguished vertex of G is constrained to be on the external face, a bend-minimum orthogonal drawing of G that respects this constraint can be computed in $O(n)$ time. Different from previous approaches, our algorithm does not use minimum cost flow models and computes drawings where every edge has at most two bends.

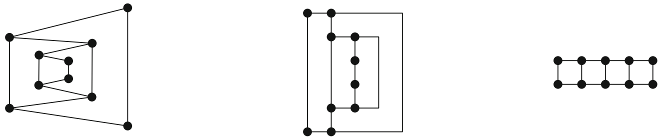
1 Introduction

A pioneering paper by Storer [22] asks whether a crossing-free orthogonal drawing with the minimum number of bends can be computed in polynomial time. The question posed by Storer is in the fixed embedding setting, i.e., the input is a plane 4-graph (an embedded planar graph with vertex degree at most four) and the wanted output is an embedding-preserving orthogonal drawing with the minimum number of bends. Tamassia [23] answers Storer's question in the affirmative by describing an $O(n^2 \log n)$-time algorithm. The key idea of Tamassia's result is the equivalence between the bend minimization problem and the problem of computing a min-cost flow on a suitable network. To date, the most efficient known solution of the bend-minimization problem for orthogonal drawings in the fixed embedding setting is due to Cornelsen and Karrenbauer [6], who show a novel technique to compute a min-cost flow on an uncapacitated network and apply this technique to Tamassia's model achieving $O(n^{\frac{3}{2}})$-time complexity.

A different level of complexity for the bend minimization problem is encountered in the variable embedding setting, that is when the algorithm is asked to find a bend-minimum solution over all planar embeddings of the graph. For example, the orthogonal drawing of Fig. 1(c) has a different planar embedding

Research supported in part by the project: “Algoritmi e sistemi di analisi visuale di reti complesse e di grandi dimensioni - Ricerca di Base 2018, Dipartimento di Ingegneria dell'Università degli Studi di Perugia” and by MIUR project “MODE – MOrphing graph Drawings Efficiently”, prot. 20157EFM5C_001.

T. Biedl and A. Kerren (Eds.): GD 2018, LNCS 11282, pp. 481–494, 2018.
https://doi.org/10.1007/978-3-030-04414-5_34



(a) (b) (c)

Fig. 1. (a) A planar embedded 3-graph G. (b) An embedding-preserving bend-minimum orthogonal drawing of G. (c) A bend-minimum orthogonal drawing of G.

than the graph of Fig. 1(a) and it has no bends, while the drawing of Fig. 1(b) preserves the embedding but it is suboptimal in terms of bends.

Garg and Tamassia [13] prove that the bend-minimization problem for orthogonal drawings is NP-complete for planar 4-graphs, while Di Battista et al. [8] show that it can be solved in $O(n^5 \log n)$ time for planar 3-graphs. Generalizations of the problem in the variable embedding setting where edges have some flexibility (i.e., they can bend a few times without cost for the optimization function) have also been the subject of recent studies by Bläsius et al. [2].

Improving the $O(n^5 \log n)$ time complexity of the algorithm by Di Battista et al. [8] has been an elusive open problem for more than a decade (see, e.g., [3]), until a paper by Chang and Yen [4] has shown how to compute a bend-minimum orthogonal drawing of a planar 3-graph in the variable embedding setting in $\tilde{O}(n^{\frac{17}{7}})$ time, which can be read as $O(n^{\frac{17}{7}} \log^k n)$ time for a positive constant k.

Similar to [8], the approach in [4] uses an SPQR-tree to explore all planar embeddings of a planar 3-graph and combines partial solutions associated with the nodes of this tree to compute a bend-minimum drawing. Both in [8] and in [4], the computationally most expensive task is computing min-cost flows on suitable variants of Tamassia's network. However, Chang and Yen elegantly prove that a simplified flow network where all edges have unit capacity can be adopted to execute this task. This, combined with a recent result [5] about min-cost flows on unit-capacity networks, yields the improved time complexity.

Contribution and Outline. This paper provides new algorithms to compute bend-minimum orthogonal drawings of planar 3-graphs, which improve the time complexity of the state-of-the-art solution. We prove the following.

Theorem 1. *Let G be an n-vertex planar 3-graph. A bend-minimum orthogonal drawing of G can be computed in $O(n^2)$ time. If either a distinguished edge or a distinguished vertex of G is constrained to be on the external face, a bend-minimum orthogonal drawing of G that respects the given constraint can be computed in $O(n)$ time. Furthermore, the computed drawings have at most two bends per edge, which is worst-case optimal.*

As in [8] and in [4], the algorithmic approach of Theorem 1 computes a bend-minimum orthogonal drawing by visiting an SPQR-tree of the input graph. However, it does not need to compute min-cost flows at any steps of the visit, which is the fundamental difference with the previous techniques. This makes it possible to design the first quadratic-time algorithm to compute bend-minimum orthogonal drawings of planar 3-graphs in the variable embedding setting.

The second part of the statement of Theorem 1 extends previous studies by Nishizeki and Zhou [26], who give a first example of a linear-time algorithm in the variable embedding setting for planar 3-graphs that are partial two-trees. The bend-minimum drawings of Theorem 1 have at most two bends per edge, which is a desirable property for an orthogonal representation. We recall that every planar 4-graph (except the octahedron) has an orthogonal drawing with at most two bends per edge [1,17], but minimizing the number of bends may require some edges with a $\Omega(n)$ bends [8,24]. It is also proven that every planar 3-graph (except K_4) has an orthogonal drawing with at most one bend per edge [16], but the drawings of the algorithm in [16] are not bend-minimum. Finally, a non-flow based algorithm having some similarities with ours is given in [12]; it neither computes bend-minimum drawings nor guarantees at most two bends per edge.

The paper is organized as follows. Preliminary definitions and results are in Sect. 2. In Sect. 3 we prove key properties of bend-minimum orthogonal drawings of planar 3-graphs used in our approach. Sect. 4 describes our drawing algorithms. Open problems are in Sect. 5. All full proofs and more figures can be found in [11].

2 Preliminaries

We assume familiarity with basic definitions on graph connectivity and planarity (see Appendix A of [11]). If G is a graph, $V(G)$ and $E(G)$ denote the sets of vertices and edges of G. We consider *simple* graphs, i.e., graphs with neither self-loops nor multiple edges. The *degree* of a vertex $v \in V(G)$, denoted as $\deg(v)$, is the number of its neighbors. $\Delta(G)$ denotes the maximum degree of a vertex of G; if $\Delta(G) \leq h$ ($h \geq 1$), G is an *h-graph.* A graph G is *rectilinear planar* if it admits a planar drawing where each edge is either a horizontal or a vertical segment (i.e., it has no bend). Rectilinear planarity testing is NP-complete for planar 4-graphs [13], but it is polynomially solvable for planar 3-graphs [4,8] and linear-time solvable for subdivisions of planar triconnected cubic graphs [18]. By extending a result of Thomassen [25] on those 3-graphs that have a rectilinear drawing with all rectangular faces, Rahman et al. [21] characterize rectilinear plane 3-graphs. For a plane graph G, let $C_o(G)$ be its external cycle ($C_o(G)$ is simple if G is biconnected). Also, if C is a simple cycle of G, $G(C)$ is the plane subgraph of G that consists of C and of the vertices and edges inside C. An edge $e = (u, v) \notin E(G(C))$ is a *leg* of C if exactly one of the vertices u and v belongs to C; such a vertex is a *leg-vertex* of C. If C has exactly k legs and no edge embedded outside C joins two of its vertices, C is a *k-legged cycle* of G.

Theorem 2. [21] *Let G be a biconnected plane 3-graph. G admits an orthogonal drawing without bends if and only if:* (*i*) *$C_o(G)$ contains at least four vertices of degree 2;* (*ii*) *each 2-legged cycle contains at least two vertices of degree 2;* (*iii*) *each 3-legged cycle contains at least one vertex of degree 2.*

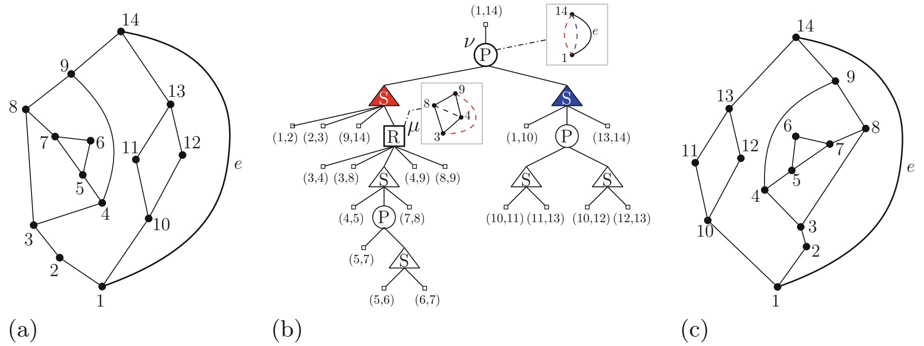


Fig. 2. (a) A plane 3-graph G. (b) The SPQR-tree of G with respect to e; the skeletons of a P-node ν and of an R-node μ are shown. (c) A different embedding of G obtained by changing the embedding of skel(ν) and of skel(μ).

As in [21], we call *bad* any 2-legged and any 3-legged cycle that does not satisfy Condition (*ii*) and (*iii*) of Theorem 2, respectively.

SPQR-Trees of Planar 3-Graphs. Let G be a biconnected graph. An *SPQR-tree* T of G represents the decomposition of G into its triconnected components and can be computed in linear time [7,14,15]. Each triconnected component corresponds to a node μ of T; the triconnected component itself is called the *skeleton* of μ and denoted as skel(μ). A node μ of T can be of one of the following types: (*i*) *R-node*, if skel(μ) is a triconnected graph; (*ii*) *S-node*, if skel(μ) is a simple cycle of length at least three; (*iii*) *P-node*, if skel(μ) is a bundle of at least three parallel edges; (*iv*) *Q-nodes*, if it is a leaf of T; in this case the node represents a single edge of the graph and its skeleton consists of two parallel edges. Note that, neither two S- nor two P-nodes are adjacent in T. A *virtual edge* in skel(μ) corresponds to a tree node ν adjacent to μ in T. If T is rooted at one of its Q-nodes ρ, every skeleton (except the one of ρ) contains exactly one virtual edge that has a counterpart in the skeleton of its parent: This virtual edge is the *reference edge* of skel(μ) and of μ, and its endpoints are the *poles* of skel(μ) and of μ. The edge of G corresponding to the root ρ of T is the *reference edge* of G, and T is the SPQR-tree of G *with respect to e*. For every node $\mu \neq \rho$ of T, the subtree T_μ rooted at μ induces a subgraph G_μ of G called the *pertinent graph* of μ, which is described by T_μ in the decomposition: The edges of G_μ correspond to the Q-nodes (leaves) of T_μ. Graph G_μ is also called a *component* of G with respect to the reference edge e, namely G_μ is a P-, an R-, or an S-component depending on whether μ is a P-, an R-, or an S-component, respectively.

The SPQR-tree T rooted at a Q-node ρ implicitly describes all planar embeddings of G with the reference edge of G on the external face. All such embeddings are obtained by combining the different planar embeddings of the skeletons of P- and R-nodes: For a P-node μ, the different embeddings of skel(μ) are the different permutations of its non-reference edges. If μ is an R-node, skel(μ) has two possible planar embeddings, obtained by flipping skel(μ) minus its reference edge at its poles. See Fig. 2 for an illustration. The child node of ρ and its pertinent graph are called the *root child* of T and the *root child component* of G, respectively. An *inner node* of T is neither the root nor the root child of T. The pertinent graph of an inner node is an *inner component* of G. The next lemma gives basic properties of T when $\Delta(G) \leq 3$.

Lemma 1. *Let G be a biconnected planar 3-graph and let T be the SPQR-tree of G with respect to a reference edge e. The following properties hold:*

T1 *Each P-node μ has exactly two children, one being an S-node and the other being an S- or a Q-node; if μ is the root child, both its children are S-nodes.*
T2 *Each child of an R-node is either an S-node or a Q-node.*
T3 *For each inner S-node μ, the edges of $\text{skel}(\mu)$ incident to the poles of μ are (real) edges of G. Also, there cannot be two incident virtual edges in $\text{skel}(\mu)$.*

3 Properties of Bend-Minimum Orthogonal Representations of Planar 3-Graphs

We prove relevant properties of bend-minimum orthogonal drawings of planar 3-graphs that are independent of vertex and bend coordinates, but only depend on the vertex angles and edge bends. To this aim, we recall the concept of *orthogonal representation* [23] and define some types of "shapes" that we use to construct bend-minimum orthogonal representations.

Orthogonal Representations. Let G be a plane 3-graph. If $v \in V(G)$ and if e_1 and e_2 are two (possibly coincident) edges incident to v that are consecutive in the clockwise order around v, we say that $a = \langle e_1, v, e_2 \rangle$ is an *angle at v* of G or simply an *angle* of G. Let Γ and Γ' be two embedding-preserving orthogonal drawings of G. We say that Γ and Γ' are *equivalent* if: (i) For any angle a of G, the geometric angle corresponding to a is the same in Γ and Γ', and (ii) for any edge $e = (u, v)$ of G, the sequence of left and right bends along e moving from u to v is the same in Γ and in Γ'. An *orthogonal representation H* of G is a class of equivalent orthogonal drawings of G; H can be described by the embedding of G together with the geometric value of each angle of G (90, 180, 270°)[1] and with the sequence of left and right bends along each edge. Figure 3(a) shows a bend-minimum orthogonal representation of the graph in Fig. 2(a).

Let p be a path between two vertices u and v in H. The *turn number* of p is the absolute value of the difference between the number of right and the number of left turns encountered along p moving from u to v (or vice versa). The turn number of p is denoted by $t(p)$. A turn along p is caused either by a bend on an edge of p or by an angle of 90/270 degrees at a vertex of p. For example, $t(p) = 2$ for the path $p = \langle 3, 4, 5, 6, 7 \rangle$ in the orthogonal representation of Fig. 3(a). We remark that if H is a bend-minimum orthogonal representation, the bends along an edge, going from an end-vertex to the other, are all left or all right turns [23].

[1] Angles of 360 degrees only occur at 1-degree vertices; we can avoid to specify them.

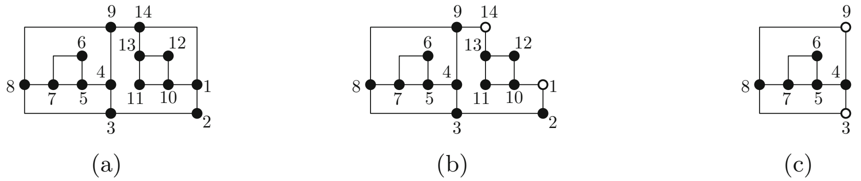


Fig. 3. (a) A bend-minimum orthogonal representation H with four bends of the graph in Fig. 2(a). (b) The component H_ν, which is L-shaped; the two poles of the component are the white vertices. (c) The component H_μ, which is D-shaped.eps

Shapes of Orthogonal Representations. Let G be a biconnected planar 3-graph, T be the SPQR-tree of G with respect to a reference edge $e \in E(G)$, and H be an orthogonal representation of G with e on the external face. For a node μ of T, denote by H_μ the restriction of H to a component G_μ. We also call H_μ a *component* of H. In particular, H_μ is a P-, an R-, or an S-component depending on whether μ is a P-, an R-, or an S-component, respectively. If μ is the root child of T, then H_μ is the *root child component* of H. Denote by u and v the two poles of μ and let p_l and p_r be the two paths from u to v on the external boundary of H_μ, one walking clockwise and the other walking counterclockwise. These paths are the *contour paths* of H_μ. If μ is an S-node, p_l and p_r share some edges (they coincide if H_μ is just a sequence of edges). If μ is either a P- or an R-node, p_l and p_r are edge disjoint; in this case, we define the following *shapes* for H_μ, depending on $t(p_l)$ and $t(p_r)$ and where the poles are external corners:

- H_μ is *C-shaped*, or *-shaped*, if $t(p_l) = 4$ and $t(p_r) = 2$, or vice versa;
- H_μ is *D-shaped*, or *-shaped*, if $t(p_l) = 0$ and $t(p_r) = 2$, or vice versa;
- H_μ is *L-shaped*, or *-shaped*, if $t(p_l) = 3$ and $t(p_r) = 1$, or vice versa;
- H_μ is *X-shaped*, or *-shaped*, if $t(p_l) = t(p_r) = 1$.

For example, H_ν in Fig. 3(b) is -shaped, while H_μ in Fig. 3(c) is -shaped. Concerning S-components, the following lemma rephrases a result in [8, Lemma 4.1], and it is also an easy consequence of Property T3 in Lemma 1.

Lemma 2. *Let H_μ be an inner S-component with poles u and v and let p_1 and p_2 be any two paths connecting u and v in H_μ. Then $t(p_1) = t(p_2)$.*

Based on Lemma 2, we describe the shape of an inner S-component H_μ in terms of the turn number of any path p between its two poles: We say that H_μ is *k-spiral* and has *spirality* k if $t(p) = k$. The notion of spirality of an orthogonal component was introduced in [8]. Differently from [8], we restrict the definition of spirality to inner S-components and we always consider absolute values, instead of both positive and negative values depending on whether the left turns are more or fewer than the right turns. For instance, in the representation of Fig. 3(a) the two series with poles $\{1, 14\}$ (the two filled S-nodes in Fig. 2(b)) have spirality three and one, respectively; the series with poles $\{4, 8\}$ (child of the R-node) has spirality zero, while the series with poles $\{5, 7\}$ has spirality two.

We now give a key result that claims the existence of a bend-minimum orthogonal representation with specific properties for any biconnected planar 3-graph.

This result will be used to design our drawing algorithm. Given an orthogonal representation H, we denote by $\overline{H}$ the orthogonal representation obtained from H by replacing each bend with a dummy vertex: $\overline{H}$ is the *rectilinear image* of H; a dummy vertex in $\overline{H}$ is a *bend vertex*. Also, if w is a degree-2 vertex with neighbors u and v, *smoothing* w is the reverse operation of an edge subdivision, i.e., it replaces the two edges (u, w) and (w, v) with the single edge (u, v).

Lemma 3. *A biconnected planar* 3*-graph* G *with a distinguished edge* e *has a bend-minimum orthogonal representation* H *with* e *on the external face such that:*

O1 *Every edge of* H *has at most two bends, which is worst-case optimal.*
O2 *Every inner P-component or R-component of* H *is either* *- or* *-shaped.*
O3 *Every inner S-component of* H *has spirality at most four.*

Proof (sketch). We prove in three steps the existence of a bend-minimum orthogonal representation H that satisfies O1-O3. We start by a bend-minimum orthogonal representation of G with e on the external face, and in the first step we prove that it either satisfies O1 or it can be locally modified, without changing its planar embedding, so to satisfy O1. In the second step, we prove that from the orthogonal representation obtained in the first step we can derive a new orthogonal representation (still with same embedding) that satisfies O2 in addition to O1. Finally, we prove that this last representation also satisfies O3.

Step 1: Property O1. Suppose that H is a bend-minimum orthogonal representation of G with e on the external face and having an edge g (possibly $g = e$) with at least three bends. Let $\overline{H}$ be the rectilinear image of H, and let $\overline{G}$ be the plane graph underlying $\overline{H}$. Since $\overline{H}$ has no bend, $\overline{G}$ satisfies Conditions $(i)-(iii)$ of Theorem 2. Let v_1, v_2, v_3 be three bend vertices in $\overline{H}$ that correspond to three bends of g in H. Assume first that g is an internal edge of G and let $\overline{G'}$ be the plane graph obtained from $\overline{G}$ by smoothing v_1. We claim that $\overline{G'}$ still satisfies Conditions $(i) - (iii)$ of Theorem 2. Indeed, if this is not the case, there must be a bad cycle in $\overline{G'}$ that contains both v_2 and v_3. This is a contradiction, because no bad cycle can contain two vertices of degree two. Hence, there exists an (embedding-preserving) representation $\overline{H'}$ of $\overline{G'}$ without bends, which is the rectilinear image of an orthogonal representation of G with fewer bends than H, a contradiction. Assume now that g is on the external cycle $C_o(G)$ of G. If $C_o(\overline{G})$ contains more than four vertices of degree two, we can smooth v_1 and apply the same argument as above to contradict the optimality of H (note that, such a smoothing does not violate Condition (i) of Theorem 2). Suppose vice versa that $C_o(\overline{G})$ contains exactly four vertices of degree two (three of them being v_1, v_2, and v_3). In this case, just smoothing v_1 violates Condition (i) of Theorem 2. However, we can smooth v_1 and subdivide an edge of $C_o(\overline{G}) \cap C_o(G)$ (such an edge exists since $C_o(G)$ has at least three edges and, by hypothesis and a simple counting argument, at least one of its edges has no bend in H). The resulting plane graph $\overline{G''}$ still satisfies the three conditions of Theorem 2 and admits a representation $\overline{H''}$ without bends; the representation of which $\overline{H''}$ is the rectilinear image is a bend-minimum orthogonal representation of G with at most two bends per edge. To see that two bends per edge is worst-case optimal, just consider a bend-minimum representation of the complete graph K_4.

Step 2: Property O2. Let H be a bend-minimum orthogonal representation of G that satisfies O1 and let $\overline{H}$ be its rectilinear image. The plane underlying graph $\overline{G}$ of $\overline{H}$ satisfies the three conditions of Theorem 2. Rhaman, Nishizeki, and Naznin [21, Lemma 3] prove that, in this case, $\overline{G}$ has an embedding-preserving orthogonal representation $\overline{H'}$ without bends in which every 2-legged cycle C is either □-shaped or □-shaped, where the two poles of the shape are the two leg-vertices of C. On the other hand, if G_μ is an inner P- or R-component, the external cycle $C_o(G_\mu)$ is a 2-legged cycle of G, where the two leg-vertices of $C_o(G_\mu)$ are the poles of G_μ. Hence, the representation H' of G whose rectilinear image is $\overline{H'}$ satisfies O2, as H'_μ is either □-shaped or □-shaped. Also, the bends of H' are the same as in H, because the bend vertices of $\overline{H}$ coincide with those of $\overline{H'}$. Hence, H' still satisfies O1 and has the minimum number of bends.

Step 3: Property O3. Suppose now that H is a bend-minimum orthogonal representation of G (with e on the external face) that satisfies both O1 and O2. More precisely, assume that $H = H'$ is the orthogonal representation obtained in the previous step, where its rectilinear image $\overline{H}$ is computed by the algorithm of Rhaman et al. [21]. By a careful analysis of how this algorithm works, we prove that each series gets spirality at most four in H (see Appendix B of [11]).

4 Drawing Algorithm

Let G be a biconnected 3-planar graph with a distinguished edge e and let T be the SPQR-tree of G with respect to e. Section 4.1 gives a linear-time algorithm to compute bend-minimum orthogonal representations of the inner components of T. Section 4.2 handles the root child of T to complete a bend-minimum representation with e on the external face and it proves Theorem 1. Lemma 3 allows us to restrict our algorithm to search for a representation satisfying Properties O1-O3.

4.1 Computing Orthogonal Representations for Inner Components

Let T be the SPQR-tree of G with respect to reference edge e and let μ be an inner node of T. A key ingredient of our algorithm is the concept of 'equivalent' orthogonal representations of G_μ. Intuitively, two representations of G_μ are equivalent if one can replace the other in any orthogonal representation of G. Similar equivalence concepts have been used for orthogonal drawings [8,10]. As we shall prove (see Theorem 3), for planar 3-graphs a simpler definition of equivalent representations suffices. If μ is a P- or an R-node, two representations H_μ and H'_μ are *equivalent* if they are both □-shaped or both □-shaped. If μ is an inner S-node, H_μ and H'_μ are *equivalent* if they have the same spirality.

Lemma 4. *If H_μ and H'_μ are two equivalent orthogonal representations of G_μ, the two contour paths of H_μ have the same turn number as those of H'_μ.*

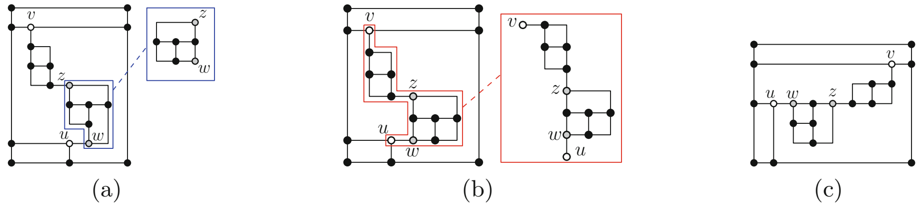


Fig. 4. (a) An orthogonal representation H; a D-shaped R-component with poles $\{w, z\}$ and an equivalent representation of it are in the blue frames. (b) A representation obtained from H by replacing the R-component with the equivalent one; a 1-spiral S-component with poles $\{u, v\}$ and an equivalent one are shown in the red frames. (c) The representation obtained by replacing the S-component with the equivalent one.

Suppose that H_μ is an inner component of H with poles u and v, and let p_l and p_r be the contour paths of H_μ. *Replacing* H_μ in H with an equivalent representation H'_μ means to insert H'_μ in H in place of H_μ, in such a way that: (i) if H_μ and H'_μ are ◻-shaped, the contour path p' of H'_μ for which $t(p') = t(p_l)$ is traversed clockwise from u to v on the external boundary of H'_μ (as for p_l on the external boundary of H_μ); (ii) in all cases, the external angles of H'_μ at u and v are the same as in H_μ. This operation may require to mirror H'_μ (see Fig. 4). The next theorem uses arguments similar to [8].

Theorem 3. *Let H be an orthogonal representation of a planar 3-graph G and H_μ be the restriction of H to G_μ, where μ is an inner component of the SPQR-tree T of G with respect to a reference edge e. Replacing H_μ in H with an equivalent representation H'_μ yields a planar orthogonal representation H' of G.*

We are now ready to describe our drawing algorithm. It is based on a dynamic programming technique that visits bottom-up the SPQR-tree T with respect to the reference edge e of G. Based on Lemma 3 and Theorem 3, the algorithm stores for each visited node μ of T a *set of candidate orthogonal representations* of G_μ, together with their cost in terms of bends. For a Q-node, the set of candidate orthogonal representations consists of three representations, with 0, 1, and 2 bends, respectively. This suffices by Property O1. For a P- or an R-node, the set of candidate representations consists of a bend-minimum ◻-shaped and a bend-minimum ◻-shaped representation. This suffices by Property O2. For an S-node, the set of candidate representations consists of a bend-minimum representation for each value of spirality $0 \leq k \leq 4$. This suffices by Property O3. In the following we explain how to compute the set of candidate representations for a node μ that is a P-, an S-, or an R-node (computing the set of a Q-node is trivial). To achieve overall linear-time complexity, the candidate representations stored at μ are described incrementally, linking the desired representation in the set of the children of μ for each virtual edge of skel(μ).

Candidate Representations for a P-node. By property T1 of Lemma 1, μ has two children μ_1 and μ_2, where μ_1 is an S-node and μ_2 is an S-node or a

Q-node. The cost of the L-shaped representation of μ is the sum of the costs of μ_1 and μ_2 both with spirality one. The cost of the C-shaped representation of μ is the minimum between the cost of μ_1 with spirality two and the cost of μ_2 with spirality two. This immediately implies the following.

Lemma 5. *Let μ be an inner P-node. There exists an $O(1)$-time algorithm that computes a set of candidate orthogonal representations of G_μ, each having at most two bends per edge.*

Candidate Representations for an S-node. By property T3 of Lemma 1, skel(μ) without its reference edge is a sequence of edges such that the first edge and the last edge are real (they correspond to Q-nodes) and at most one virtual edge, corresponding to either a P- or an R-node, appears between two real edges. Let c_0 be the sum of the costs of the cheapest (in terms of bends) orthogonal representations of all P-nodes and R-nodes corresponding to the virtual edges of skel(μ). By Property O2, each of these representations is either C- or L-shaped. Let n_Q be the number of edges of skel(μ) that correspond to Q-nodes and let n_D be the number of edges of skel(μ) that correspond to P- and R-nodes whose cheapest representation is C-shaped. Obviously, any bend-minimum orthogonal representation of G_μ satisfying O2 has cost at least c_0. We have the following.

Lemma 6. *An inner S-component admits a bend-minimum orthogonal representation respecting Properties O1-O3 and with cost c_0 if its spirality $k \leq n_Q + n_D - 1$ and with cost $c_0 + k - n_Q - n_D + 1$ if $k > n_Q + n_D - 1$.*

Note that the possible presence in skel(μ) of virtual edges corresponding to P- and R-nodes whose cheapest representation is L-shaped does not increase the spirality reachable at cost c_0 by the S-node. Lemma 6 also provides an alternative proof of a known result ([8, Lemma 5.2]), stating that for a planar 3-graph the number of bends of a bend-minimum k-spiral representation of an inner S-component does not decrease when k increases. Moreover, since for an inner S-component $n_Q \geq 2$, a consequence of Lemma 6 is Corollary 1. It implies that every bend-minimum k-spiral representation of an inner S-component does not require additional bends with respect to the bend-minimum representations of their subcomponents when $k \in \{0, 1\}$.

Corollary 1. *For each $k \in \{0, 1\}$, every inner S-component admits a bend-minimum orthogonal representation of cost c_0 with spirality k.*

Lemma 7. *Let μ be an inner S-node and n_μ be the number of vertices of* skel(μ)*. There exists an $O(n_\mu)$-time algorithm that computes a set of candidate orthogonal representations of G_μ, each having at most two bends per edge.*

Candidate Representations for an R-node. If μ is an R-node, its children are S- or Q-nodes (Property T2 of Lemma 1). To compute a bend-minimum orthogonal representation of G_μ that satisfies Properties O1-O3, we devise a variant of the linear-time algorithm by Rahman, Nakano, and Nishizeki [19] that exploits the properties of inner S-components.

Lemma 8. *Let μ be an inner R-node and n_μ be the number of vertices of* skel(μ). *There exists an $O(n_\mu)$-time algorithm that computes a set of candidate orthogonal representations of G_μ, each having at most two bends per edge.*

Proof (sketch). Let $\{u, v\}$ be the poles of μ. Our algorithm works in two steps. First, it computes an □-shaped orthogonal representation $\tilde{H}_\mu^{□}$ and a □-shaped orthogonal representation $\tilde{H}_\mu^{□}$ of $\tilde{G}_\mu = \text{skel}(\mu) \setminus (u, v)$, with a variant of the recursive algorithm in [19]. Then, it computes a bend-minimum □-shaped representation $H_\mu^{□}$ and a bend-minimum □-shaped representation $H_\mu^{□}$ of G_μ, by replacing each virtual edge e_S in each of $\tilde{H}_\mu^{□}$ and $\tilde{H}_\mu^{□}$ with the representation in the set of the corresponding S-node whose spirality equals the number of bends of e_S. Every time the algorithm needs to insert a degree-2 vertex along an edge of a bad cycle, it adds this vertex on a virtual edge, if such an edge exists. By Corollary 1, this vertex does not cause an additional bend in the final representation when the virtual edge is replaced by the corresponding S-component.

4.2 Handling the Root Child Component

Let T be the SPQR-tree of G with respect to edge $e = (u, v)$ and let μ be the root child of T. Assuming to have already computed the set of candidate representations for the children of μ, we compute an orthogonal representation H_μ of G_μ and a bend-minimum orthogonal representation H of G (with e on the external face) depending on the type of μ.

Algorithm P-root-child. Let μ be a P-node with children μ_1 and μ_2. By Property T1 of Lemma 1, both μ_1 and μ_2 are S-nodes. Let k_1 (k_2) be the maximum spirality of a representation H_{μ_1} (H_{μ_2}) at the same cost $c_{0,1}$ ($c_{0,2}$) as a 0-spiral representation. W.l.o.g., let $k_1 \geq k_2$. We have three cases:

Case 1: $k_1 \geq 4$. Compute a □-shaped H_μ by merging a 4-spiral and a 2-spiral representation of μ_1 and μ_2, respectively; add e with 0 bends to get H. Case 2: $k_1 = 3$. Compute an □-shaped H_μ by merging a 3-spiral and a 1-spiral representation of μ_1 and μ_2, respectively; add e with 1 bend to get H. Case 3: $k_1 = 2$ or $k_2 = k_1 = 1$. Compute a □-shaped H_μ by merging a 2-spiral and a 0-spiral representation of μ_1 and μ_2, respectively; add e with 2 bends to get H.

Lemma 9. *P-root-child computes a bend-minimum orthogonal representation of G with e on the external face and at most two bends per edge in $O(1)$ time.*

Algorithm S-root-child. Let μ be an S-node. if G_μ starts and ends with one edge, we compute the candidate orthogonal representations of G_μ as if it were an inner S-node, and we obtain H by adding e with zero bends to the 2-spiral representation of G_μ. Else, if G_μ only starts or ends with one edge, we add e to the other end of G_μ, compute the candidate representations of $G_\mu \cup \{e\}$ as if it were an inner S-node, and obtain G by adopting the representation of $G_\mu \cup \{e\}$ with spirality 3 and by identifying the first and last vertex. Finally, if skel$(\mu) \setminus \{e\}$ starts and ends with an R- or a P-node, we add two copies e', e'' of e

at the beginning and at the end of G_μ, compute the candidate representations of $G_\mu \cup \{e', e''\}$ as if it were an inner S-node, and obtain H from the representation of $G_\mu \cup \{e', e''\}$ with spirality 4, by identifying the first and last vertex of $G_\mu \cup \{e', e''\}$ and by smoothing the resulting vertex.

Lemma 10. *S-root-child computes a bend-minimum orthogonal representation of G with e on the external face and at most two bends per edge in $O(n_\mu)$ time, where n_μ is the number of vertices of* $\text{skel}(\mu)$.

Algorithm R-root-child. Let μ be an R-node and let ϕ_1 and ϕ_2 be the two planar embeddings of $\text{skel}(\mu)$ obtained by choosing as external face one of those incident to e. For each ϕ_i, compute an orthogonal representation H_i of G by: (i) finding a representation $\widetilde{H}_i$ of $\text{skel}(\mu)$ (included e) with the variant of [19] given in the proof of Lemma 8, but this time assuming that all the four designated corners of the external face in the initial step must be found; (ii) replacing each virtual edge that bends $k \geq 0$ times in $\widetilde{H}_i$ with a minimum-bend k-spiral representation of its corresponding S-component. H is the cheapest of H_1 and H_2. Since the variant of [19] applied to $\text{skel}(\mu)$ still causes at most two bends per edge, with the same arguments as in Lemma 8 we have:

Lemma 11. *R-root-child computes a bend-minimum orthogonal representation of G with e on the external face and at most two bends per edge in $O(n_\mu)$ time, where n_μ be the number of vertices of* $\text{skel}(\mu)$.

Proof of Theorem 1. If G is biconnected, Lemmas 5, 7, 8, 9–11 yield an $O(n)$-time algorithm that computes a bend-minimum orthogonal representation of G with a distinguished edges e on the external face and at most two bends per edge. Call BendMin-RefEdge this algorithm. An extension of BendMin-RefEdge to a simply-connected graph G, which still runs in $O(n)$ time, is easily derivable by exploiting the block-cut-vertex tree of G (see Appendix C of [11]). Running BendMin-RefEdge for every possible reference edge, we find in $O(n^2)$ time a bend-minimum orthogonal representation of G over all its planar embeddings. If v is a distinguished vertex of G, running BendMin-RefEdge for every edge incident to v, we find in $O(n)$ time a bend-minimum orthogonal representation of G with v on the external face (recall that $\deg(v) \leq 3$). Finally, an orthogonal drawing of G is computed in $O(n)$ time from an orthogonal representation of G [7].

5 Open Problems

We suggest two research directions related to our results: (i) Is there an $O(n)$-time algorithm to compute a bend-minimum orthogonal drawing of a 3-connected planar cubic graph, for every possible choice of the external face? (ii) It is still unknown whether an $O(n)$-time algorithm for the bend-minimization problem in the fixed embedding setting exists [9]. This problem could be tackled with non-flow based approaches. A positive result in this direction is given in [20] for plane 3-graphs.

References

1. Biedl, T.C., Kant, G.: A better heuristic for orthogonal graph drawings. Comput. Geom. **9**(3), 159–180 (1998). https://doi.org/10.1016/S0925-7721(97)00026-6
2. Bläsius, T., Rutter, I., Wagner, D.: Optimal orthogonal graph drawing with convex bend costs. ACM Trans. Algorithms **12**(3), 33:1–33:32 (2016). https://doi.org/10.1145/2838736
3. Brandenburg, F., Eppstein, D., Goodrich, M.T., Kobourov, S., Liotta, G., Mutzel, P.: Selected open problems in graph drawing. In: Liotta, G. (ed.) GD 2003. LNCS, vol. 2912, pp. 515–539. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24595-7_55
4. Chang, Y., Yen, H.: On bend-minimized orthogonal drawings of planar 3-graphs. In: Aronov, B., Katz, M.J. (eds.) 33rd International Symposium on Computational Geometry, SoCG 2017, 4–7 July 2017, Brisbane, Australia. LIPIcs, vol. 77, pp. 29:1–29:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017). https://doi.org/10.4230/LIPIcs.SoCG.2017.29, http://www.dagstuhl.de/dagpub/978-3-95977-038-5
5. Cohen, M.B., Madry, A., Tsipras, D., Vladu, A.: Matrix scaling and balancing via box constrained newton's method and interior point methods. In: Umans, C. (ed.) 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15–17, 2017. pp. 902–913. IEEE Computer Society (2017). https://doi.org/10.1109/FOCS.2017.88, http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8100284
6. Cornelsen, S., Karrenbauer, A.: Accelerated bend minimization. J. Graph Algorithms Appl. **16**(3), 635–650 (2012). https://doi.org/10.7155/jgaa.00265
7. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice-Hall, Englewood Cliffs (1999)
8. Di Battista, G., Liotta, G., Vargiu, F.: Spirality and optimal orthogonal drawings. SIAM J. Comput. 27(6), 1764–1811 (1998). https://doi.org/10.1137/S0097539794262847
9. Di Giacomo, E., Liotta, G., Tamassia, R.: Graph drawing. In: Goodman, J., O'Rourke, J., Toth, C. (eds.) Handbook of Discrete and Computational Geometry, 3rd edn, pp. 1451–1477. Chapman and Hall/CRC, Boca Raton (2017)
10. Didimo, W., Liotta, G., Patrignani, M.: On the Complexity of HV-rectilinear Planarity Testing. In: Duncan, C., Symvonis, A. (eds.) GD 2014. LNCS, vol. 8871, pp. 343–354. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45803-7_29
11. Didimo, W., Liotta, G., Patrignani, M.: Bend-minimum orthogonal drawings inquadratic time. CoRR 1804.05813v3 (2018). http://arxiv.org/abs/1804.05813v3
12. Garg, A., Liotta, G.: Almost bend-optimal planar orthogonal drawings of biconnected degree-3 planar graphs in quadratic time. In: Kratochvíyl, J. (ed.) GD 1999. LNCS, vol. 1731, pp. 38–48. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-46648-7_4
13. Garg, A., Tamassia, R.: On the computational complexity of upward and rectilinear planarity testing. SIAM J. Comput. **31**(2), 601–625 (2001). https://doi.org/10.1137/S0097539794277123
14. Gutwenger, C., Mutzel, P.: A linear time implementation of SPQR-trees. In:Marks, J. (ed.) Graph Drawing, 8th International Symposium, GD 2000,Colonial Williamsburg, VA, USA, September 20-23, 2000, Proceedings. LectureNotes in Computer Science, vol. 1984, pp. 77–90. Springer (2000).https://doi.org/10.1007/3-540-44541-2_8

15. Hopcroft, J.E., Tarjan, R.E.: Dividing a graph into triconnected components. SIAM J. Comput. **2**(3), 135–158 (1973). https://doi.org/10.1137/0202012
16. Kant, G.: Drawing planar graphs using the canonical ordering. Algorithmica 16(1), 4–32 (1996). https://doi.org/10.1007/BF02086606
17. Liu, Y., Morgana, A., Simeone, B.: A linear algorithm for 2-bend embeddings of planar graphs in the two-dimensional grid. Discrete Appl. Math. **81**(1–3), 69–91 (1998). https://doi.org/10.1016/S0166-218X(97)00076-0
18. Rahman, M.S., Egi, N., Nishizeki, T.: No-bend orthogonal drawings of subdivisions of planar triconnected cubic graphs. IEICE. Transactions **88-D(1)**, 23–30 (2005)
19. Rahman, M.S., Nakano, S., Nishizeki, T.: A linear algorithm for bend-optimal orthogonal drawings of triconnected cubic plane graphs. J. Graph Algorithms Appl. **3**(4), 31–62 (1999). http://www.cs.brown.edu/publications/jgaa/accepted/99/SaidurNakanoNishizeki99.3.4.pdf
20. Rahman, M.S., Nishizeki, T.: Bend-minimum orthogonal drawings of plane 3-graphs. In: Goos, G., Hartmanis, J., van Leeuwen, J., Kučera, L. (eds.) WG 2002. LNCS, vol. 2573, pp. 367–378. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36379-3_32
21. Rahman, M.S., Nishizeki, T., Naznin, M.: Orthogonal drawings of plane graphs without bends. J. Graph Algorithms Appl. **7**(4), 335–362 (2003). http://jgaa.info/accepted/2003/Rahman+2003.7.4.pdf
22. Storer, J.A.: The node cost measure for embedding graphs on the planar grid (extended abstract). In: Miller, R.E., Ginsburg, S., Burkhard, W.A., Lipton, R.J. (eds.) Proceedings of the 12th Annual ACM Symposium on Theory of Computing, 28–30 April 1980, Los Angeles, California, USA, pp. 201–210. ACM (1980). https://doi.org/10.1145/800141.804667
23. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. SIAM J. Comput. 16(3), 421–444 (1987). https://doi.org/10.1137/0216030
24. Tamassia, R., Tollis, I.G., Vitter, J.S.: Lower bounds for planar orthogonal drawings of graphs. Inf. Process. Lett. **39**(1), 35–40 (1991). https://doi.org/10.1016/0020-0190(91)90059-Q
25. Thomassen, C.: Plane representations of graphs. In: Bondy, J., Murty, U. (eds.) Progress in Graph Theory, pp. 43–69 (1987)
26. Zhou, X., Nishizeki, T.: Orthogonal drawings of series-parallel graphs with minimum bends. SIAM J. Discrete Math. **22**(4), 1570–1604 (2008). https://doi.org/10.1137/060667621