

On the Complexity of HV-rectilinear Planarity Testing*

Walter Didimo¹, Giuseppe Liotta¹, and Maurizio Patrignani²

¹ Dept. of Engineering, University of Perugia, Italy

² Dept. of Engineering, Roma Tre University, Italy

Abstract. An HV-restricted planar graph G is a planar graph with vertex-degree at most four and such that each edge is labeled either H (horizontal) or V (vertical). The *HV-rectilinear planarity testing* problem asks whether G admits a planar drawing where every edge labeled V is drawn as a vertical segment and every edge labeled H is drawn as a horizontal segment. We prove that HV-rectilinear planarity testing is NP-complete even for graphs having vertex degree at most three, which solves an open problem posed by both Manuch *et al.* (GD 2010) and Durucher *et al.* (LATIN 2014). We also show that HV-rectilinear planarity can be tested in polynomial time for partial 2-trees of maximum degree four, which extends a previous result by Durucher *et al.* (LATIN 2014) about HV-restricted planarity testing of biconnected outerplanar graphs of maximum degree three. When the test is positive, our algorithm returns an orthogonal representation of G that satisfies the given H- and V-labels on the edges.

1 Introduction

Let $G = (V, E)$ be a planar graph with vertex-degree at most four. A *rectilinear orthogonal drawing* Γ of G is a planar drawing of G where each vertex $v \in V$ corresponds to a distinct point p_v of the plane and each edge $(u, v) \in E$ corresponds to a horizontal or vertical segment between p_u and p_v . If G is a planar embedded graph, i.e., a graph with a given planar embedding (a planar embedding defines for each vertex $v \in V$ the circular order of the edges incident to v and it also specifies the external face), we assume that a rectilinear orthogonal drawing of G preserves its embedding. If v is a vertex of a planar embedded graph and if e_1 and e_2 are two (possibly coincident) edges of 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* . Two rectilinear orthogonal drawings Γ and Γ' of the same planar embedded graph G are *shape equivalent* if for any angle a of G , the geometric angle corresponding to a is the same in Γ and Γ' . A *rectilinear orthogonal representation* \mathcal{H} of a planar embedded graph G is a class of shape equivalent rectilinear orthogonal drawings of G ; \mathcal{H} can be described by the embedding of G equipped with a label for each angle of G ; the labels for an angle can be R , F , L , or LL , corresponding to a geometric angle of 90, 180, 270, and 360 degrees, respectively.

The *rectilinear planarity testing* problem asks whether a planar graph G with vertex-degree at most four admits a rectilinear orthogonal drawing (or equivalently, a rectilinear orthogonal representation). The problem can be solved in polynomial time in the

* This research is supported in part by the Italian Ministry of Education, University, and Research (MIUR) under PRIN 2012C4E3KT national research project “AMANDA – Algorithms for MAssive and Networked DATA”

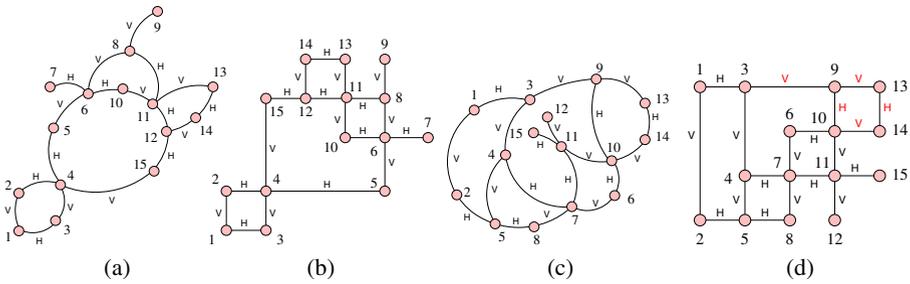


Fig. 1. (a) An HV-graph. (b) An HV-drawing of the HV-graph of Fig. 1(a). (c) An HV-graph that does not admit an HV-drawing. (d) A rectilinear orthogonal drawing of the graph of Fig. 1(c).

fixed embedding setting, i.e., when the rectilinear orthogonal representation must preserve a planar embedding of G given as part of the input [18]. The rectilinear planarity testing problem is however NP-complete in the *variable embedding setting*, i.e., over all planar embeddings of G [11]. Polynomial-time solutions exist if G is a biconnected series-parallel graph or if G has maximum vertex-degree three (see, e.g., [7,16,20]).

The rectilinear planarity testing problem is a classical subject of investigation in the graph drawing literature, and it can be regarded as a special case of the bend minimization problem for orthogonal drawings, probably one of the most explored topics in the field (see, e.g., [1,2,7,10,14,15,18]). Direction constrained versions of the rectilinear planarity testing problem also have a long tradition in the literature. An example is when every edge of G is labeled ‘Up’, or ‘Down’, or ‘Left’, or ‘Right’ and one wants to test whether G has a rectilinear drawing where every edge has a direction consistent with its label (see, e.g. [12,19]); the 3D version of this problem has also been studied [5,6,8].

In this paper we study another direction constrained version of the rectilinear planarity testing problem that has been receiving increasing interest. An *HV-restricted planar graph* (*HV-graph* for short) $G = (V, E)$ is a planar graph with vertex-degree at most four such that each edge is labeled either H (horizontal) or V (vertical). Denote by $E_H \subset E$ and $E_V \subseteq E$ the subsets of H- and V-labeled edges, respectively. An *HV-drawing* of G is a rectilinear orthogonal drawing of G such that each edge $e \in E_H$ (resp. $e \in E_V$) corresponds to an orthogonal (resp. vertical) segment. An *HV-realization* of G is a rectilinear orthogonal representation \mathcal{H} of G such that every drawing of \mathcal{H} is an HV-drawing of G , up to a rotation of 90 degrees. The *HV-rectilinear planarity testing* problem asks whether an HV-graph admits an HV-realization. Figure 1 (b) shows an HV-drawing of the HV-graph in Fig. 1 (a). Figure 1 (c) shows an HV-graph that does not admit HV-realizations, although it admits a rectilinear orthogonal drawing (Fig. 1 (d)).

Manuch *et al.* [13] ask what is the time complexity of HV-rectilinear planarity testing both in the fixed and in the variable embedding setting. Durocher *et al.* [9] describe a polynomial-time testing algorithm in the fixed embedding setting; for the variable embedding setting they present a quadratic-time testing algorithm for biconnected outerplanar graphs of vertex-degree at most three. The authors leave as open problem both to characterize the outerplanar HV-graphs that admit an HV-realization and to establish

the complexity of HV-rectilinear planarity testing in the variable embedding setting for general HV-graphs. We study these problems and establish the following results:

(i) HV-rectilinear planarity testing is NP-complete in the variable embedding setting even for HV-graphs with vertex-degree at most three. We recall, for a contrast, that Garg and Tamassia proved the NP-completeness of rectilinear planarity testing for planar graphs of vertex-degree at most four [11], but that rectilinear planarity testing can be solved in linear time for planar graphs of vertex-degree at most three [16].

(ii) There exists a polynomial-time algorithm to test HV-rectilinear planarity for HV-graphs that are partial 2-trees. In the affirmative case, the algorithm returns an HV-realization of the graph. Recall that biconnected outerplanar graphs are a sub-family of partial 2-trees, hence our result provides an algorithmic answer to the open problem of Durocher *et al.* [9], even for graphs with maximum vertex-degree four.

The remainder of the paper is organized as follows. Section 2 proves that HV-rectilinear planarity testing is NP-complete. Section 3 describes a polynomial-time algorithm for HV-rectilinear planarity testing of series-parallel graphs, which is used as a building-block for the design of an HV-rectilinear planarity testing algorithm for partial 2-trees presented in Sec. 4. Conclusions and open problems are in Sec. 5. For space reasons several proofs are sketched.

2 NP-completeness of HV-rectilinear Planarity Testing

It is easy to see that HV-rectilinear planarity testing is in NP, since the problem is polynomial when a planar embedding of the graph is given [9] and since all planar embeddings can be non-deterministically explored [1]. We show the hardness of this problem even on instances of maximum vertex-degree three by reducing SWITCH FLOW NETWORK to it. Hence, the following theorem holds.

Theorem 1. *HV-rectilinear planarity testing is NP-complete even for HV-graphs of maximum vertex-degree three.*

A *switch-flow network* \mathcal{N} is an undirected graph where each edge e is labeled with a range $[c' \dots c'']$ of nonnegative integers, called the *capacity range* of e . For simplicity, the capacity range $[c \dots c]$ is denoted with $[c]$. A *flow* for a switch-flow network is an orientation of its edges and an assignment of integer values to them. A flow is *feasible* if it satisfies the following two properties: (i) the total flow entering a vertex from the incoming edges is equal to the total flow exiting the vertex from the outgoing edges, and (ii) the flow assigned to an edge is an integer within the capacity range of the edge. Given a switch-flow network \mathcal{N} , the SWITCH FLOW NETWORK problem is the problem of finding a feasible flow for \mathcal{N} .

The SWITCH FLOW NETWORK problem is trivially in NP, by assigning to the edges all possible flow values and orientations and computing the sum of the flows at each vertex. In [11] it is shown that SWITCH FLOW NETWORK is NP-hard even in a very restrictive setting, that is, if its instances are such that: (a) the lower bounds of the capacities ranges of the edges are either zero (as in $[0 \dots c]$) or equal to the upper bounds (as in $[c]$), (b), edges with a proper capacity range (as in $[0 \dots c]$) do not form a cut, and

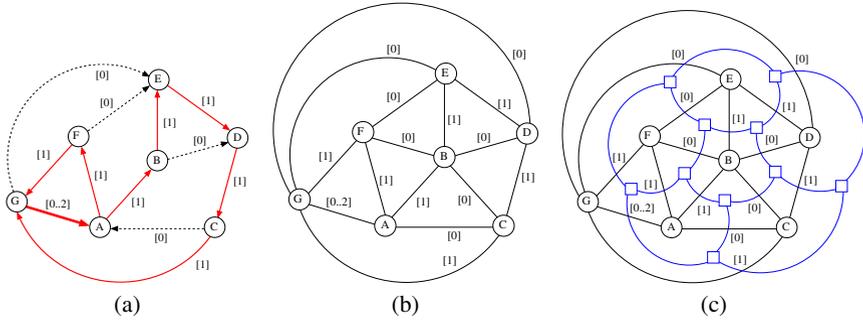


Fig. 2. (a) A feasible flow for an instance \mathcal{N} of the SWITCH FLOW NETWORK problem; the instance is the underlying undirected graph. The thick arrow represents a flow of two units while the thinner arrows represent flows of one unit. (b) The maximal planar SWITCH FLOW NETWORK instance \mathcal{N}^* obtained from \mathcal{N} . (b) Graph \mathcal{N}^* and its dual \mathcal{D} .

(c) the network is planar. Starting from an instance \mathcal{N} of the SWITCH FLOW NETWORK problem satisfying Properties (a), (b), and (c), we create an instance G of HV-planarity testing of maximum degree three as follows (refer to Figs 2 and 3):

Step 1. Construct a maximal planar instance \mathcal{N}^* by inserting dummy edges with capacity range $[0]$ into \mathcal{N} . Observe that \mathcal{N}^* admits a feasible flow if and only if \mathcal{N} does.

Step 2. Compute the dual plane graph D of \mathcal{N}^* . Observe that, since \mathcal{N}^* is a maximal triconnected graph, each vertex of D has degree three and D is also triconnected. We label the edges of D with the capacity range of the corresponding edge of \mathcal{N}^* .

Step 3. Compute an orthogonal drawing Γ_D of D with the linear-time algorithm in [17]. This algorithm takes as input a 4-plane biconnected graph and computes a drawing with at most $2n + 4$ bends and such that each edge has at least one vertical segment.

Step 4. Transform Γ_D into a positive instance F of HV-planarity testing by replacing orthogonal and vertical segments with rectangular boxes and by labeling each horizontal and vertical edge of F with labels H and V, respectively (see also Fig. 3(b)). Note that F has maximum vertex-degree three and, as D is triconnected, it has a unique HV-realization \mathcal{H}_F up to horizontal and vertical flips.

Step 5. Build the instance G of HV-planarity testing. First, identify for each edge e of D with a label different from $[0]$ a rectangular box of F corresponding to a vertical segment of Γ_D . If the label of e is $[c]$, insert the HV-graph T_c , called *tendrils*, in the rectangular box, attaching it with two edges e' and e'' called the *handles* of the tendrils. See Fig 3(d) and 3(e) for T_1 and T_2 . If e is labeled $[0\dots c]$ insert into the rectangular box the HV-graph W_c , called *wiggle*. Wiggles W_1 and W_2 are shown in Figs. 3(f) and 3(g).

Lemma 1. An HV-realization of the HV-graph G corresponds to a feasible flow of the switch network \mathcal{N} and vice versa.

Proof sketch: First, we show that, starting from an HV-realization \mathcal{H}_G of G , a feasible flow for \mathcal{N} can be found. Observe that each tendrils T_h necessarily has the HV-realization \mathcal{H}_{T_h} or $\overline{\mathcal{H}}_{T_h}$, giving to its left and right faces f_l and f_r , $4h$ and $-4h$ (or

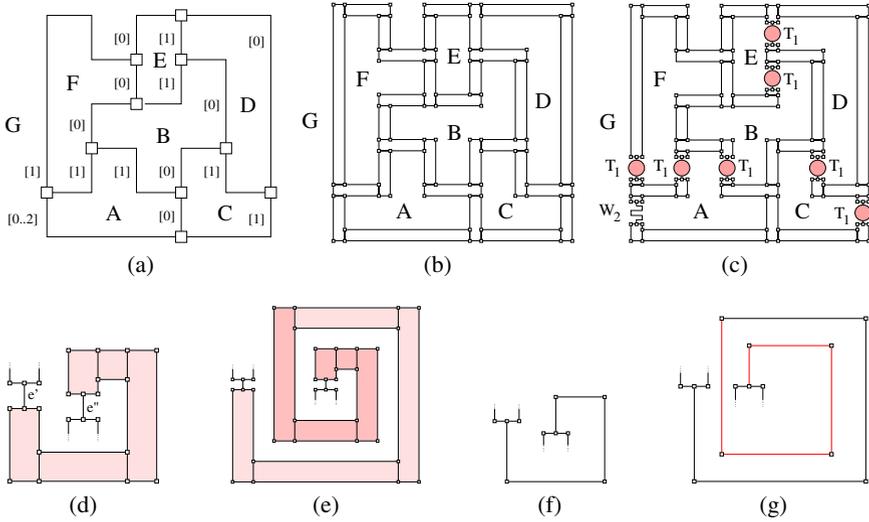


Fig. 3. (a) An orthogonal drawing Γ_D of the dual graph D . (b) The positive instance F of HV-planarity testing built by Step 4. (c) The instance G of HV-planarity testing corresponding to the SWITCH FLOW NETWORK instance \mathcal{N} depicted in Fig. 2(a). (d) Tendril T_1 . Edges e' and e'' are the handles of the tendril. (e) Tendril T_2 . (f) Wiggle W_1 . (g) Wiggle W_2 .

$-4h$ and $4h$, respectively) right angles. Since in any HV-realization of G , the subgraph of G obtained by neglecting tendrils and wiggles is drawn as it is in F , and since in F each face is balanced in terms of left and right turns, it follows that the HV-realization chosen for tendrils and wiggles of G decides a flow traversing the faces of G . By construction, such a flow (divided by 4) gives a feasible flow for \mathcal{N} . Conversely, suppose a feasible flow exists for \mathcal{N} . By choosing the corresponding HV-realizations for tendrils and wiggles, each face is balanced in terms of right and left angles, yielding an HV-realization for G . \square

3 Testing Algorithm for Series-Parallel Graphs

The decomposition of a biconnected graph G into its triconnected components is described by the *SPQR-tree* data structure, which implicitly represents all planar embeddings of G . We assume familiarity with *SPQR-trees* and related terminology [4]. If the *SPQR-tree* T of G has no *R-nodes* (corresponding to triconnected components that are triconnected graphs), G is a *series-parallel graph* and T is called an *SPQ-tree*: *S-nodes* represent series compositions, *P-nodes* parallel compositions, and *Q-nodes* single edges. Series-parallel graphs are a super-class of biconnected outerplanar graphs.

In [7] a polynomial-time algorithm is described that computes an orthogonal drawing of a series-parallel graph G , with the minimum number of bends over all planar embeddings of G . Our testing algorithm enhances the approach given in [7] to deal with H- and V-labels on the edges. As in [7] we use a variant of the *SPQ-trees* called *SPQ*-tree*.

The SPQ^* -tree T of G for a given reference edge e implicitly describes all possible planar embeddings of G with e on the external face and is such that: (i) each edge or a chain of edges joined by vertices of degree two in G is represented by a Q^* -node; (ii) the root of T is a P -node with two children, one of which is the Q^* -node associated with e ; (iii) each S -node has two children. An example of an SPQ^* -tree of a series-parallel graph is shown in Fig. 4(b). See [7] for more details. Another basic ingredient of the algorithm in [7] is the concept of *spirality* of an orthogonal representation \mathcal{H} ; it gives a measure of how much \mathcal{H} is “rolled-up”. We now recall basic definitions and results about spirality, restricted to rectilinear orthogonal representations, and we provide additional properties for the spirality of HV-realizations.

Spirality of Rectilinear Orthogonal Representations and HV Representations. Let T be the SPQ^* -tree of a series-parallel graph G for a given reference edge $e = (s, t)$. Let \mathcal{H} be a rectilinear orthogonal representation of G for some planar embedding of G with edge e on the external face. Also, let μ be a node of T with poles u and w , and let \mathcal{H}_μ be the restriction of \mathcal{H} to the pertinent graph G_μ of μ . We also say that \mathcal{H}_μ is a *component* of \mathcal{H} . For each pole $v \in \{u, w\}$, let $\text{indeg}_\mu(v)$ and $\text{outdeg}_\mu(v)$ be the degree of v inside and outside \mathcal{H}_μ , respectively. Define two (possibly coincident) *alias vertices* of v , denoted by v' and v'' , as follows: (a) If $\text{indeg}_\mu(v) = 1$, then $v' = v'' = v$; (b) If $\text{indeg}_\mu(v) > 1$ and $\text{outdeg}_\mu(v) = 1$, then $v' = v''$ is a dummy vertex that splits the edge of v that is outside \mathcal{H}_μ ; (c) If $\text{indeg}_\mu(v) = \text{outdeg}_\mu(v) = 2$, then v' and v'' are dummy vertices, each splitting a distinct edge of v that is outside \mathcal{H}_μ .

Let A^v denote the set of distinct alias vertices of a pole v (if G has vertex-degree at most three, $|A^v| = 1$). Let P^{uw} be any simple path from u to w inside \mathcal{H}_μ and let u' and w' be alias vertices of u and of w , respectively. The path $S^{u'w'}$ obtained concatenating (u', u) , P^{uw} , and (w, w') is called a *spine* of \mathcal{H}_μ . Denote by $n(S^{u'w'})$ the number of right turns minus the number of left turns encountered moving along $S^{u'w'}$ from u' to w' . The *spirality* $\sigma(\mathcal{H}_\mu)$ of a component \mathcal{H}_μ with poles u and w is defined based on the following cases: (i) If $A^u = \{u'\}$ and $A^w = \{w'\}$, then $\sigma(\mathcal{H}_\mu) = n(S^{u'w'})$. (ii) If $A^u = \{u'\}$ and $A^w = \{w', w''\}$, then $\sigma(\mathcal{H}_\mu) = (n(S^{u'w'}) + n(S^{u'w''}))/2$. (iii) If $A^u = \{u', u''\}$ and $A^w = \{w'\}$, then $\sigma(\mathcal{H}_\mu) = (n(S^{u'w'}) + n(S^{u''w'}))/2$. (iv) If $A^u = \{u', u''\}$ and $A^w = \{w', w''\}$, assume w.l.o.g. that (u, u') immediately precedes (u, u'') counterclockwise around u and that (w, w') immediately precedes (w, w'') clockwise around w . Then $\sigma(\mathcal{H}_\mu) = (n(S^{u'w'}) + n(S^{u''w''}))/2$.

We now briefly recall the spirality properties used in this paper, restricted to our setting. We assume that G is a series-parallel graph equipped with an SPQ^* -tree T .

Substituting Components with the Same Spirality. Let \mathcal{H} and \mathcal{H}' be two rectilinear orthogonal representations of the same planar graph G and let μ be a node of T . We say that \mathcal{H} and \mathcal{H}' are μ -*different* if their planar embeddings may differ only for the clockwise orderings of the edges inside G_μ and if the labels of the external angles at the poles of μ stay the same in the two representations. See Figs. 4(c) and 4(d) for an example. Suppose that \mathcal{H} and \mathcal{H}' are μ -different and that $\sigma(\mathcal{H}_\mu) = \sigma(\mathcal{H}'_\mu)$. The next theorem claims that it is always possible to get a new rectilinear orthogonal representation \mathcal{H}'' by *substituting* \mathcal{H}_μ with \mathcal{H}'_μ in \mathcal{H} (see, e.g., Fig. 4(e)).

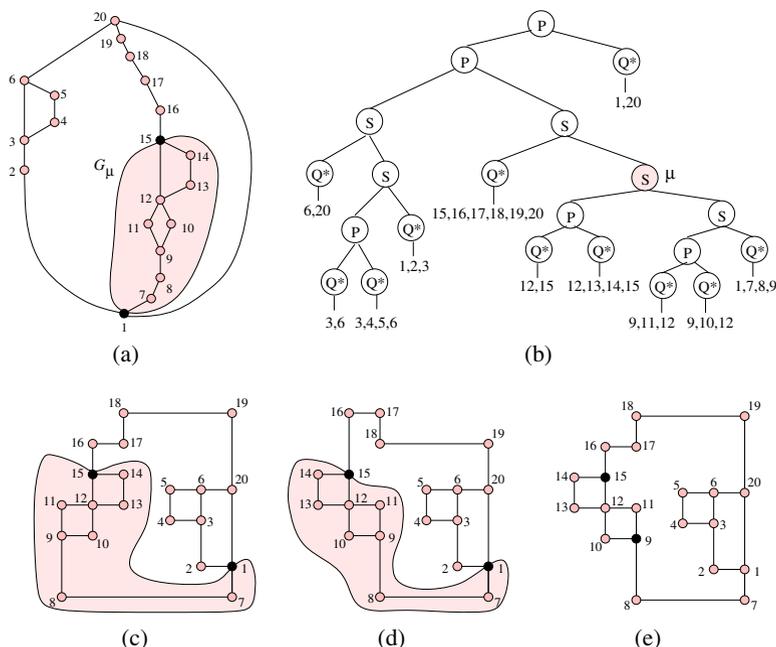


Fig. 4. (a) A planar graph G ; (b) The SPQ^* -tree of G for the reference edge $(1, 20)$, and a highlighted node μ (the pertinent graph G_μ of μ is highlighted in the graph); (c)-(d) Two rectilinear orthogonal representations \mathcal{H} and \mathcal{H}' that are μ -different; both \mathcal{H}_μ and \mathcal{H}'_μ have spirality 2. (e) The rectilinear orthogonal representation obtained by substituting \mathcal{H}_μ with \mathcal{H}'_μ in \mathcal{H} .

Theorem 2. [7] *Let \mathcal{H} and \mathcal{H}' be two rectilinear orthogonal representations of the same planar graph G such that \mathcal{H} and \mathcal{H}' are μ -different and $\sigma(\mathcal{H}_\mu) = \sigma(\mathcal{H}'_\mu)$. Then the embedded labeled graph \mathcal{H}'' obtained by substituting \mathcal{H}_μ with \mathcal{H}'_μ in \mathcal{H} is a rectilinear orthogonal representation.*

Intuitively, Theorem 2 implies that in an orthogonal representation \mathcal{H} we can always substitute a certain component \mathcal{H}_μ with a component \mathcal{H}'_μ having the same spirality, independently of the planar embedding of \mathcal{H}_μ and \mathcal{H}'_μ .

Spirality Relationships for Series- and Parallel-Compositions. Let \mathcal{H} be a rectilinear orthogonal representation of G . For an S - or a P -node μ of T , the spirality of \mathcal{H}_μ is related to the spiralities of the orthogonal representations of the pertinent graphs of the children of μ . For an S -node the relationship is given by Lemma 2. For a non-root P -node the relationship depends on the number of its children (see Lemmas 3 and 4).

Lemma 2. [7] *Let μ be an S -node of T with children μ_1 and μ_2 . Then: $\sigma(\mathcal{H}_\mu) = \sigma(\mathcal{H}_{\mu_1}) + \sigma(\mathcal{H}_{\mu_2})$.*

Lemma 3. [7] *Let μ be a P -node of T with three children μ_1, μ_2 , and μ_3 . Let e_i be the edge of \mathcal{H}_{μ_i} incident to pole w ($i = 1, 2, 3$) and let e_{out} be the external edge of \mathcal{H}_μ*

incident to w ; also, suppose that e_{out}, e_1, e_2, e_3 are encountered in this clockwise order around w . Then: $\sigma(\mathcal{H}_\mu) = \sigma(\mathcal{H}_{\mu_1}) + 2 = \sigma(\mathcal{H}_{\mu_2}) = \sigma(\mathcal{H}_{\mu_3}) - 2$.

Lemma 4. [7] *Let μ be a non-root P -node of T with two children μ_1, μ_2 . Suppose that while moving clockwise around w from an external edge of \mathcal{H}_μ incident to w , the edges of \mathcal{H}_{μ_1} incident to w precede those of \mathcal{H}_{μ_2} incident to w . For each pole $v \in \{u, w\}$ and for $i = 1, 2$, let e_{out} be the external edge of \mathcal{H}_μ incident to v that is circularly consecutive around v to an edge e_i of \mathcal{H}_{μ_i} incident to v , and let l_v^i be the label of the angle at v formed by e_{out} and e_i . Also, let $\alpha_v^i = 0$ if $l_v^i = F$ and $\alpha_v^i = 1$ if $l_v^i = R$. Then: (i) $\sigma(\mathcal{H}_\mu) = \sigma(\mathcal{H}_{\mu_1}) + k_u^i \alpha_u^1 + k_w^i \alpha_w^1$ and (ii) $\sigma(\mathcal{H}_\mu) = \sigma(\mathcal{H}_{\mu_2}) - k_u^i \alpha_u^2 - k_w^i \alpha_w^2$, where $k_v^i = 1$ if $\text{indeg}_{\mu_i}(v) = 1$ and $\text{outdeg}_\mu(v) = 1$, while $k_v^i = 1/2$ otherwise.*

Finally, if μ is the root node it has two children, one of which is associated with the reference edge e (see, e.g., Figs. 4(a) and 4(b)). The following result comes immediately from [7], considering that in a rectilinear orthogonal representation each edge (included e) is a segment, and the number of angles with R -labels minus the number of angles with L -labels in an internal face equals 4.

Lemma 5. *Let μ be the root of T and let ν be the child of μ that does not correspond to the reference edge (s, t) . Let f be the internal face of \mathcal{H} that contains (s, t) , and let l_s (resp. l_t) be the label of the angle in f at vertex s (resp. at vertex t). Assume that moving along e from s to t corresponds to walking counterclockwise on the boundary of f . For $v \in \{s, t\}$, let $\alpha_v = 0$ if $l_v = F$, $\alpha_v = 1$ if $l_v = R$, and $\alpha_v = -1$ if $l_v = L$. Then: $\sigma(\mathcal{H}_\nu) + k_s \alpha_s + k_t \alpha_t = 4$, where $k_v = 1$ if $\text{indeg}_\nu(v) = 1$ and $k_v = 0$ otherwise.*

Spirality and HV-realizations. Let G be an HV-graph and let T be an SPQ^* -tree of G for a reference edge $e = (s, t)$. W.l.o.g, from now on we assume that for any 2-degree vertex v the two edges incident to v have different labels (otherwise, they can be simply replaced by a single edge with the same label as the original edges). With this assumption, the chain associated with a Q^* -node always consists of an alternated sequence of H-labeled and V-labeled edges. The following lemma can be easily proved.

Lemma 6. *Let μ be a Q^* -node of T and let k be the number of edges of G_μ . If k is even (resp. odd), then there exists an HV-realization of G_μ with spirality j , for every odd (resp. even) $j \in [-k + 1, k - 1]$.*

Testing Algorithm. Let G be a series-parallel HV-graph and let T be the SPQ^* -tree for a given reference edge $e = (s, t)$. The testing is done by traversing T bottom-up; if the test fails, G does not admit an HV-realization with e on the external face, a new reference edge is chosen and the test is repeated. If the test succeed for some reference edge of G , the algorithm reconstructs an HV-realization of G with a top-down visit of T , exploiting suitable information stored in the nodes of T during the bottom-up traversal.

Theorem 3. *Let G be a series-parallel HV-graph with n vertices and maximum vertex-degree four. There exists an $O(n^4)$ -time algorithm that tests whether G admits an HV-realization and, if so, it computes an HV-realization of G . Also, if G has vertex-degree at most 3, the time-complexity can be reduced to $O(n^3 \log n)$.*

Proof sketch: Let T be the SPQ^* -tree for a given reference edge e . We describe an algorithm that tests whether G admits an HV-realization with e on the external face. Let μ the current node of T when traversing T from bottom to top. Let u and w be the poles of μ . The algorithm associates with μ a set of *tuples*, each one corresponding to a distinct value of spirality that an HV-realization \mathcal{H}_μ of G_μ can have within an HV-realization \mathcal{H} of G . The size of the set of tuples associated with μ is $O(n)$, since, by definition, the absolute value of spirality of \mathcal{H}_μ cannot exceed the length of the shortest path from u to w in G_μ plus one. A tuple contains a value of spirality σ_μ and an encoding of an embedding of G_μ for which an HV-realization of G_μ with spirality σ_μ exists; this encoding describes for a P -node the cyclic order of the edges of G_μ incident to u and w (i.e., the permutation of the edges of the skeleton of μ); if μ is either a Q^* -node or an S -node, the algorithm does not keep any embedding information for μ (because the embedding is uniquely defined). It is sufficient to keep only one representative tuple for each possible value of spirality, since by Theorem 2 rectilinear orthogonal representations with the same spirality are interchangeable.

If during the bottom-up visit we obtain an empty set of tuples for a node of T , the algorithm reports that G does not admit an HV-realization with e on the external face. Otherwise, let ν be the child of the root that does not correspond to edge e ; if the set of tuples of ν has a tuple whose spirality verifies the condition of Lemma 5, then the algorithm reports that an HV-realization of G exists; otherwise, it reports again that G does not have an HV-realization with e on the external face. Note that, the values α_s and α_t in Lemma 5 are uniquely determined by the H- and V-labels once the circular ordering of the edges around s and t is decided. If the algorithm fails on T , a different edge e is chosen and it is executed again. The tuples of Q^* -nodes are computed using Lemma 6, and those of non-root P -nodes using Lemmas 3 and 4. For an S -node μ with children μ_1 and μ_2 , based on Lemma 2, we consider all distinct values of spirality obtained by summing up the spiralities of a tuple of μ_1 and of a tuple of μ_2 . However, if μ_1 and μ_2 share a pole v of degree 4, the H- and V-labels on the edges incident to v may not be compatible with some pairs of spirality values for μ_1 and μ_2 , and these pairs must be discarded.

The tuple sets for all nodes of T are computed in $O(|S|n^2 + |P|n + |Q^*|n)$ time, where $|S|, |P|, |Q|$ denote the number of S -, P -, and Q^* -nodes in T , respectively. Hence, we have $O(n^3)$ -time complexity for a specific reference edge and $O(n^4)$ over all possible reference edges for G . If the test is positive the algorithm reconstructs an HV-realization of G in $O(n^2)$ time, by visiting T top-down.

If G has maximum vertex-degree 3, there cannot be forbidden pairs of spirality values for the children of an S -node, and finding its possible spiralities corresponds to computing a Cartesian sum of two sets of integers, which takes $O(n \log n)$ time [3]. Hence, the overall time complexity for series-parallel HV-graphs of vertex-degree at most 3 is $O(n(|S|(n \log n) + |P|n + |Q^*|n)) = O(n^3 \log n)$. \square

4 Testing Algorithm for Partial 2-Trees

We extend the HV-realizability testing algorithm described in the proof of Theorem 3 to simply connected graphs that do not contain rigid components. Namely, a 2-*tree* is a

graph obtained by starting from an edge and iteratively attaching a new vertex per time to two already adjacent vertices. A *partial 2-tree* is any subgraph of a 2-tree. We give a testing algorithm for the class of partial 2-trees, which includes series-parallel graphs.

Let G be an HV-graph that is a connected partial 2-tree (if G is not connected one can execute the test independently on each connected component). We assume that G does not contain three adjacent edges with the same label (H or V), because in this case it is trivial to conclude that G does not admit an HV-realization. The testing algorithm exploits a constrained version of the algorithm described in Sec. 3 for the biconnected components and the popular data structure known as the *block-cutvertex tree* \mathcal{T} of G , which describes the decomposition of G into its biconnected components, also called *blocks*. A block consisting of a single edge is called a *trivial block*. \mathcal{T} has a node v_B for each block B of G and a node v_c for each cutvertex c of G ; there is an arc (v_B, v_c) in \mathcal{T} if c belongs to B in G . A key-ingredient of our testing algorithm is the following; the proof is easy and is omitted for space reasons.

Lemma 7. *Let B_1 and B_2 be any two blocks of G and let Π be the path from v_{B_1} to v_{B_2} in \mathcal{T} . Let v_{c_1} and v_{c_2} be the cutvertex-nodes on Π adjacent to v_{B_1} and to v_{B_2} , respectively (c_1 and c_2 may coincide). In any planar embedding of G , either c_1 is on the external face of B_1 or c_2 is on the external face of B_2 .*

Let B be a block of G and c a cutvertex of G that belongs to B . Suppose we want to construct an HV-realization of G for a planar embedding where B has c on the external face and such that some other block is attached to c in the external face of B . To do this, we need the angle at c in the external face of B to be greater than 90 degrees. We say that B is *HV-extrovert with respect to c* if B admits an HV-realization \mathcal{H}_B such that: (i) c on the external face of \mathcal{H}_B ; (ii) the external angle at c is greater than 90 degrees. We also say that \mathcal{H}_B is *extrovert with respect to c* . The testing algorithm works as follows:

Step 1. Consider all degree-1 block-nodes of \mathcal{T} and for each of these nodes v_B let v_c be its adjacent cutvertex-node; test if B is HV-extrovert with respect to c (we explain how to test it right after the description of Step 2); if so, store its extrovert HV-realization in a list L and remove v_B from \mathcal{T} , otherwise mark B as *not HV-extrovert*. At the end of this step remove from \mathcal{T} all cutvertex-nodes of degree less than 2, previously attached to some degree-1 block-node.

Step 2. Check whether one of the following cases holds; if not repeat Step 1:

Case 1. Two blocks that are not HV-extrovert are found: in this case the test is negative, because the property of Lemma 7 cannot be satisfied in any HV-realization of G .

Case 2. \mathcal{T} becomes empty. The test is positive. Indeed, in this case there exists a planar embedding of G and every block B has an extrovert HV-realization, stored in L , that is compatible with this embedding; an HV-realization of G can be easily obtained by suitably merging the extrovert HV-realizations stored in L .

Case 3. \mathcal{T} consists of just one block-node v_B marked as not HV-extrovert. In this case the algorithm tests whether B admits any HV-realization \mathcal{H}_B , using the algorithm described in Sec. 3. In the affirmative case the test is positive and we can still construct an HV-realization of G by suitably merging the HV-realizations stored in L . Namely: (i) embed the HV-realization of each block that shares a cutvertex c with B inside a face

of G with angle at c larger than 90 degrees (such a face always exists, because c has degree at most 3 in B); (ii) merge the extrovert HV-realizations of the other blocks as in Case 2. If B does not admit any HV-realization, then the test is clearly negative.

We now explain how to test whether a block B is HV-extrovert with respect to a desired cutvertex c . If B is trivial (i.e., a single edge) the test is clearly positive. If B is not trivial, then c has degree greater than 2 in G and we distinguish between two cases:

Vertex c has Degree 3 in G . In this case c has degree 2 in B . Let s and u be the two vertices adjacent to c in B , and let T be the SPQ^* -tree of B with reference edge (s, c) . Execute an HV-realizability testing of B with respect to T , using the algorithm of Sec. 3. If the test is negative, B does not have an HV-realization with c on the external face, and we conclude that B is not HV-extrovert with respect to c . If the test is positive, there exists an HV-realization \mathcal{H}_B with c on the external face, but we have to verify that we can get an HV-realization with the external angle at c greater than 90 degrees. If edges (s, c) and (u, c) have the same label (H or V), we do not need to do any additional check, as \mathcal{H}_B surely has two angles of 180 degrees at c . If (s, c) and (u, c) have different labels, let ν be the child-node of the root of T that does not correspond to the reference edge; it suffices to check whether ν has a tuple whose spirality value satisfies the equation of Lemma 5, with the constraint that α_c (i.e., α_t of Lemma 5) is 1 (corresponding to an internal angle at c of 90 degrees, and hence to an external angle at c of 270 degrees).

Vertex c has degree 4 in G . If c has degree 2 in B , the algorithm applies the same check as in the previous case. If c has degree 3 in B , let $e = (s, c)$ and $e' = (u, c)$ be the edges of B incident to c with the same label (H or V), and let $e'' = (v, c)$ be the third edge of B incident to c . Let T be the SPQ^* -tree of B with reference edge $e = (s, c)$. Note that T has a P -node μ such that one of its poles is c and such that G_μ contains both e' and e'' . Also, in any HV-realization that is extrovert with respect to c , e and e' must be both on the external face. Hence, to check whether B is HV-extrovert with respect to c , we can execute an HV-realizability testing of B on tree T , using the algorithm of Sec. 3, with the restriction that when we compute the tuple set of μ we only consider the arrangement of its children corresponding to having e' on the external face.

About the computational complexity of the test described above, let n be the number of vertices of G , let B_1, B_2, \dots, B_h be the biconnected components of G , and let n_i be the number of vertices of B_i ($i = 1, \dots, h$). Also, let \mathcal{T} the block-cutvertex tree of G . For each block-node v_{B_i} of \mathcal{T} , the algorithm takes $O(n_i^3)$ to test whether B_i is HV-extrovert with respect to a desired cutvertex using the algorithm of Sec. 3, as it only needs to test the HV-realizability of B_i for a suitably chosen reference edge. Also, if one block B_i is not HV-extrovert, an additional HV-realizability testing on B_i is run over all possible reference edges, spending $O(n_i^4)$ time. Since $\sum_{i=1, \dots, h} n_i = O(n)$, the overall time complexity of the algorithm is $O(n^4)$. The time complexity is reduced to $O(n^3 \log n)$ if G has vertex-degree at most 3, with the same argument as in Theorem 3.

Theorem 4. *Let G be an HV-graph that is a partial 2-tree with n vertices and maximum-vertex degree four. There exists an $O(n^4)$ -time algorithm that tests whether G admits an HV-realization and, if so, it computes an HV-realization of G . Also, if G has vertex-degree at most 3, the time-complexity can be reduced to $O(n^3 \log n)$.*

5 Conclusions and Open Problems

We suggest the study of these problems: (i) Can the polynomial bound of Theorem 4 be improved? Recall that there is a linear-time algorithm for the rectilinear planarity testing of series-parallel graphs with vertex-degree at most three [20], but it relies on properties that do not apply for HV-graphs. (ii) Find a combinatorial characterization, e.g. in terms of forbidden substructures, for the HV-graphs that have an HV-realization.

References

1. Bertolazzi, P., Di Battista, G., Didimo, W.: Computing orthogonal drawings with the minimum number of bends. *IEEE Transactions on Computers* 49(8), 826–840 (2000)
2. Bläsius, T., Krug, M., Rutter, I., Wagner, D.: Orthogonal graph drawing with flexibility constraints. *Algorithmica* 68(4), 859–885 (2014)
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*. MIT Press (2009)
4. Di Battista, G., Tamassia, R.: On-line planarity testing. *SIAM J. on Comp.* 25, 956–997 (1996)
5. Di Battista, G., Kim, E., Liotta, G., Lubiw, A., Whitesides, S.: The shape of orthogonal cycles in three dimensions. *Discrete & Computational Geometry* 47(3), 461–491 (2012)
6. Di Battista, G., Liotta, G., Lubiw, A., Whitesides, S.: Embedding problems for paths with direction constrained edges. *Theoretical Computer Science* 289(2), 897–917 (2002)
7. Di Battista, G., Liotta, G., Vargiu, F.: Spirality and optimal orthogonal drawings. *SIAM J. on Comp.* 27(6), 1764–1811 (1998)
8. Di Giacomo, E., Liotta, G., Patrignani, M.: A note on 3D orthogonal drawings with direction constrained edges. *Inf. Proc. Lett.* 90(2), 97–101 (2004)
9. Durocher, S., Felsner, S., Mehrabi, S., Mondal, D.: Drawing HV-restricted planar graphs. In: Pardo, A., Viola, A. (eds.) *LATIN 2014*. LNCS, vol. 8392, pp. 156–167. Springer, Heidelberg (2014)
10. Felsner, S., Kaufmann, M., Valtr, P.: Bend-optimal orthogonal graph drawing in the general position model. *Computational Geometry* 47(3), 460–468 (2014)
11. Garg, A., Tamassia, R.: On the computational complexity of upward and rectilinear planarity testing. *SIAM J. on Comp.* 31(2), 601–625 (2001)
12. Hoffman, F.: Embedding rectilinear graphs in linear time. *Inf. Proc. Lett.* 29(2), 75–79 (1988)
13. Mañuch, J., Patterson, M., Poon, S.-H., Thachuk, C.: Complexity of finding non-planar rectilinear drawings of graphs. In: Brandes, U., Cornelsen, S. (eds.) *GD 2010*. LNCS, vol. 6502, pp. 305–316. Springer, Heidelberg (2011)
14. Mutzel, P., Weiskircher, R.: Bend minimization in planar orthogonal drawings using integer programming. *SIAM J. on Opt.* 17(3), 665–687 (2006)
15. Rahman, M.S., Nakano, S.I., Nishizeki, T.: A linear algorithm for bend-optimal orthogonal drawings of triconnected cubic plane graphs. *J. of Graph Alg. and Appl.* 3(4), 31–62 (1999)
16. Rahman, M.S., Nishizeki, T., Naznin, M.: Orthogonal drawings of plane graphs without bends. *J. of Graph Alg. and Appl.* 7(4), 335–362 (2003)
17. Tamassia, R., Tollis, I.G.: Planar grid embedding in linear time. *IEEE Transactions on Circuits Systems CAS-36*(9), 1230–1234 (1989)
18. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. *SIAM J. on Comp.* 16(3), 421–444 (1987)
19. Vijayan, G., Wigderson, A.: Rectilinear graphs and their embeddings. *SIAM J. on Comp.* 14(2), 355–372 (1985)
20. Zhou, X., Nishizeki, T.: Orthogonal drawings of series-parallel graphs with minimum bends. *SIAM J. on Discr. Math.* 22(4), 1570–1604 (2008)