



Planar Rectilinear Drawings of Outerplanar Graphs in Linear Time

Fabrizio Frati^(✉)

Roma Tre University, Rome, Italy
frati@dia.uniroma3.it

Abstract. We show how to test in linear time whether an outerplanar graph admits a planar rectilinear drawing, both if the graph has a prescribed plane embedding and if it does not. Our algorithm returns a planar rectilinear drawing if the graph admits one.

1 Introduction

Planar orthogonal graph drawings with a minimum number of bends have been studied for decades. In 1987, Tamassia [20] proved that, for an n -vertex planar graph with a prescribed plane embedding, a planar orthogonal drawing with the minimum number of bends can be constructed in polynomial time, thereby establishing a result that lies at the very foundations of the graph drawing research area. The running time of Tamassia's algorithm is $O(n^2 \log n)$, which has been improved to $O(n^{7/4} \sqrt{\log n})$ [11] and then to $O(n^{3/2})$ by Cornelsen and Karrenbauer [4]. However, achieving a linear running time is still an elusive goal.

Bend minimization in the variable embedding setting is a much harder problem; indeed, Garg and Tamassia [12] proved that testing whether a graph admits a planar orthogonal drawing with zero bends is NP-hard. However, some natural restrictions on the input make the problem tractable. A successful story is the one about n -vertex degree-3 planar graphs. Di Battista et al. [5] proved that, for such graphs, a planar orthogonal drawing with the minimum number of bends can be constructed in $O(n^5 \log n)$ time. After some improvements [3, 9], a recent breakthrough result by Didimo et al. [8] has shown that $O(n)$ time is indeed sufficient. Di Battista et al. [5] also presented an $O(n^4)$ -time algorithm for minimizing the number of bends in a planar orthogonal drawing of an n -vertex biconnected series-parallel graph. This result was first extended to not necessarily biconnected series-parallel graphs by Bläsius et al. [1] and then improved to an $O(n^3 \log n)$ running time by Di Giacomo et al. [6].

Evidence has shown that the bend-minimization problem is not much easier if one is only interested in the construction of planar orthogonal drawings with zero bends; these are also called *planar rectilinear drawings* (see Figs. 1(a)

Partially supported by MIUR Project “AHeAD” under PRIN 20174LF3T8, by H2020-MSCA-RISE Proj. “CONNECT” n° 734922, and by Roma Tre University Azione 4 Project “GeoView”.

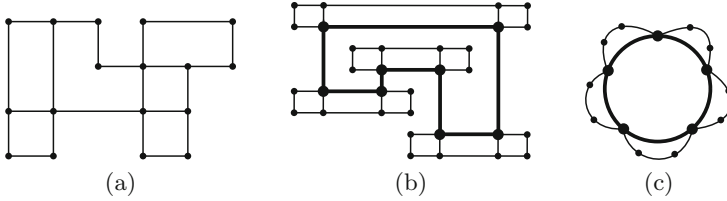


Fig. 1. (a) An outerplanar rectilinear drawing. (b) A planar rectilinear drawing of an outerplanar graph G . The graph G has no outerplanar rectilinear drawing. (c) An outerplanar graph G that has no 3-cycle and no planar rectilinear drawing.

and 1(b) for two such drawings). Namely, the cited NP-hardness proof of Garg and Tamassia [12] is designed for planar rectilinear drawings. Further, almost every efficient algorithm for testing the existence of planar rectilinear drawings [14, 17, 19] has been eventually subsumed by an algorithm in the more general bend-minimization scenario [8, 18, 21]. A notable exception is that of planar graphs with a fixed embedding, for which the fastest known algorithms for the bend-minimization problem and for the rectilinear-planarity testing problem run in $O(n^{3/2})$ time [4] and in $O(n \log^3 n)$ time [2, 20], respectively.

In this paper, we show that the existence of a planar rectilinear drawing can be tested in $O(n)$ time for an n -vertex outerplanar graph. Our algorithm is constructive and covers both the fixed and the variable embedding scenarios, where the previously best known time bounds were $O(n \log^3 n)$ [2, 20] and $O(n^3 \log n)$ [6], respectively; our algorithm also allows us to test in $O(n)$ time whether an n -vertex outerplanar graph admits an outerplanar rectilinear drawing. Given how common it is to study outerplanar graphs for a problem which is too difficult or too computationally expensive on general planar graphs, it is surprising that a systematic study of planar orthogonal and rectilinear drawings of outerplanar graphs has not been done before. The only result we are aware of that is tailored for outerplanar graphs is the one by Nomura et al. [16], which states that an outerplanar graph with maximum degree 3 admits a planar rectilinear drawing if and only if it does not contain any 3-cycle. This characterization is not true for outerplanar graphs with vertices of degree 4; see, e.g., Fig. 1(c).

We outline our algorithm for the variable embedding setting.

The first, natural, idea is to reduce the problem to the 2-connected case. This reduction builds on (an involved version of) a technique introduced by Didimo et al. [8] that, roughly speaking, allows us to perform postorder traversals of the block-cut-vertex tree of the graph in total linear time so that each edge is traversed in both directions; during these traversals, information is computed that allows us to decide whether solutions for the subproblems associated to the blocks of the graph can be combined into a solution for the entire graph. This reduction to the 2-connected case comes at the expense of having to solve a harder problem, in which some vertices of the graph have restrictions on their incident angles in the sought planar rectilinear drawing.

An analogous technique allows us to reduce the problem to the case in which the input 2-connected graph has a prescribed edge that is required to be incident to the outer face. The role that in the previous reduction is played by the block-cut-vertex tree is here undertaken by the “extended dual tree” of the outerplanar graph. Each edge of this tree is dual to an edge of the outerplanar graph; the latter edge splits the outerplanar graph into two smaller outerplanar graphs. These are the sub-instances whose solutions might be combined into a solution for the entire graph; whether this combination is possible is decided based on information that is computed during the traversals of the extended dual tree.

The core of our algorithm consists of an efficient solution for the problem of testing whether a 2-connected outerplanar graph admits a planar rectilinear drawing in which a prescribed edge is required to be incident to the outer face. Our starting point is a characterization of the positive instances in terms of the existence of a sequence of numerical values satisfying some conditions; these values represent certain geometric angles of a planar rectilinear drawing. Some of these numerical values can be chosen optimally, based on recursive solutions to smaller subproblems; further, a constant number of them have to be chosen in all possible ways; finally, we reduce the problem of finding the remaining numerical values to the one of testing for the existence of a set of integers, each of which is required to be in a certain interval, so that a linear equation on these integers is satisfied. We characterize the solutions to the latter problem so that not only it can be solved efficiently, but a solution can be modified in constant time if the interval associated to each integer changes slightly; this change corresponds to a different edge chosen to be incident to the outer face.

Together with our submission to GD 2020, another paper on the rectilinear-planarity testing problem was accepted to the same conference. Namely, Didimo et al. [7] presented an $O(n)$ -time algorithm which tests whether an n -vertex 2-connected series-parallel graph with fixed embedding admits a planar rectilinear drawing; the techniques by Didimo et al. are different from ours.

In what follows, we assume w.l.o.g. that every considered graph is connected and has maximum degree 4. Because of space limitations, we only present our algorithm for outerplanar graphs with a variable embedding. We also neglect the construction of planar rectilinear drawings and focus on testing their existence. Finally, all proofs are omitted and deferred to the full version of the paper [10].

2 Preliminaries

A *block* of a connected graph G is a maximal 2-connected subgraph of G ; it is *trivial* if it is a single edge and *non-trivial* otherwise. The *BC-tree* T of G [13, 15] is the tree that has a *B-node* for each block of G and a *C-node* for each cut-vertex of G ; a B-node b and a C-node c are adjacent in T if c lies in the block corresponding to b (we often identify a C-node and the corresponding cut-vertex).

A *drawing* of a graph maps each vertex to a point in the plane and each edge to a curve between its endpoints. A drawing is *planar* if no two edges cross and it is *rectilinear* if each edge is either a horizontal or a vertical segment. A planar

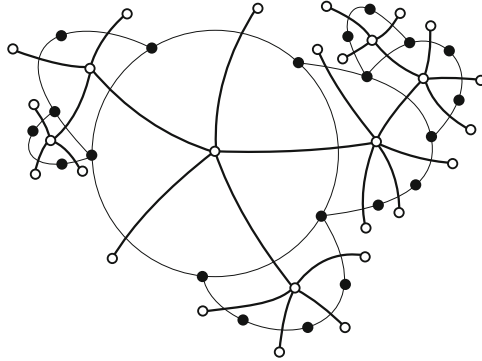


Fig. 2. The extended dual tree (represented by white disks and thick lines) of an outerplane embedding of a 2-connected outerplanar graph.

drawing divides the plane into topologically connected regions, called *faces*; the only unbounded face is the *outer face*, while all the other faces are *internal*.

Two planar drawings Γ_1 and Γ_2 of a connected planar graph G are *equivalent* if: (i) for each vertex w of G , the clockwise order of the edges incident to w is the same in Γ_1 and Γ_2 ; and (ii) the clockwise order of the edges incident to the outer face is the same in Γ_1 and Γ_2 . A *plane embedding* is an equivalence class of planar drawings. Two drawings that correspond to the same plane embedding have faces delimited by the same walks, hence we often speak about *faces of a plane embedding*. We denote by $f_{\mathcal{E}}^*$ the outer face of a plane embedding \mathcal{E} .

Two planar rectilinear drawings of a 2-connected planar graph G are *equivalent* if they correspond to the same plane embedding \mathcal{E} and if, for every face f of \mathcal{E} and for every vertex w incident to f , the angle at w in f is the same in both drawings. A *rectilinear representation* of G is an equivalence class of planar rectilinear drawings of G . A rectilinear representation is hence a pair (\mathcal{E}, ϕ) , where \mathcal{E} is a plane embedding of G and ϕ is a function that assigns an *angle* $\phi(w, f) \in \{90^\circ, 180^\circ, 270^\circ\}$ to every pair (w, f) such that w is a vertex incident to a face f of \mathcal{E} . For a rectilinear representation (\mathcal{E}, ϕ) and a vertex u incident to $f_{\mathcal{E}}^*$, we denote by $\phi^{\text{int}}(u)$ the sum of the *internal angles* incident to u , that is, $\phi^{\text{int}}(u) = \sum_f \phi(u, f)$, where the sum is over all the internal faces f of \mathcal{E} incident to u . For planar graphs that are not 2-connected, the notions of equivalence between planar rectilinear drawings and of rectilinear representation are similar, however a vertex w might have several occurrences w^1, \dots, w^x on the boundary of a face f , hence ϕ assigns an angle to every pair (w^k, f) , for $k \in \{1, \dots, x\}$; further, the value 360° is admissible for $\phi(w^k, f)$.

An *outerplanar drawing* is a planar drawing such that all the vertices are incident to the outer face. An *outerplane embedding* is an equivalence class of outerplanar drawings. A graph is *outerplanar* if it admits an outerplanar drawing. The *extended dual tree* \mathcal{T} of an outerplane embedding \mathcal{O} of an n -vertex 2-connected outerplanar graph is obtained from the dual graph of \mathcal{O} by replacing the vertex corresponding to $f_{\mathcal{O}}^*$ with n new degree-1 nodes. See Fig. 2.

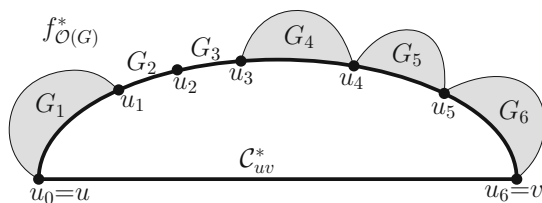


Fig. 3. A 2-connected outerplanar graph G rooted at uv . The uv -subgraphs of G are colored gray; G_2 and G_3 are trivial, while $G_1, G_4, G_5,$ and G_6 are not.

3 Testing Algorithm for Outerplanar Graphs

In this section, we show how to test in $O(n)$ time whether an n -vertex outerplanar graph G with a variable embedding admits a planar rectilinear drawing. In Sect. 3.1, we assume that G is 2-connected and that an edge of G is prescribed to be incident to the outer face. In Sect. 3.2, we get rid of the second assumption. In Sect. 3.3, we get rid of the first assumption.

3.1 2-Connected Outerplanar Graphs with a Prescribed Edge

Let G be an n -vertex 2-connected outerplanar graph, let \mathcal{O} be its outerplane embedding, let uv be an edge incident to $f_{\mathcal{O}}^*$, and let χ be a set of degree-2 vertices of G . A χ -constrained representation of G is a rectilinear representation (\mathcal{E}, ϕ) of G such that, for every vertex $x \in \chi$ and every face f of \mathcal{E} incident to x , we have $\phi(x, f) \in \{90^\circ, 270^\circ\}$. We now show how to test in $O(n)$ time whether G admits a χ -constrained representation (\mathcal{E}, ϕ) in which uv is incident to $f_{\mathcal{E}}^*$.

The reason for introducing this seemingly artificial problem is the following. Consider an outerplanar graph H and assume that H contains a degree-4 cut-vertex v that belongs to two non-trivial blocks H_{b_1} and H_{b_2} of H . For $i = 1, 2$, in any plane embedding of H , the two edges of H_{b_i} incident to v are consecutive in clockwise order around v . Hence, in the restriction of a rectilinear representation of H to H_{b_i} , the angles incident to v are 90° and 270° . In Sect. 3.3, we will use the algorithm that tests whether a block H_{b_i} of a simply-connected outerplanar graph H admits a χ -constrained representation as one of the main ingredients for testing whether H admits a rectilinear representation.

We say that uv is the root of G . Let $C_{uv}^* = (u = u_0, u_1, \dots, u_k = v)$ be the cycle delimiting the internal face of \mathcal{O} incident to uv ; see Fig. 3. The blocks of the graph obtained from G by removing the edge uv are the uv -subgraphs of G . These are denoted by G_1, \dots, G_k , where the root of G_i is $u_{i-1}u_i$. The assumption that uv is incident to the outer face of the desired plane embedding \mathcal{E} ensures that C_{uv}^* lies in the outer face of each uv -subgraph G_i of G in \mathcal{E} . Conversely, each uv -subgraph G_i of G might lie inside or outside C_{uv}^* in \mathcal{E} .

A χ -constrained representation (\mathcal{E}, ϕ) in which the root uv is incident to $f_{\mathcal{E}}^*$ and the angles $\phi^{\text{int}}(u)$ and $\phi^{\text{int}}(v)$ are equal to μ and ν , respectively, is called a (χ, μ, ν) -representation of G . We show how to test, for any $\mu, \nu \in$

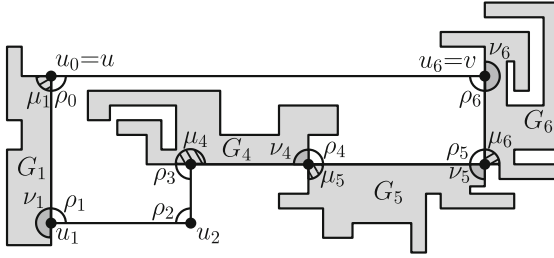


Fig. 4. A planar rectilinear drawing of the outerplanar graph G in Fig. 3.

$\{90^\circ, 180^\circ, 270^\circ\}$, whether G admits a (χ, μ, ν) -representation. The following lemma is one of the main ingredients for our algorithm. For $i = 1, \dots, k$, let $\chi_i := \chi \cap V(G_i)$. An *in-out assignment* is an assignment of each non-trivial uv -subgraph G_i of G either to the inside or to the outside of C_{uv}^* . Refer to Fig. 4.

Lemma 1. *For any $\mu, \nu \in \{90^\circ, 180^\circ, 270^\circ\}$, we have that G admits a (χ, μ, ν) -representation if and only if there exist an in-out assignment \mathcal{A} and a sequence of values $\rho_0, \rho_1, \mu_1, \nu_1, \rho_2, \mu_2, \nu_2, \dots, \rho_k, \mu_k, \nu_k$ in $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ so that the following properties are satisfied.*

- (V1): for $i = 0, \dots, k$, we have $\rho_i \geq 90^\circ$; further, if $u_i \in \chi$, then either $\rho_i = 90^\circ$ or $\rho_i = 270^\circ$;
- (V2): for $i = 1, \dots, k$, if G_i is trivial then $\mu_i = \nu_i = 0^\circ$, otherwise $\mu_i, \nu_i \in \{90^\circ, 180^\circ\}$ and G_i admits a (χ_i, μ_i, ν_i) -representation;
- (V3): for $i = 1, \dots, k - 1$, we have that $\nu_i + \rho_i + \mu_{i+1} \leq 270^\circ$;
- (V4): $\rho_0 + \mu_1 = \mu$ and $\rho_k + \nu_k = \nu$; and
- (V5): for $i = 1, \dots, k$, if G_i is trivial or is assigned by \mathcal{A} to the outside of C_{uv}^* , then $\sigma_i = 0^\circ$, else $\sigma_i = \mu_i + \nu_i$; then we have $\sum_{i=0}^k \rho_i + \sum_{i=1}^k \sigma_i = (k-1) \cdot 180^\circ$.

Let (\mathcal{E}, ϕ) be a rectilinear representation of G and let $f_{\mathcal{E}}^{uv}$ be the internal face of \mathcal{E} incident to uv . Then, roughly speaking, ρ_i represents $\phi(u_i, f_{\mathcal{E}}^{uv})$; further, if G_i is non-trivial, then μ_i and ν_i represent the sums of the internal angles incident to u_{i-1} and u_i , respectively, in the restriction of (\mathcal{E}, ϕ) to G_i .

Due to Lemma 1, our goal becomes that of testing for the existence of a sequence $\rho_0, \rho_1, \mu_1, \nu_1, \dots, \rho_k, \mu_k, \nu_k$ and of an in-out assignment \mathcal{A} such that Properties (V1)–(V5) are satisfied. Property (V2) implies that, for every trivial uv -subgraph G_i of G , the values μ_i and ν_i can be set equal to 0° without loss of generality. The values μ_i and ν_i can also be chosen “optimally” for every non-trivial uv -subgraph G_i of G , except for G_1 and G_k . This choice selects one of the pairs (μ_i, ν_i) with $\mu_i, \nu_i \in \{90^\circ, 180^\circ\}$ such that G_i admits a (χ_i, μ_i, ν_i) -representation, as required by Property (V2), and has to guarantee that Property (V3) is not violated. Subject to these constraints, the *optimal pair* (μ_i, ν_i) for G_i is the one for which $\mu_i + \nu_i$ is minimum. Choosing the optimal pair for G_i only requires to check information associated to G_{i-1} , G_i , and G_{i+1} , hence it can be done in $O(1)$ time, and thus in $O(k)$ time for all graphs G_2, \dots, G_{k-1} .

If the *optimal sequence* of values $\mu_2, \nu_2, \mu_3, \nu_3, \dots, \mu_{k-1}, \nu_{k-1}$ was established (otherwise we conclude that G admits no (χ, μ, ν) -representation), then we consider all the $3^6 \in O(1)$ tuples $(\mu_1, \nu_1, \mu_k, \nu_k, \rho_0, \rho_k)$ such that $\mu_1, \nu_1, \mu_k, \nu_k \in \{0^\circ, 90^\circ, 180^\circ\}$ and $\rho_0, \rho_k \in \{90^\circ, 180^\circ, 270^\circ\}$. For each of the tuples, we test in $O(1)$ time whether it violates Properties (V1)–(V4), and in the positive case we discard the tuple. If we discarded all the tuples $(\mu_1, \nu_1, \mu_k, \nu_k, \rho_0, \rho_k)$, we conclude that G admits no (χ, μ, ν) -representation. Every tuple that was not discarded, together with the optimal sequence, forms a *promising sequence*. We process each promising sequence independently and check whether it is *extensible*, i.e., whether there exist an in-out assignment \mathcal{A} and values $\rho_1, \rho_2, \dots, \rho_{k-1}$ that, together with the promising sequence, satisfy Properties (V1)–(V5).

In particular, the choice of \mathcal{A} and $\rho_1, \rho_2, \dots, \rho_{k-1}$ is done so as to satisfy Property (V5), i.e., $\sum_{i=0}^k \rho_i + \sum_{i=1}^k \sigma_i = (k-1) \cdot 180^\circ$, while complying with Properties (V1) and (V3). For example, if a vertex u_i belongs to χ , then we need to set either $\rho_i = 90^\circ$ or $\rho_i = 270^\circ$. Recall that the sum $\mu_i + \nu_i$ is now fixed, for $i = 1, \dots, k$. Suppose, for example, that $\mu_i + \nu_i = 360^\circ$. If \mathcal{A} assigns G_i to the inside of \mathcal{C}_{uv}^* , this contributes 360° to the sum $\sum_{i=1}^k \sigma_i$, otherwise it contributes 0° . Let a be the number of uv -subgraphs for which $\mu_i + \nu_i = 360^\circ$; then \mathcal{A} can assign any $0 \leq a' \leq a$ of such graphs to the inside of \mathcal{C}_{uv}^* , and this will contribute $a' \cdot 360^\circ$ to the sum $\sum_{i=1}^k \sigma_i$. Similar considerations allow us to reduce the problem of testing for the extensibility of a promising sequence to the problem of deciding whether integer values $0 \leq a' \leq a, 0 \leq b' \leq b, 0 \leq c' \leq c$, and $0 \leq d' \leq d$ exist such that $4a' + 3b' + 2c' + d' = t$, where a, b, c, d , and t are given integers inherent to the values of the promising sequence; the values a, b, c, d , and t can be computed in $O(k)$ time from the promising sequence.

If t is “very small” or “very large”, then the existence of the values a', b', c' , and d' can be decided in $O(1)$ time by means of exhaustive search. For values of t that are neither too small nor too large, we characterize the positive instances a, b, c, d, t as the ones satisfying a certain (constant size) boolean formula. For example, if $d \geq 3$, a solution to $4a' + 3b' + 2c' + d' = t$ subject to $0 \leq a' \leq a, 0 \leq b' \leq b, 0 \leq c' \leq c$, and $0 \leq d' \leq d$ always exists, while if $d = 2$ it exists if and only if $b > 0$, or $c > 0$, or $t \not\equiv 3 \pmod 4$.

This concludes the description of our algorithm to test whether G admits a (χ, μ, ν) -representation. The algorithm runs in $O(k)$ time, assuming that the pairs (μ_i, ν_i) with $\mu_i, \nu_i \in \{90^\circ, 180^\circ\}$ such that each uv -subgraph G_i of G admits a (χ_i, μ_i, ν_i) -representation are known. In particular, once the values $\mu_1, \nu_1, \dots, \mu_k, \nu_k, \rho_0, \rho_k, a, b, c, d, t$ have been computed, the algorithm concludes the test in $O(1)$ time (for each promising sequence).

We now use this algorithm inductively on the subgraphs of G . Namely, we root the extended dual tree \mathcal{T} of the outerplane embedding of G at the leaf r^* such that the edge of \mathcal{T} incident to r^* is dual to uv . Then, for any internal node s of \mathcal{T} , let G_s be the subgraph of G dual to the subtree of \mathcal{T} rooted at s , let $\chi_s = \chi \cap V(G_s)$, and let s_1, \dots, s_k be the children of s in \mathcal{T} . We use the described algorithm to determine in $O(k)$ time the pairs (μ, ν) with $\mu, \nu \in \{90^\circ, 180^\circ, 270^\circ\}$ such that G_s admits a (χ_s, μ, ν) -representation, starting from the pairs (μ_i, ν_i)

with $\mu_i, \nu_i \in \{0^\circ, 90^\circ, 180^\circ\}$ such that G_{s_i} admits a $(\chi_{s_i}, \mu_i, \nu_i)$ -representation, for $i = 1, \dots, k$. This results in the following.

Theorem 1. *Let G be an n -vertex 2-connected outerplanar graph, uv be an edge incident to the outer face of the outerplane embedding of G , and χ be a subset of the degree-2 vertices of G . There is an $O(n)$ -time algorithm which tests, for any values $\mu, \nu \in \{90^\circ, 180^\circ, 270^\circ\}$, whether G admits a (χ, μ, ν) -representation.*

By independently considering all the pairs (μ, ν) with $\mu, \nu \in \{90^\circ, 180^\circ, 270^\circ\}$, Theorem 1 also allows us to test whether a χ -constrained representation of G exists such that uv is incident to the outer face.

3.2 2-Connected Outerplanar Graphs

We now get rid of the assumption that there is a prescribed edge uv incident to the outer face of the rectilinear representation we seek, while maintaining the assumption that the input n -vertex outerplanar graph G is 2-connected. We are again required to look for χ -constrained representations.

Our $O(n)$ -time algorithm to solve this problem will actually perform a more general task. Namely, our algorithm will label every vertex u of G whose degree is not larger than 3 with a set $\gamma(u)$ which contains all the values $\mu \in \{90^\circ, 180^\circ, 270^\circ\}$ such that G admits a χ -constrained representation (\mathcal{E}, ϕ) in which u is incident to $f_{\mathcal{E}}^*$ and $\phi^{\text{int}}(u) = \mu$.

First, we reduce the problem of computing the labels $\gamma(u)$ to the problem of computing labels $\mathcal{N}_{s \rightarrow t}$ and $\mathcal{N}_{t \rightarrow s}$ for each edge st of the extended dual tree \mathcal{T} of the outerplane embedding \mathcal{O} of G . The label $\mathcal{N}_{s \rightarrow t}$ is defined as follows (the label $\mathcal{N}_{t \rightarrow s}$ is defined symmetrically). Refer to Fig. 5. Let uv be the edge of G dual to st . The removal of the edge st splits \mathcal{T} into two trees. Let $\mathcal{T}_{s \rightarrow t}$ be the one containing s . If s is an internal node of \mathcal{T} , then let $G_{s \rightarrow t}$ be the graph $\bigcup_{x \in \mathcal{T}_{s \rightarrow t}} \mathcal{C}_x$, where \mathcal{C}_x is the cycle of G delimiting the face of \mathcal{O} dual to the node x of \mathcal{T} . If s is a leaf of \mathcal{T} , then let $G_{s \rightarrow t}$ be the edge uv . In both cases, $G_{s \rightarrow t}$ is rooted at uv . Let $\chi_{s \rightarrow t} = \chi \cap V(G_{s \rightarrow t})$. If s is an internal node of \mathcal{T} , then $\mathcal{N}_{s \rightarrow t}$ contains all the pairs (μ, ν) with $\mu, \nu \in \{90^\circ, 180^\circ, 270^\circ\}$ such that $G_{s \rightarrow t}$ admits a $(\chi_{s \rightarrow t}, \mu, \nu)$ -representation, while if s is a leaf, then $\mathcal{N}_{s \rightarrow t} = \{(0^\circ, 0^\circ)\}$.

The sets $\gamma(u)$ can be easily recovered from the labels $\mathcal{N}_{s \rightarrow t}$. Let $u_1 v_1, \dots, u_n v_n$ be the edges of G incident to $f_{\mathcal{O}}^*$, in any order. The labels $\mathcal{N}_{s \rightarrow t}$ are computed by means of n postorder traversals of \mathcal{T} ; during the h -th traversal, \mathcal{T} is rooted at the leaf r_h^* such that the edge $r_h^* r_h$ incident to r_h^* is dual to $u_h v_h$.

When processing a node s with parent t during one of the traversals, we compute the label $\mathcal{N}_{s \rightarrow t}$. The computation of $\mathcal{N}_{s \rightarrow t}$ exploits the values of the already computed labels $\mathcal{N}_{s_1 \rightarrow s}, \dots, \mathcal{N}_{s_k \rightarrow s}$, where s_1, \dots, s_k are the neighbors of s in \mathcal{T} different from t . This is the problem we solved in Sect. 3.1! Namely, we want to compute the pairs (μ, ν) with $\mu, \nu \in \{90^\circ, 180^\circ, 270^\circ\}$ such that $G_{s \rightarrow t}$ admits a $(\chi_{s \rightarrow t}, \mu, \nu)$ -representation (these define $\mathcal{N}_{s \rightarrow t}$), starting from the pairs (μ_i, ν_i) with $\mu_i, \nu_i \in \{0^\circ, 90^\circ, 180^\circ\}$ such that $G_{s_i \rightarrow s}$ admits a $(\chi_{s_i \rightarrow s}, \mu_i, \nu_i)$ -representation (these define $\mathcal{N}_{s_1 \rightarrow s}, \dots, \mathcal{N}_{s_k \rightarrow s}$). When $\mathcal{N}_{s_i \rightarrow s} = \emptyset$, in

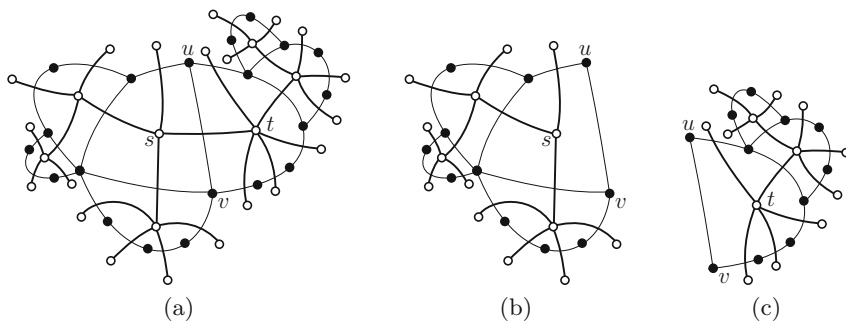


Fig. 5. (a) The graph G (represented with thin lines and black disks) and the extended dual tree \mathcal{T} (represented with thick lines and white disks) of the outerplane embedding \mathcal{O} of G . (b) The graph $G_{s \rightarrow t}$ and the tree $\mathcal{T}_{s \rightarrow t}$. (c) The graph $G_{t \rightarrow s}$ and the tree $\mathcal{T}_{t \rightarrow s}$.

particular, we also have $\mathcal{N}_{s \rightarrow t} = \emptyset$. That is, the non-existence of a $(\chi_{s_i \rightarrow s}, \mu_i, \nu_i)$ -representation of G_{s_i} propagates towards the root of \mathcal{T} in the current traversal.

Clearly, we cannot afford to perform each traversal independently of the other ones, as this would result in a quadratic running time. Then, as in [8], we reuse the already computed labels $\mathcal{N}_{s \rightarrow t}$; this implies that a postorder traversal is not invoked on a tree $\mathcal{T}_{s \rightarrow t}$ if the label $\mathcal{N}_{s \rightarrow t}$ has been computed by a previous traversal. As a result, during the traversals of \mathcal{T} , each edge is traversed at most once in each direction and each node with degree k is processed $O(k)$ times. Differently from [8], we need to handle the possibility that, when a node s of \mathcal{T} is visited in a traversal after the first one, we might not have the sets $\mathcal{N}_{s_i \rightarrow s}$ ready, even for most children of s . This is a consequence of the propagation of the empty sets $\mathcal{N}_{s \rightarrow t}$ described above. Indeed, we cannot even afford to look at all the children s_i of s and see which sets $\mathcal{N}_{s_i \rightarrow s}$ have already been computed and which have not; if the degree of s is k , this would take $\Omega(k)$ time whenever we visit s (potentially k times), which would again result in a quadratic running time.

We cope with this problem by using, for each node of \mathcal{T} , some auxiliary labels that are dynamically computed during the traversals. For example, a label $\eta(s)$ points to a neighbor s_i of s for which $\mathcal{N}_{s_i \rightarrow s} = \emptyset$, two labels $\text{START}(s)$ and $\text{END}(s)$ delimit the interval of neighbors of s for which an optimal pair has already been computed, and a label $a(s)$ stores the number of computed optimal pairs (μ_i, ν_i) such that $\mu_i + \nu_i = 360^\circ$. The labels allow us to quickly determine which sets $\mathcal{N}_{s_i \rightarrow s}$ have already been computed and which have not, and to invoke a traversal recursively on the subtrees $\mathcal{T}_{s_i \rightarrow s}$ for which the sets $\mathcal{N}_{s_i \rightarrow s}$ have not been computed yet. Some labels (for example $a(s)$) store aggregate information on the values of the optimal pairs for the graphs $G_{s_i \rightarrow s}$. Thus, when the sets $\mathcal{N}_{s_i \rightarrow s}$ have been computed for all the children s_i of s , and we are hence in a position to apply the algorithm described in Sect. 3.1, we do not have to spend $O(k)$ time to compute the values a, b, c, d , and t , but we can extract them from the labels associated to s in $O(1)$ time, and then decide in $O(1)$ time whether a solution to the equation $4a' + 3b' + 2c' + d' = t$ subject to $0 \leq a' \leq a, 0 \leq b' \leq b$,

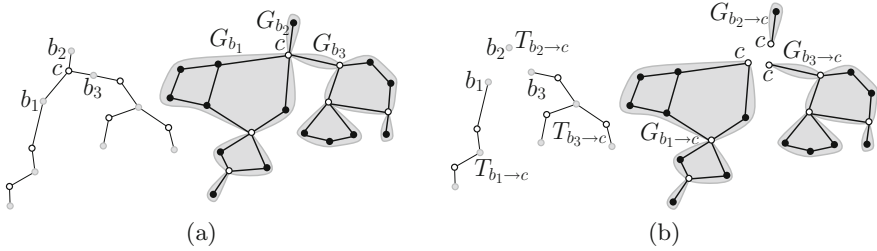


Fig. 6. (a) A graph G and its BC-tree T . The cut-vertices of G are empty disks; the blocks of G are surrounded by gray regions. (b) The graphs $G_{b_i \rightarrow c}$ and the trees $T_{b_i \rightarrow c}$, for $i = 1, 2, 3$, where c is a C-node and b_1, b_2 , and b_3 are its adjacent B-nodes in T .

$0 \leq c' \leq c$, and $0 \leq d' \leq d$ exists; this ultimately determines whether a pair (μ, ν) belongs to $\mathcal{N}_{s \rightarrow t}$. We obtain the following.

Theorem 2. *Let G be an n -vertex 2-connected outerplanar graph and χ be a set of degree-2 vertices of G . There is an $O(n)$ -time algorithm that labels every vertex u of G whose degree is not larger than 3 with a set $\gamma(u)$, which contains all the values $\mu \in \{90^\circ, 180^\circ, 270^\circ\}$ such that G admits a χ -constrained representation (\mathcal{E}, ϕ) in which u is incident to $f_{\mathcal{E}}^*$ and $\phi^{\text{int}}(u) = \mu$.*

3.3 General Outerplanar Graphs

In this section we remove the assumption that the input graph is 2-connected and show how to test whether an n -vertex outerplanar graph G admits a planar rectilinear drawing in $O(n)$ time. Consider the BC-tree T of G [13, 15] and denote by G_b the block corresponding to a B-node b .

We now define a set χ_b for every non-trivial block G_b of G . We initialize $\chi_b = \emptyset$. Then, for every cut-vertex c that is shared by two non-trivial blocks G_{b_1} and G_{b_2} of G , we add c to both χ_{b_1} and χ_{b_2} . This concludes the construction of the sets χ_b . The next lemma justifies our study of χ -constrained representations.

Lemma 2. *For any non-trivial block G_b of G , the restriction to G_b of any rectilinear representation of G is a χ_b -constrained representation of G_b .*

Refer to Fig. 6. Consider any edge bc of T , where b is a B-node and c is a C-node. The removal of bc splits T into two trees. Let $T_{b \rightarrow c}$ be the one containing b . Let $G_{b \rightarrow c}$ be the subgraph of G composed of the blocks corresponding to B-nodes in $T_{b \rightarrow c}$. Let $\chi_{b \rightarrow c}$ be the restriction of χ to the vertices of $G_{b \rightarrow c}$.

We present an $O(n)$ -time algorithm that computes, for every edge bc of T where b is a B-node and c is a C-node, a set $\mathcal{N}_{b \rightarrow c}$, which contains all the values $\mu \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ such that $G_{b \rightarrow c}$ admits a rectilinear representation $(\mathcal{E}_{b \rightarrow c}, \phi_{b \rightarrow c})$ in which c is incident to $f_{\mathcal{E}_{b \rightarrow c}}^*$ and $\phi_{b \rightarrow c}^{\text{int}}(c) = \mu$.

Let b be a B-node of T , let c_i be a C-node adjacent to b , and let $b_{i,1}, \dots, b_{i,m(i)}$ be the B-nodes adjacent to c_i and different from b . We say that c_i is a

friendly neighbor of b if, for every $j = 1, \dots, m(i)$, we have that $\mathcal{N}_{b_i, j \rightarrow c_i} \cap \{0^\circ, 90^\circ, 180^\circ\} \neq \emptyset$ and G_b is trivial, or we have that $\mathcal{N}_{b_i, j \rightarrow c_i} \cap \{0^\circ, 90^\circ\} \neq \emptyset$ and G_b is non-trivial.

Determining the set $\mathcal{N}_{b \rightarrow c}$ for every edge bc of T is sufficient for determining whether G admits a rectilinear representation.

Lemma 3. *We have that G admits a rectilinear representation if and only if there exists a B -node b^* in T such that: (i) if G_{b^*} is non-trivial, then it admits a χ_{b^*} -constrained representation; and (ii) every neighbor of b^* in T is friendly.*

The first step of our algorithm for computing the sets $\mathcal{N}_{b \rightarrow c}$ labels, for each non-trivial block G_b of G , each vertex v of G_b whose degree in G_b is smaller than or equal to 3 with a set $\gamma_b(v)$; this set contains all the values $\mu \in \{90^\circ, 180^\circ, 270^\circ\}$ such that G_b admits a χ_b -constrained representation (\mathcal{E}_b, ϕ_b) in which v is incident to $f_{\mathcal{E}_b}^*$ and $\phi_b^{\text{int}}(v) = \mu$. By Theorem 2, this can be done in $O(n_b)$ time for each non-trivial block G_b of G with n_b vertices, and hence in $O(n)$ time for all the non-trivial blocks of G . Further, for each trivial block G_b of G , we label each end-vertex c of G_b with a set $\gamma_b(c) = \{0^\circ\}$.

For each leaf b of T , we then have $\mathcal{N}_{b \rightarrow c} = \gamma_b(c)$. If b is an internal node of T , in order to compute $\mathcal{N}_{b \rightarrow c}$, our algorithm exploits the following tool.

Lemma 4. *Let b be an internal B -node of T and let c be a C -node of T adjacent to b . Further, let c_1, \dots, c_h be the C -nodes adjacent to b and different from c ; for $i = 1, \dots, h$, let $b_{i,1}, \dots, b_{i,m(i)}$ be the B -nodes adjacent to c_i and different from b . Finally, let $\mu \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$. We have that $\mu \in \mathcal{N}_{b \rightarrow c}$ if and only if $\mu \in \gamma_b(c)$ and c_i is a friendly neighbor of b , for every $i = 1, \dots, h$.*

Similarly to Sect. 3.2, we construct the sets $\mathcal{N}_{b \rightarrow c}$ by performing several traversals of T . Some auxiliary labels are used also in this case, namely $\eta(b)$ points to a neighbor c_i of b that is not friendly and $\xi(b)$ tells us whether every neighbor, or almost every neighbor, of b is friendly. Lemma 4 is used in order to propagate the values $\mathcal{N}_{b \rightarrow c}$ in the tree. We obtain the following main theorem.

Theorem 3. *There is an $O(n)$ -time algorithm that tests whether an n -vertex outerplanar graph admits a planar rectilinear drawing.*

4 Conclusions

In this paper, we proved that the existence of a planar rectilinear drawing of an outerplanar graph can be tested in linear time, both if the plane embedding of the outerplanar graph is prescribed and if it is not. We conclude with two natural generalizations of the questions we answered in this paper. Is it possible to determine in $O(n)$ time the minimum number of bends for a planar orthogonal drawing of an n -vertex outerplanar graph? Is it possible to test in $O(n)$ time whether an n -vertex series-parallel graph admits a planar rectilinear drawing? Didimo et al. [7] proved that the latter question has a positive answer for 2-connected series-parallel graphs with fixed embedding.

Acknowledgments. Thanks to Maurizio “Titto” Patrignani for many explanations about results and techniques from the state of the art.

References

1. Bläsius, T., Lehmann, S., Rutter, I.: Orthogonal graph drawing with inflexible edges. *Comp. Geom.* **55**, 26–40 (2016)
2. Borradaile, G., Klein, P.N., Mozes, S., Nussbaum, Y., Wulff-Nilsen, C.: Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. *SIAM J. Comput.* **46**(4), 1280–1303 (2017)
3. Chang, Y., Yen, H.: On bend-minimized orthogonal drawings of planar 3-graphs. In: Aronov, B., Katz, M.J. (eds.) *SoCG 2017. LIPIcs*, vol. 77, pp. 29:1–29:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2017)
4. Cornelsen, S., Karrenbauer, A.: Accelerated bend minimization. *J. Graph Algorithms Appl.* **16**(3), 635–650 (2012)
5. Di Battista, G., Liotta, G., Vargiu, F.: Spirality and optimal orthogonal drawings. *SIAM J. Comput.* **27**(6), 1764–1811 (1998)
6. Di Giacomo, E., Liotta, G., Montecchiani, F.: Sketched representations and orthogonal planarity of bounded treewidth graphs. In: Archambault, D., Tóth, C.D. (eds.) *GD 2019. LNCS*, vol. 11904, pp. 379–392. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35802-0_29
7. Didimo, W., Kaufmann, M., Liotta, G., Ortali, G.: Rectilinear planarity testing of plane series-parallel graphs in linear time. *CoRR abs/2008.03784* (2020)
8. Didimo, W., Liotta, G., Ortali, G., Patrignani, M.: Optimal orthogonal drawings of planar 3-graphs in linear time. In: Chawla, S. (ed.) *SODA 2020*, pp. 806–825. SIAM (2020)
9. Didimo, W., Liotta, G., Patrignani, M.: Bend-minimum orthogonal drawings in quadratic time. In: Biedl, T., Kerren, A. (eds.) *GD 2018. LNCS*, vol. 11282, pp. 481–494. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04414-5_34
10. Frati, F.: Planar rectilinear drawings of outerplanar graphs in linear time. *CoRR abs/2006.06951* (2020)
11. 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). https://doi.org/10.1007/3-540-62495-3_49
12. Garg, A., Tamassia, R.: On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.* **31**(2), 601–625 (2001)
13. Harary, F.: *Graph Theory*. Addison-Wesley Pub. Co., Reading (1969)
14. Hasan, M.M., Rahman, M.S.: No-bend orthogonal drawings and no-bend orthogonally convex drawings of planar graphs (extended abstract). In: Du, D.-Z., Duan, Z., Tian, C. (eds.) *COCOON 2019. LNCS*, vol. 11653, pp. 254–265. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26176-4_21
15. Hopcroft, J.E., Tarjan, R.E.: Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM* **16**(6), 372–378 (1973)
16. Nomura, K., Tayu, S., Ueno, S.: On the orthogonal drawing of outerplanar graphs. *IEICE Trans.* **88–A**(6), 1583–1588 (2005)
17. Rahman, M.S., Egi, N., Nishizeki, T.: No-bend orthogonal drawings of series-parallel graphs. In: Healy, P., Nikolov, N.S. (eds.) *GD 2005. LNCS*, vol. 3843, pp. 409–420. Springer, Heidelberg (2006). https://doi.org/10.1007/11618058_37

18. 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
19. Rahman, M.S., Nishizeki, T., Naznin, M.: Orthogonal drawings of plane graphs without bends. *J. Graph Algorithms Appl.* **7**(4), 335–362 (2003)
20. Tamassia, R.: On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.* **16**(3), 421–444 (1987)
21. Zhou, X., Nishizeki, T.: Orthogonal drawings of series-parallel graphs with minimum bends. *SIAM J. Discrete Math.* **22**(4), 1570–1604 (2008)