Ortho-Radial Drawing in Near-Linear Time

Yi-Jun Chang
National University of Singapore, Singapore

Abstract

An *orthogonal drawing* is an embedding of a plane graph into a grid. In a seminal work of Tamassia (SIAM Journal on Computing 1987), a simple combinatorial characterization of angle assignments that can be realized as bend-free orthogonal drawings was established, thereby allowing an orthogonal drawing to be described combinatorially by listing the angles of all corners. The characterization reduces the need to consider certain geometric aspects, such as edge lengths and vertex coordinates, and simplifies the task of graph drawing algorithm design.

Barth, Niedermann, Rutter, and Wolf (SoCG 2017) established an analogous combinatorial characterization for *ortho-radial drawings*, which are a generalization of orthogonal drawings to *cylindrical grids*. The proof of the characterization is existential and does not result in an efficient algorithm. Niedermann, Rutter, and Wolf (SoCG 2019) later addressed this issue by developing quadratic-time algorithms for both testing the realizability of a given angle assignment as an ortho-radial drawing without bends and constructing such a drawing.

In this paper, we improve the time complexity of these tasks to near-linear time. We establish a new characterization for ortho-radial drawings based on the concept of a *good sequence*. Using the new characterization, we design a simple greedy algorithm for constructing ortho-radial drawings.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Graph drawing, ortho-radial drawing, topology-shape-metric framework

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.35

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2305.00425

1 Introduction

A *plane graph* is a *planar graph* $G = (V, E)$ with a *combinatorial embedding* $\mathcal{E}$. The combinatorial embedding $\mathcal{E}$ fixes a circular ordering $\mathcal{E}(v)$ of the edges incident to each vertex $v \in V$, specifying the counter-clockwise ordering of these edges surrounding v in the drawing. An *orthogonal drawing* of a plane graph is a drawing of G such that each edge is drawn as a sequence of horizontal and vertical line segments. For example, see Figure 1 for an orthogonal drawing of K_4 with 4 bends. Alternatively, an orthogonal drawing of G can be seen as an embedding of G into a grid such that the edges of G correspond to internally disjoint paths in the grid. Orthogonal drawing is one of the most classical drawing styles studied in the field of graph drawing, and it has a wide range of applications, including VLSI circuit design [7, 40], architectural floor plan design [33], and network visualization [5, 22, 26, 30].

The topology-shape-metric framework. One of the most fundamental quality measures of orthogonal drawings is the number of *bends*. The *bend minimization* problem, which asks for an orthogonal drawing with the smallest number of bends, has been extensively studied over the past 40 years [14, 16, 17, 38, 39, 25]. In a seminal work, Tamassia [39] introduced the *topology-shape-metric* framework to tackle the bend minimization problem. Tamassia showed that an orthogonal drawing can be described combinatorially by an *orthogonal representation*, which consists of an assignment of an angle of degree in $\{90°, 180°, 270°, 360°\}$ to each corner and a designation of the *outer face*. Specifically, Tamassia [39] showed that an orthogonal representation can be realized as an orthogonal drawing with zero bends if and only if the following two conditions are satisfied:

50th International Colloquium on Automata, Languages, and Programming (ICALP 2023).
Editors: Kousha Etessami, Uriel Feige, and Gabriele Puppis; Article No. 35; pp. 35:1–35:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

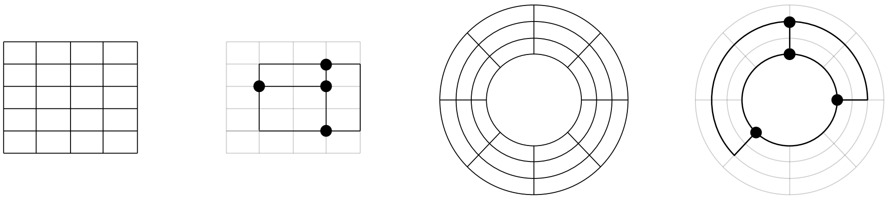


Figure 1 A grid, an orthogonal drawing, a cylindrical grid, and an ortho-radial drawing.

(O1) The sum of angles around each vertex is 360°.

(O2) The sum of angles around each face with k corners is $(k+2) \cdot 180°$ for the outer face and is $(k-2) \cdot 180°$ for the other faces.

An orthogonal representation is *valid* if it satisfies the above conditions (O1) and (O2). Given a valid orthogonal representation, an orthogonal drawing realizing the orthogonal representation can be computed in linear time [29, 39]. This result (shape → metric) allows us to reduce the task of finding a bend-minimized orthogonal drawing (topology → metric) to the conceptually much simpler task of finding a bend-minimized valid orthogonal representation (topology → shape). By focusing on orthogonal representations, we may neglect certain geometric aspects of graph drawing such as edge lengths and vertex coordinates, making the task of algorithm design easier. In particular, given a fixed combinatorial embedding, the task of finding a bend-minimized orthogonal representation can be easily reduced to the computation of a min-cost flow [39].

1.1 Ortho-radial drawing

Ortho-radial drawing is a natural generalization of orthogonal drawing to *cylindrical grids*, whose grid lines consist of concentric circles and straight lines emanating from the center of the circles. Formally, an ortho-radial drawing is defined as a planar embedding where each edge is drawn as a sequence of lines that are either a circular arc of some circle centered on the origin or a line segment of some straight line passing through the origin. We do not allow a vertex to be drawn on the origin, and we do not allow an edge to pass through the origin in the drawing. For example, see Figure 1 for an ortho-radial drawing of K_4 with two bends.

The study of ortho-radial drawing is motivated by its applications [4, 23, 42] in network visualization [41]. For example, ortho-radial drawing is naturally suitable for visualizing metro systems with radial routes and circle routes.

There are three types of faces in an ortho-radial drawing. The face that contains an unbounded region is called the *outer face*. The face that contains the origin is called the *central face*. The remaining faces are called *regular faces*. It is possible that the outer face and the central face are the same face.

Given a plane graph, an *ortho-radial representation* is defined as an assignment of an angle to each corner together with a designation of the central face and the outer face. Barth, Niedermann, Rutter, and Wolf [2] showed that an ortho-radial representation can be realized as an ortho-radial drawing with zero bends if the following three conditions are satisfied:

(R1) The sum of angles around each vertex is 360°.

(R2) The sum s of angles around each face F with k corners satisfies the following.

- $s = (k-2) \cdot 180°$ if F is a regular face.
- $s = k \cdot 180°$ if F is either the central face or the outer face, but not both.
- $s = (k+2) \cdot 180°$ if F is both the central face and the outer face.

(R3) There exists a choice of the *reference edge* $e^\star$ such that the ortho-radial representation does not contain a *strictly monotone cycle*.

Intuitively, this shows that the ortho-radial representations that can be realized as ortho-radial drawings with zero bends can be characterized similarly by examining the angle sum around each vertex and each face, with the additional requirement that the representation does not have a strictly monotone cycle.

The definition of a strictly monotone cycle is technical and depends on the choice of the reference edge $e^\star$, so we defer its formal definition to a subsequent section. The reference edge $e^\star$ is an edge in the contour of the outer face and is required to lie on the outermost circular arc used in an ortho-radial drawing. Informally, a strictly monotone cycle has a structure that is like a loop of ascending stairs or a loop of descending stairs, so a strictly monotone cycle cannot be drawn. The necessity of (R1)–(R3) is intuitive to see. The more challenging and interesting part of the proof in [2] is to show that these three conditions are actually sufficient.

1.2 Previous methods

The proof by Barth, Niedermann, Rutter, and Wolf [2] that conditions (R1)–(R3) are necessary and sufficient is only *existential* in that it does not yield efficient algorithms to check the validity of a given ortho-radial representation and to construct an ortho-radial drawing without bends realizing a given ortho-radial representation.

Checking (R1) and (R2) can be done in linear time in a straightforward manner. The difficult part is to design an efficient algorithm to check (R3). The most naive approach of examining all cycles costs exponential time. The subsequent work by Niedermann, Rutter, and Wolf [35] addressed this gap by showing an $O(n^2)$-time algorithm to decide whether a strictly monotone cycle exists for a given reference edge $e^\star$, where n is the number of vertices in the input graph. They also show an $O(n^2)$-time algorithm to construct an ortho-radial drawing without bends, for any given ortho-radial representation with a reference edge $e^\star$ that does not lead to a strictly monotone cycle.

Rectangulation. The idea behind the proof of Barth, Niedermann, Rutter, and Wolf [2] is a reduction to the easier case where each regular face is *rectangular*. For this case, they provided a proof that conditions (R1)–(R3) are necessary and sufficient, and they also provided an efficient drawing algorithm via a reduction to a flow computation given that (R1)–(R3) are satisfied. For any given ortho-radial representation with n vertices, it is possible to add $O(n)$ additional edges to turn it into an ortho-radial representation where each regular face is rectangular. A major difficulty in the proof of [2] is that they need to ensure that the addition of the edges preserves not only (R1) and (R2) but also (R3). The lack of an efficient algorithm to check whether (R3) is satisfied is precisely the reason that the proof of [2] does not immediately lead to a polynomial-time algorithm.

Quadratic-time algorithms. The above issue was addressed in a subsequent work by Niedermann, Rutter, and Wolf [35]. They provided an $O(n^2)$-time algorithm to find a strictly monotone cycle if one exists, given a fixed choice of the reference edge $e^\star$. This immediately leads to an $O(n^2)$-time algorithm to decide whether a given ortho-radial representation, with a fixed reference edge $e^\star$, admits an ortho-radial drawing. Moreover, combining this $O(n^2)$-time algorithm with the proof of [2] discussed above yields an $O(n^4)$-time drawing algorithm. The time complexity is due to the fact that $O(n)$ edge additions are needed for rectangulation, for each edge addition there are $O(n)$ candidate reference edges to consider, and to test the feasibility of each candidate edge they need to run the $O(n^2)$-time algorithm to test whether the edge addition creates a strictly monotone cycle.

The key idea behind the $O(n^2)$-time algorithm for finding a strictly monotone cycle is a structural theorem that if there is a strictly monotone cycle, then there is a unique outermost one which can be found by a *left-first* DFS starting from any edge in the outermost strictly monotone cycle. The DFS algorithm costs $O(n)$ time. Guessing an edge in the outermost monotone cycle adds an $O(n)$ factor overhead in the time complexity.

Using further structural insights on the augmentation process of [2], the time complexity of the above $O(n^4)$-time drawing algorithm can be lowered to $O(n^2)$ [35]. The reason for the quadratic time complexity is that for each of the $O(n)$ edge additions, a left-first DFS starting from the newly added edge is needed to test whether the addition of this edge creates a strictly monotone cycle.

1.3 Our new method

For both validity testing (checking whether a given angle assignment induces a strictly monotone cycle) and drawing (finding a geometric embedding realizing a given ortho-radial representation), the two algorithms in [35] naturally cost $O(n^2)$ time, as they both require performing left-first DFS $O(n)$ times.

In this paper, we present a new method for ortho-radial drawing that is not based on rectangulation and left-first DFS. We design a simple $O(n \log n)$-time greedy algorithm that simultaneously accomplishes both validity testing and drawing, for the case where the reference edge $e^\star$ is fixed. If a reference edge $e^\star$ is not fixed, our algorithm costs $O(n \log^2 n)$ time, where the extra $O(\log n)$ factor is due to a binary search over the set of candidates for the reference edge. At a high level, our algorithm tries to construct an ortho-radial drawing in a piece-by-piece manner. If at some point no progress can be made in that the current partial drawing cannot be further extended, then the algorithm can identify a strictly monotone cycle to certify the non-existence of a drawing.

Good sequences. The core of our method is the notion of a *good sequence*, which we briefly explain below. An ortho-radial representation satisfying (R1) and (R2), with a fixed reference edge $e^\star$, determines whether an edge e is a vertical edge (i.e., e is drawn as a segment of a straight line passing through the origin) or horizontal (i.e., e is drawn as a circular arc of some circle centered on the origin). Let E_h denote the set of horizontal edges, oriented in the clockwise direction, and let $\mathcal{S}_\mathsf{h}$ denote the set of connected components induced by E_h. Note that each component $S \in \mathcal{S}_\mathsf{h}$ is either a path or a cycle. The exact definition of a good sequence is technical, so we defer it to a subsequent section. Intuitively, a good sequence is an ordering of $\mathcal{S}_\mathsf{h} = (S_1, S_2, \ldots, S_k)$, where $k = |\mathcal{S}_\mathsf{h}|$, that allows us to design a simple linear-time greedy algorithm constructing an ortho-radial drawing in such a way that S_1 is drawn on the circle $r = k$, S_2 is drawn on the circle $r = k - 1$, and so on.

In general, a good sequence might not exist, even if the given ortho-radial representation admits an ortho-radial drawing. In such a case, we show that we may add *virtual edges* to transform the ortho-radial representation into one where a good sequence exists. We will design a greedy algorithm for adding virtual edges and constructing a good sequence. In each step, we add virtual vertical edges to the current graph and append a new element $S \in \mathcal{S}_\mathsf{h}$ to the end of our sequence. In case we are unable to find any suitable $S \in \mathcal{S}_\mathsf{h}$ to extend the sequence, we can extract a strictly monotone cycle to certify the non-existence of an ortho-radial drawing. A major difference between our method and the approach based on rectangulation in [2, 35] is that the cost for adding a new virtual edge is only $O(\log n)$ in our algorithm. As we will later see, in our algorithm, in order to identify new virtual edges to be added, we only need to do some simple local checks such as calculating the sum of angles, and there is no need to do a full left-first DFS to test whether a newly added edge creates a strictly monotone cycle.

Open questions. While we show a nearly linear-time algorithm for the (shape → metric)-step (i.e., from ortho-radial representations to ortho-radial drawings), essentially nothing is known about the (topology → shape)-step (from planar graphs to ortho-radial representations). While the task of finding a *bend-minimized orthogonal representation* of a given plane graph can be easily reduced to the computation of a minimum cost flow [39], such a reduction does not apply to ortho-radial representations, as network flows do not work well with the notion of strictly monotone cycles. It remains an open question as to whether a bend-minimized ortho-radial representation of a plane graph can be computed in polynomial time.

1.4 Related work

The bend minimization problem for orthogonal drawings for planar graphs of maximum degree 4 without a fixed combinatorial embedding is NP-hard [24, 25]. If the combinatorial embedding is fixed, the topology-shape-metric framework of Tamassia [39] reduces the bend minimization problem to a min-cost flow computation. The algorithm of Tamassia [39] costs $O(n^2 \log n)$ time. The time complexity was later improved to $O\left(n^{7/4}\sqrt{\log n}\right)$ [25] and then to $O\left(n^{3/2}\log n\right)$ [14]. A recent $O(n \operatorname{poly}\log n)$-time planar min-cost flow algorithm [20] implies that the bend minimization problem can be solved in $O(n \operatorname{poly}\log n)$ time if the combinatorial embedding is fixed.

If the combinatorial embedding is not fixed, the NP-hardness result of [24, 25] can be bypassed if the first bend on each edge does not incur any cost [9] or if we restrict ourselves to some special class of planar graphs. In particular, for planar graphs with maximum degree 3, it was shown that the bend-minimization can be solved in polynomial time [16]. After a series of improvements [13, 17, 19], we now know that a bend-minimized orthogonal drawing of a planar graph with maximum degree 3 can be computed in $O(n)$ time [17].

The topology-shape-metric framework [39] is not only useful in bend minimization, but it is also, implicitly or explicitly, behind the graph drawing algorithms for essentially all computational problems in orthogonal drawing and its variants, such as morphing orthogonal drawings [8], allowing vertices with degree greater than 4 [15, 31, 36], restricting the direction of edges [18, 21], drawing cluster graphs [10], and drawing dynamic graphs [11].

The study of ortho-radial drawing by Barth, Niedermann, Rutter, and Wolf [2, 35] extended the topology-shape-metric framework [39] to accommodate cylindrical grids. Before these works [2, 35], a combinatorial characterization of drawable ortho-radial representation is only known for paths, cycles, and theta graphs [28], and for the special case where the graph is 3-regular and each regular face in the ortho-radial representation is a rectangle [27].

1.5 Organization

In Section 2, we discuss the basic graph terminology used in this paper, review some results in previous works [2, 35], and state our main theorems. In Section 3, we give a technical overview of our proof. We conclude in Section 4 with discussions on possible future directions.

2 Preliminaries

Throughout the paper, let $G = (V, E)$ be a planar graph of maximum degree at most 4 with a fixed combinatorial embedding $\mathcal{E}$ in the sense that, for each vertex $v \in V$, a circular ordering $\mathcal{E}(v)$ of its incident edges is given to specify the counter-clockwise ordering of these edges surrounding v in a planar embedding. As we will discuss in the full version of the paper, we may assume that the input graph G is *simple* and *biconnected*. In this section, we introduce some basic graph terminology and review some results from the paper [3], which is a merge of the two papers [2, 35] on ortho-radial drawing.

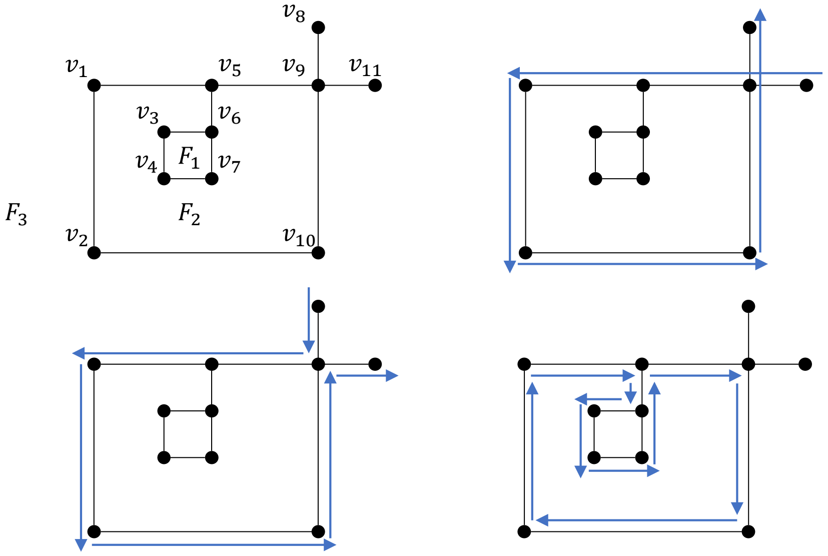


Figure 2 A non-crossing-free path, a crossing-free path, and a facial cycle.

Paths and cycles. Unless otherwise stated, all edges, paths, and cycles are assumed to be directed. We write $\overline{e}$, $\overline{P}$, and $\overline{C}$ to denote the *reversal* of an edge e, a path P, and a cycle C, respectively. We allow paths and cycles to have repeated vertices and edges. We say that a path or a cycle is *simple* if it does not have repeated vertices. Following [3], we say that a path or a cycle is *crossing-free* if it satisfies the following conditions:

- The path or the cycle does not contain repeated undirected edges.
- For each vertex v that appears multiple times in the path or the cycle, the ordering of the edges incident to v appearing in the path or the cycle respects the ordering of $\mathcal{E}(v)$ or its reversal.

Although a crossing-free path or a crossing-free cycle might touch a vertex multiple times, the path or the cycle never crosses itself. For any face F, we define the *facial cycle* C_F to be the clockwise traversal of its contour. In general, a facial cycle might not be a simple cycle as it can contain repeated edges. If we assume that G is biconnected, then each facial cycle of G must be a simple crossing-free cycle. See Figure 2 for an illustration of different types of paths and cycles. The path $(v_{11}, v_9, v_5, v_1, v_2, v_{10}, v_9, v_8)$ is not crossing-free as the path crosses itself at v_9. The path $(v_8, v_9, v_5, v_1, v_2, v_{10}, v_9, v_{11})$ is crossing-free since it respects the ordering $\mathcal{E}(v)$ for $v = v_9$. The cycle $C = (v_1, v_5, v_6, v_3, v_4, v_7, v_6, v_5, v_9, v_{10}, v_2)$ is the facial cycle of F_2. The cycle C is not a crossing-free cycle as it traverses the undirected edge $\{v_5, v_6\}$ twice, from opposite directions.

Ortho-radial representations and drawings. A *corner* is an ordered pair of undirected edges (e_1, e_2) incident to v such that e_2 immediately follows e_1 in the counter-clockwise circular ordering $\mathcal{E}(v)$. Given a planar graph $G = (V, E)$ with a fixed combinatorial embedding $\mathcal{E}$, an ortho-radial representation $\mathcal{R} = (\phi, F_{\mathsf{c}}, F_{\mathsf{o}})$ of G is defined by the following components:

- An assignment ϕ of an angle $a \in \{90°, 180°, 270°\}$ to each corner of G.
- A designation of a face of G as the central face F_{c}.
- A designation of a face of G as the outer face F_{o}.

For the special case where v has only one incident edge e, we view (e, e) as a 360° corner. This case does not occur if we consider biconnected graphs.

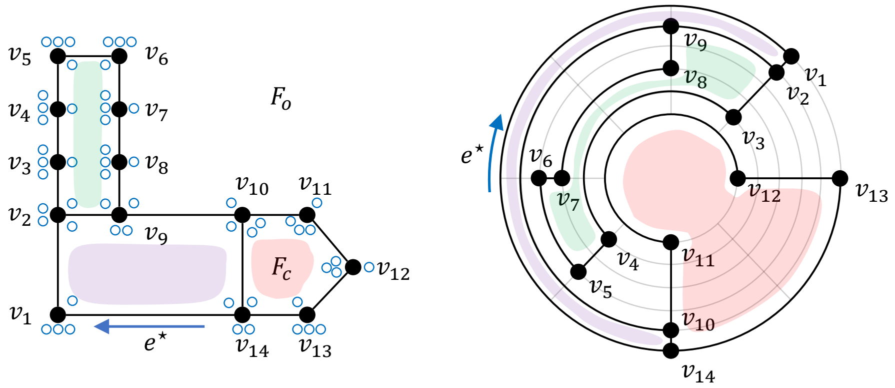


Figure 3 A drawing of an ortho-radial representation with a reference edge, where the small blue circles in the left figure denote the angles in the representation that are realized in the right figure.

An ortho-radial representation $\mathcal{R} = (\phi, F_{\mathsf{c}}, F_{\mathsf{o}})$ is *drawable* if the representation can be realized as an ortho-radial drawing of G with zero bends such that the angle assignment ϕ is satisfied, the central face F_{c} contains the origin, the outer face F_{o} contains an unbounded region.

Recall that, by the definition of ortho-radial drawing, in an ortho-radial drawing with zero bends, each edge is either drawn as a line segment of a straight line passing the origin or drawn as a circular arc of a circle centered at the origin. We also consider the setting where the *reference edge* $e^\star$ is fixed. In this case, there is an additional requirement that the reference edge $e^\star$ has to lie on the outermost circular arc used in the drawing and follows the clockwise direction. If such a drawing exists, we say that $(\mathcal{R}, e^\star)$ is *drawable.* See Figure 3 for an example of a drawing of an ortho-radial representation $\mathcal{R}$ with the reference edge $e^\star = (v_{14}, v_5)$. In the figure, we use $\circ$, $\circ\circ$, and $\circ\circ\circ$ to indicate a 90°, a 180°, and a 270° angle assigned to a corner, respectively.

It was shown in [3] that $(\mathcal{R}, e^\star)$ is drawable if and only if the ortho-radial representation $\mathcal{R}$ satisfies (R1) and (R2) and the reference edge $e^\star$ does not lead to a strictly monotone cycle. Since it is straightforward to test whether (R1) and (R2) are satisfied in linear time, from now on, unless otherwise stated, we assume that (R1) and (R2) are satisfied for the ortho-radial representation $\mathcal{R}$ under consideration.

Combinatorial rotations. Consider a 2-length path $P = (u, v, w)$ that passes through v such that $u \neq w$. Given the angle assignment ϕ, P is either a 90° left turn, a straight line, or a 90° right turn. We define the *combinatorial rotation* of P as follows.

$$\mathsf{rotation}(P) = \begin{cases} -1, & P \text{ is a } 90^\circ \text{ left turn,} \\ 0, & P \text{ is a straight line,} \\ 1, & P \text{ is a } 90^\circ \text{ right turn.} \end{cases}$$

More formally, let $S = (e_1, \ldots, e_k)$ be the contiguous subsequence of edges starting from $e_1 = \{u, v\}$ and ending at $e_k = \{v, w\}$ in the circular ordering $\mathcal{E}(v)$ of the undirected edges incident to v. Then $\sum_{j=1}^{k-1} \phi(e_j, e_{j+1}) - 180^\circ$ equals the degree of the turn of P at the intermediate vertex v, so the combinatorial rotation of P is $\mathsf{rotation}(P) = \left(\sum_{j=1}^{k-1} \phi(e_j, e_{j+1}) - 180^\circ\right) / 90^\circ$.

For the special case where $u = w$, the rotation of $P = (u, v, u)$ can be a 180° left turn, in which case $\mathsf{rotation}(P) = -2$, or a 180° right turn, in which case $\mathsf{rotation}(P) = 2$. For example, consider the directed edge $e = (u, v)$ where P first goes from u to v along the right side of e and then goes from v back to u along the left side of e. Then P is considered a 180° left turn, and similarly, $\overline{P}$ is considered a 180° right turn. In particular, if $P = (u, v, u)$ is a subpath of a facial cycle C, then P is always considered as a 180° left turn, and so $\mathsf{rotation}(P) = -2$.

For a crossing-free path P of length more than 2, we define $\mathsf{rotation}(P)$ as the sum of the combinatorial rotations of all 2-length subpaths of P. Similarly, for a cycle C of length more than 2, we define $\mathsf{rotation}(C)$ as the sum of the combinatorial rotations of all 2-length subpaths of C. Same as [2, 35], based on this notion, we may restate condition (R2).

(R2′) For each face F, the combinatorial rotation of its facial cycle C_F satisfies the following:

$$\mathsf{rotation}(C_F) = \begin{cases} 4, & F \text{ is a regular face,} \\ 0, & F \text{ is either the central face or the outer face, but not both,} \\ -4, & F \text{ is both the central face and the outer face.} \end{cases}$$

For example, consider the ortho-radial representation shown in Figure 3. The path $P = (v_{10}, v_{11}, v_{12}, v_{13}, v_{14})$ has $\mathsf{rotation}(P) = -1$ since it makes two 90° left turns and one 90° right turn. The cycle $C = (v_{10}, v_{11}, v_{12}, v_{13}, v_{14})$ is the facial cycle of the central face, and it has $\mathsf{rotation}(C) = 0$.

We briefly explain the equivalence between the new and the old definitions of (R2). If F is a regular face with k corners, then in the original definition of (R2), it is required that the sum s of angles around F is $s = (k-2) \cdot 180°$. Since the facial cycle C_F traverses the contour of F in the clockwise direction, the number of 90° right turns minus the number of 90° left turns must be exactly 4. Therefore, $s = (k-2) \cdot 180°$ is the same as $\mathsf{rotation}(C_F) = 4$, as each 90° right turn contributes 1 and each 90° left turn contributes -1 in the calculation of $\mathsf{rotation}(C_F)$.

Interior and exterior regions of a cycle. Any cycle C partitions the remaining graph into two parts. If C is a facial cycle, then one part is empty. The direction of C is clockwise with respect to one of the two parts. The part with respect to which C is clockwise, together with C itself, is called the *interior* of C. Similarly, the part with respect to which C is counter-clockwise, together with C itself, is called the *exterior* of C. In particular, if a vertex v lies in the interior of C, then v must be in the exterior of $\overline{C}$.

This above definition is consistent with the notion of facial cycle in that any face F is in the interior of its facial cycle C_F. Depending on the context, the interior or the exterior of a cycle can be viewed as a subgraph, a set of vertices, a set of edges, or a set of faces. For example, consider the cycle $C = (v_1, v_2, v_{10}, v_9, v_5)$ of the plane graph shown in Figure 2. The interior of C is the subgraph induced by v_8, v_{11}, and all vertices in C. The exterior of C is the subgraph induced by v_3, v_4, v_6, v_7, and all vertices in C. The cycle C partitions the faces into two parts: The interior of C contains F_3, and the exterior of C contains F_1 and F_2.

Let C be a simple cycle oriented in such a way that the outer face F_{o} lies in its exterior. Following [3], we say that C is *essential* if the central face F_{c} is in the interior of C. Otherwise we say that C is *non-essential.* The following lemma was proved in [3].

▶ **Lemma 1** ([3]). *Let C be a simple cycle oriented in such a way that the outer face F_{o} lies in its exterior, then the combinatorial rotation of C satisfies the following:*

$$\mathsf{rotation}(C) = \begin{cases} 4, & C \text{ is an essential cycle,} \\ 0, & C \text{ is a non-essential cycle.} \end{cases}$$

In the above lemma, we implicitly assume that (R1) and (R2) are satisfied. The intuition behind the lemma is that an essential cycle behaves like the facial cycle of the outer face or the central face, and a non-essential cycle behaves like the facial cycle of a regular face.

Subgraphs. When we take a subgraph H of G, the combinatorial embedding, the angle assignment, the central face, and the outer face of H are inherited from G naturally. For example, suppose that $\mathcal{E}(v) = (e_1, e_2, e_3)$ with $\phi(e_1, e_2) = 90°$, $\phi(e_2, e_3) = 90°$, and $\phi(e_3, e_1) = 180°$ in G. Suppose that v is incident only to two edges e_1 and e_2 in H, then the angle assignment ϕ_H for the two corners surrounding v in H will be $\phi_H(e_1, e_2) = 90°$ and $\phi_H(e_2, e_1) = 270°$.

Each face of G is contained in exactly one face of H, A face in H can contain multiples faces of G. A face of H is said to be the central face if it contains the central face of G. Similarly, A face of H is said to be the outer face if it contains the outer face of G.

For example, consider the subgraph H induced by $\{v_2, v_3, \ldots, v_9\}$ in the ortho-radial representation in Figure 3. In H, v_9 is incident to only two edges $e_1 = \{v_8, v_9\}$ and $e_2 = \{v_2, v_9\}$, and the angle assignment ϕ_H for the two corners surrounding v_9 in H are $\phi_H(e_1, e_2) = 90°$ and $\phi_H(e_2, e_1) = 270°$. The outer face and the central face of H are the same.

Defining direction via reference paths. Following [3], for any two edges $e = (u, v)$ and $e' = (x, y)$, we say that a crossing-free path P is a reference path for e and e' if P starts at u or v and ends at x or y such that P does not contain any of the edges in $\{e, \overline{e}, e', \overline{e'}\}$. Given a reference path P for $e = (u, v)$ and $e' = (x, y)$, we define the *combinatorial direction* of e' with respect to e and P as follows.

$$\mathsf{direction}(e, P, e') = \begin{cases} \mathsf{rotation}(e \circ P \circ e'), & P \text{ starts at } v \text{ and ends at } x, \\ \mathsf{rotation}(\overline{e} \circ P \circ e') + 2, & P \text{ starts at } u \text{ and ends at } x, \\ \mathsf{rotation}(e \circ P \circ \overline{e'}) - 2, & P \text{ starts at } v \text{ and ends at } y, \\ \mathsf{rotation}(\overline{e} \circ P \circ \overline{e'}), & P \text{ starts at } u \text{ and ends at } y. \end{cases}$$

Here $P \circ Q$ denotes the concatenation of the paths P and Q. An edge e is interpreted as a 1-length path. In the definition of $\mathsf{direction}(e, P, e')$, we allow the possibility that a reference path P consists of a single vertex. If $v = x$ and $u \neq w$, then we may choose P to be the 0-length path consisting of a single vertex $v = x$, in which case $\mathsf{direction}(e, P, e')$ is simply the combinatorial rotation of the 2-length path (u, v, y). We do not consider the cases where $e = e'$ or $e = \overline{e'}$.

Consider the reference edge $e = (v_{14}, v_1)$ in the ortho-radial representation of Figure 3. We measure the direction of $e' = (v_8, v_9)$ from e with different choices of the reference path P. If $P = (v_1, v_2, v_9)$, then $\mathsf{direction}(e, P, e') = \mathsf{rotation}(e \circ P \circ \overline{e'}) - 2 = -1$. If $P = (v_{14}, v_{10}, v_9)$, then we also have $\mathsf{direction}(e, P, e') = \mathsf{rotation}(\overline{e} \circ P \circ \overline{e'}) = -1$. If we select $P = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8)$, then we get a different value of $\mathsf{direction}(e, P, e') = \mathsf{rotation}(e \circ P \circ e') = 3$. As we will later discuss, $\mathsf{direction}(e, P, e') \bmod 4$ is invariant under the choice of P.

In the definition of $\mathsf{direction}(e, P, e')$, the additive $+2$ in $\mathsf{rotation}(\overline{e} \circ P \circ e') + 2$ is due to the fact that the actual path that we intend to consider is $e \circ \overline{e} \circ P \circ e'$, where we make a 180° right turn in $e \circ \overline{e}$, which contributes $+2$ in the calculation of the combinatorial rotation. Similarly, the additive -2 in $\mathsf{rotation}(e \circ P \circ \overline{e'}) - 2$ is due to the fact that the actual path that we intend to consider is $e \circ P \circ \overline{e'} \circ e'$, where we make a 180° left turn in $\overline{e'} \circ e'$. There is no additive term in $\mathsf{rotation}(\overline{e} \circ P \circ \overline{e'})$ because of the cancellation of the 180° right turn $e \circ \overline{e}$ and the 180° left turn $\overline{e'} \circ e'$. The reason why $e \circ \overline{e}$ has to be a right turn and $\overline{e'} \circ e'$ has to be a left turn will be explained later.

See Figure 4 for an example of the calculation of an edge direction. The direction of $e = (u_1, u_2)$ with respect to $e^\star = (v_1, v_2)$ and the reference path $P = (v_1, v_5, v_4, u_1)$ can be calculated by $\mathsf{rotation}(\overline{e^\star} \circ P \circ e') + 2 = 1$ according to the formula above, where the additive $+2$ is due to the 180° right turn at $e^\star \circ \overline{e^\star}$.

Edge directions. Imagining that the origin is the south pole, in an ortho-radial drawing with zero bends, each edge e is either drawn in one of the following four directions:

- e points towards the *north* direction if e is drawn as a line segment of a straight line passing the origin, where e is directed away from the origin.
- e points towards the *south* direction if e is drawn as a line segment of a straight line passing the origin, where e is directed towards the origin.
- e points towards the *east* direction if e is drawn as a circular arc of a circle centered at the origin in the clockwise direction.
- e points towards the *west* direction if e is drawn as a circular arc of a circle centered at the origin in the counter-clockwise direction.

We say that e is a *vertical* edge if e points towards north or south. Otherwise, we say that e is a *horizontal* edge. We argue that as long as (R1) and (R2) are satisfied, the direction of any edge e is uniquely determined by the ortho-radial representation.

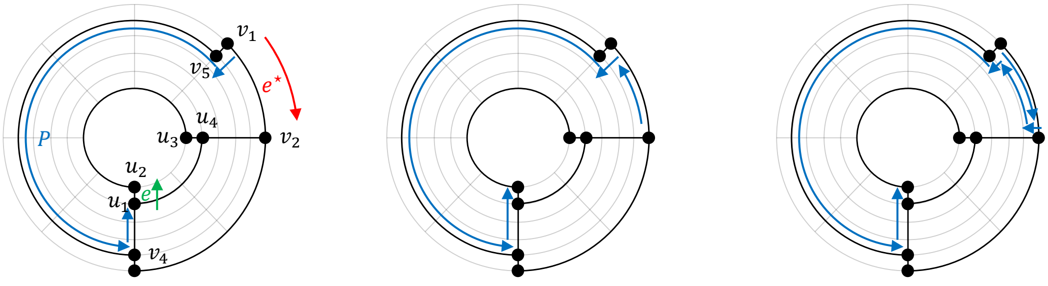
For the reference edge $e^\star$, it is required that $e^\star$ points east, and so $\overline{e^\star}$ points west. Consider any edge e that is neither $e^\star$ nor $\overline{e^\star}$. It is clear that the value of $\mathsf{direction}(e^\star, P, e)$ determines the direction of e in that the direction of e is forced to be east, south, west, or north if $\mathsf{direction}(e^\star, P, e) \bmod 4$ equals 0, 1, 2, or 3, respectively. For example, in the ortho-radial representation of Figure 3, the edge $e' = (v_8, v_9)$ is a vertical edge in the north direction, as we have calculated that $\mathsf{direction}(e^\star, P, e') \bmod 4 = 3$.

▶ **Lemma 2** ([3]). *For any two edges e and e', the value of* $\mathsf{direction}(e, P, e') \bmod 4$ *is invariant under the choice of the reference path P.*

The above lemma shows that $\mathsf{direction}(e^\star, P, e) \bmod 4$ is invariant under the choice of the reference path P, so the direction of each edge in an ortho-radial representation is well defined, even for the case that $(\mathcal{R}, e^\star)$ might not be drawable. Given the reference edge $e^\star$, we let E_{h} denote the set of all horizontal edges in the east direction, and let E_{v} denote the set of all vertical edges in the north direction.

Horizontal segments. We require that in a drawing of $(\mathcal{R}, e^\star)$, the reference edge $e^\star$ lies on the outermost circular arc used in the drawing, so not every edge in $\overline{C_{F_\mathsf{o}}}$ is eligible to be a reference edge. To determine whether an edge $e \in \overline{C_{F_\mathsf{o}}}$ is eligible to be a reference edge, we need to introduce some terminology.

Given the reference edge $e^\star$, the set E_{v} of vertical edges in the north direction and the set E_{h} of horizontal edges in the east direction are fixed. Let $\mathcal{S}_{\mathsf{h}}$ denote the set of connected components induced by E_{h}. Each component $S \in \mathcal{S}_{\mathsf{h}}$ is either a path or a cycle, and so in any drawing of $\mathcal{R}$, there is a circle C centered at the origin such that S must be drawn as C or a circular arc of C. We call each component $S \in \mathcal{S}_{\mathsf{h}}$ a *horizontal segment.*



$\overline{e^\star} \circ P \circ e$

$e^\star \circ \overline{e^\star} \circ P \circ e$

Figure 4 The calculation of $\mathsf{direction}(e^\star, P, e)$.

Each horizontal segment $S \in \mathcal{S}_\mathsf{h}$ is written as a sequence of vertices $S = (v_1, v_2, \ldots, v_s)$, where s is the number of vertices in S, such that $(v_i, v_{i+1}) \in E_\mathsf{h}$ for each $1 \leq i < s$. If S is a cycle, then we additionally have $(v_s, v_1) \in E_\mathsf{h}$, so $S = (v_1, v_2, \ldots, v_s)$ is a circular order. When S is a cycle, we use modular arithmetic on the indices so that $v_{s+1} = v_1$. We write $\mathcal{N}_\mathsf{north}(S)$ to denote the set of vertical edges $e = (x, y) \in E_\mathsf{v}$ such that $x \in S$. Similarly, $\mathcal{N}_\mathsf{south}(S)$ is the set of vertical edges $e = (x, y) \in E_\mathsf{v}$ such that $y \in S$. We assume that the edges in $\mathcal{N}_\mathsf{north}(S)$ and $\mathcal{N}_\mathsf{south}(S)$ are ordered according to the indices of their endpoints in S. The ordering is sequential if S is a path and is circular if S is a cycle. Consider the ortho-radial representation $\mathcal{R}$ given in Figure 3 as an example. The horizontal segment $S = (v_{10}, v_9, v_2)$ has $\mathcal{N}_\mathsf{south}(S) = ((v_{11}, v_{10}), (v_8, v_9), (v_3, v_2))$ and $\mathcal{N}_\mathsf{north}(S) = ((v_{10}, v_{14}), (v_2, v_1))$.

Observe that $\mathcal{N}_\mathsf{north}(S) = \emptyset$ for the horizontal segment $S \in \mathcal{S}_\mathsf{h}$ that contains $e^\star$ is a necessary condition that a drawing of $\mathcal{R}$ where $e^\star$ lies on the outermost circular arc exists. This condition can easily be checked in linear time.

Spirality. Intuitively, $\mathsf{direction}(e, P, e')$ quantifies the degree of *spirality* of e' with respect to e and P. Unfortunately, Lemma 2 does not hold if we replace $\mathsf{direction}(e, P, e') \bmod 4$ with $\mathsf{direction}(e, P, e')$. A crucial observation made in [3] is that such a replacement is possible if we add some restrictions about the positions of e, e', and P. See the following lemma.

▶ **Lemma 3** ([3]). *Let C and C' be essential cycles such that C' lies in the interior of C. Let e be an edge on C. Let e' be an edge on C'. The value of* $\mathsf{direction}(e, P, e')$ *is invariant under the choice of the reference path P, over all paths P in the interior of C and in the exterior of C'.*

Recall that we require a reference path to be crossing-free. This requirement is crucial in the above lemma. If we allow P to be a general path that is not crossing-free, then we may choose P in such a way that P repeatedly traverses a *non-essential* cycle many times, so that $\mathsf{direction}(e, P, e')$ can be made arbitrarily large and arbitrarily small.

Setting $e = e^\star$ and $C = \overline{C_{F_\mathsf{o}}}$ in the above lemma, we infer that $\mathsf{direction}(e^\star, P, e')$ is determined once we fix an essential cycle C' that contains e' and only consider reference paths P that lie in the exterior of C'. The condition for the lemma is satisfied because $\overline{C_{F_\mathsf{o}}}$ is the outermost essential cycle in that all other essential cycles are in the interior of $\overline{C_{F_\mathsf{o}}}$. The reason why we set $C = \overline{C_{F_\mathsf{o}}}$ and not $C = C_{F_\mathsf{o}}$ is that F_o has to be in the exterior of C. Note that the assumption that G is biconnected ensures that each facial cycle is simple.

Let C be an essential cycle and let e be an edge in C. In view of the above, following [3], we define the *edge label* $\ell_C(e)$ of e with respect to C as the value of $\mathsf{direction}(e^\star, P, e)$, for any choice of reference path P in the exterior of C'. For the special case that $e = e^\star$ and $C = \overline{C_{F_\mathsf{o}}}$,

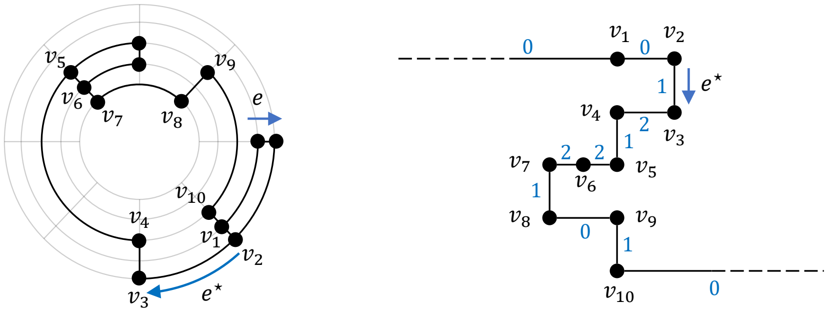


Figure 5 Changing the reference edge to e leads to a strictly monotone cycle.

we let $\ell_C(e) = 0$. Intuitively, the value $\ell_C(e)$ quantifies the degree of spirality of e from $e^\star$ if we restrict ourselves to the exterior of C. Consider the edge $e = (u_1, u_2)$ in the essential cycle $C = (u_1, u_2, u_3, u_4)$ in Figure 4 as an example. We have $\ell_C(e) = \mathsf{direction}(e^\star, P, e) = 1$, since the reference path $P = (v_1, v_5, v_4, u_1)$ lies in exterior of C.

We briefly explain the formula of $\mathsf{direction}(e, P, e')$: As discussed earlier, in the definition of $\mathsf{direction}(e, P, e')$, the additive $+2$ in $\mathsf{rotation}(\overline{e} \circ P \circ e') + 2$ is due to the fact that the actual path that we want to consider is $e \circ \overline{e} \circ P \circ e'$, where we make a 180° right turn in $e \circ \overline{e}$. The reason why $e \circ \overline{e}$ has to be a right turn is because of the scenario considered in Lemma 3, where e is an edge in C. To ensure that we stay in the interior of C in the traversal from e to e' via the path $e \circ \overline{e} \circ P \circ e'$, the 180° turn of $e \circ \overline{e}$ has to be a right turn. The remaining part of the formula of $\mathsf{direction}(e, P, e')$ can be explained similarly.

Monotone cycles. We are now ready to define the notion of strictly monotone cycles used in (R3). We say that an essential cycle C is *monotone* if all its edge labels $\ell_C(e)$ are non-negative or all its edge labels $\ell_C(e)$ are non-positive. Let C be an essential cycle that is monotone. If C contains at least one positive edge label, then we say that C is *increasing*. If C contains at least one negative edge label, then we say that C is *decreasing*. We say that C is *strictly monotone* if C is either decreasing or increasing but not both.

Intuitively, an increasing cycle is like a loop of descending stairs, and a decreasing cycle is like a loop of ascending stairs, so they are not drawable. It was proved in [3] that $(\mathcal{R}, e^\star)$ is drawable if and only if it does not contain a strictly monotone cycle. Recall again that, throughout the paper, unless otherwise stated, we assume that the given ortho-radial representation already satisfies (R1) and (R2).

▶ **Lemma 4** ([3]). *An ortho-radial representation $\mathcal{R}$, with a fixed reference edge $e^\star$ such that $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ for the horizontal segment $S \in \mathcal{S}_{\mathsf{h}}$ that contains $e^\star$, is drawable if and only if it does not contain a strictly monotone cycle.*

Consider Figure 5 as an example. The ortho-radial representation $\mathcal{R}$ is drawable with the reference edge $e^\star$. If we change the reference edge to e, then $(\mathcal{R}, e)$ become undrawable, as the essential cycle $C = (v_1, v_2, \ldots, v_{10})$ is increasing. With respect to the reference edge e, all the edge labels on the cycle C are non-negative, with some of them being positive. We are ready to state our main results.

▶ **Theorem 5.** *There is an $O(n \log n)$-time algorithm $\mathcal{A}$ that outputs either a drawing of $(\mathcal{R}, e^\star)$ or a strictly monotone cycle of $(\mathcal{R}, e^\star)$, for any given ortho-radial representation $\mathcal{R}$ of an n-vertex biconnected simple graph, with a fixed reference edge $e^\star$ such that $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ for the horizontal segment $S \in \mathcal{S}_{\mathsf{h}}$ that contains $e^\star$.*

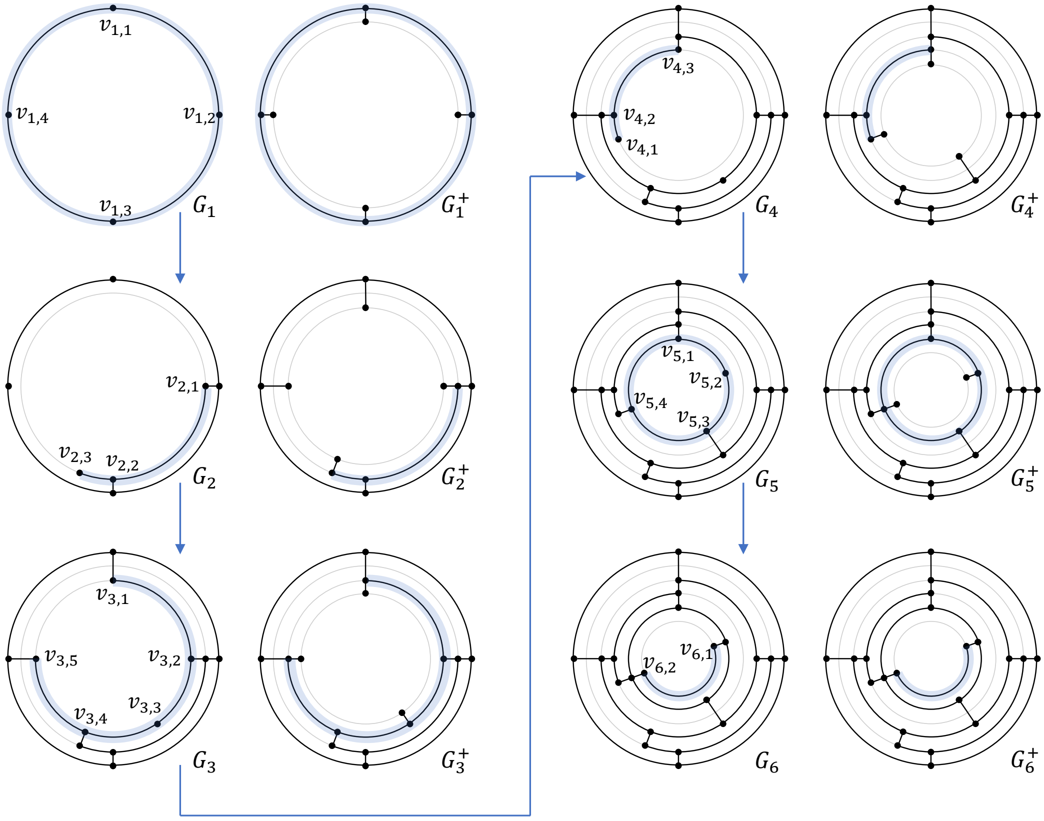


Figure 6 Constructing a good drawing for a good sequence.

The above theorem improves the previous algorithm of [35] which costs $O(n^2)$ time. If the output of $\mathcal{A}$ is a strictly monotone cycle, then the cycle certifies the non-existence of a drawing, by Lemma 4. We also extend the above theorem to the case where the reference edge is not fixed.

▶ **Theorem 6.** *There is an $O(n \log^2 n)$-time algorithm $\mathcal{A}$ that decides whether an ortho-radial representation $\mathcal{R}$ of an n-vertex biconnected simple graph is drawable. If $\mathcal{R}$ is drawable, then $\mathcal{A}$ also computes a drawing of $\mathcal{R}$.*

The proofs of Theorems 5 and 6 are left to the full version of the paper.

3 Technical overview

Let $A = (S_1, S_2, \ldots, S_k)$ be any sequence of k horizontal segments. We consider the following terminology for each $1 \le i \le k$, where k is the length of the sequence A.

- Let G_i be the subgraph of G induced by the horizontal edges in $S_1, S_2, \ldots, S_i$ and the set of all vertical edges whose both endpoints are in $S_1, S_2, \ldots, S_i$. Let F_i be the central face of G_i, and let C_i be the facial cycle of F_i.
- We extend the notion $\mathcal{N}_{\mathsf{south}}(S)$ to a sequence of horizontal segments: $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_i)$ is defined as the set of all vertical edges $e = (x, y) \in E_{\mathsf{v}}$ such that $y \in C_i$ and $x \notin C_i$.
- Let G_i^+ be the subgraph of G induced by all the edges in G_i together with the edge set $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_i)$. Let F_i^+ be the central face of G_i^+, and let C_i^+ be the facial cycle of F_i^+.

For each vertical edge $e = (x, y) \in \mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_i)$, the south endpoint x appears exactly once in C_i^+. We circularly order the edges $e = (x, y) \in \mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_i)$ according to the position of the south endpoint x in the circular ordering of C_i^+. Take the graph $G = G_6$ in Figure 6 as an example. In this graph, there are 6 horizontal segments, shaded in Figure 6:

$$S_1 = (v_{1,1}, v_{1,2}, v_{1,3}, v_{1,4}), \quad S_2 = (v_{2,1}, v_{2,2}, v_{2,3}), \quad S_3 = (v_{3,1}, v_{3,2}, v_{3,3}, v_{3,4}, v_{3,5}),$$
$$S_4 = (v_{4,1}, v_{4,2}, v_{4,3}), \quad S_5 = (v_{5,1}, v_{5,2}, v_{5,3}, v_{5,4}, v_{5,5}), \quad S_6 = (v_{6,1}, v_{6,2}).$$

With respect to the sequence $A = (S_1, S_2, \ldots, S_6)$, Figure 6 shows the graphs G_i and G_i^+, for all $1 \le i \le 6$. For example, for $i = 2$, we have:

$$\mathcal{N}_{\mathsf{south}}(S_1, S_2) = ((v_{3,1}, v_{1,1})(v_{3,2}, v_{2,1}), (v_{3,4}, v_{2,3}), (v_{3,5}, v_{1,4})),$$
$$\mathcal{N}_{\mathsf{north}}(S_2) = ((v_{2,1}, v_{1,2}), (v_{2,2}, v_{1,3}))$$
$$C_2 = (v_{1,1}, v_{1,2}, v_{2,1}, v_{2,2}, v_{2,3}, v_{2,2}, v_{1,3}, v_{1,4}),$$
$$C_2^+ = (v_{1,1}, v_{3,1}, v_{1,1}, v_{1,2}, v_{2,1}, v_{3,2}, v_{2,1}, v_{2,2}, v_{2,3}, v_{3,4}, v_{2,3}, v_{2,2}, v_{1,3}, v_{1,4}, v_{3,5}, v_{1,4}).$$

Here $\mathcal{N}_{\mathsf{south}}(S_1, S_2)$, C_2, and C_2^+ are circular orderings, and $\mathcal{N}_{\mathsf{north}}(S_2)$ is a sequential ordering, as S_2 is a path.

Good sequences. We say that a sequence of horizontal segments $A = (S_1, S_2, \ldots, S_k)$ is *good* if A satisfies the following conditions.

(S1) S_1 is the reversal of the facial cycle of the outer face F_{o}, i.e., $S_1 = \overline{C_{F_{\mathsf{o}}}}$.

(S2) For each $1 < i \le k$, $\mathcal{N}_{\mathsf{north}}(S_i)$ satisfies the following requirements.

- $\mathcal{N}_{\mathsf{north}}(S_i) \neq \emptyset$.
- If S_i is a path, then $\mathcal{N}_{\mathsf{north}}(S_i)$ is a contiguous subsequence of $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_{i-1})$.
- If S_i is a cycle, then $\mathcal{N}_{\mathsf{north}}(S_i) = \mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_{i-1})$.

Clearly, if $A = (S_1, S_2, \ldots, S_k)$ is good, then $(S_1, S_2, \ldots, S_i)$ is also good for each $1 \le i < k$. In general, a good sequence that covers the set of all horizontal segments might not exist for a given $(\mathcal{R}, e^\star)$. In particular, in order to satisfy (S1), it is necessary that the cycle $\overline{C_{F_{\mathsf{o}}}}$ is a horizontal segment. The sequence $A = (S_1, S_2, \ldots, S_6)$ shown in Figure 6 is a good sequence.

If $A = (S_1, S_2, \ldots, S_k)$ is good, then we can find a drawing of G_k in linear time by fixing the drawing of $S_1, S_2, \ldots, S_k$ sequentially, as the definition of a good sequence allows us to safely place S_i below $S_1, S_2, \ldots, S_{i-1}$ and above $S_{i+1}, S_{i+2}, \ldots, S_k$. The following lemma is proved formally in the full version of the paper.

▶ **Lemma 7.** *For a given good sequence $A = (S_1, S_2, \ldots, S_k)$, an ortho-radial drawing of G_k without bends can be constructed in time $O\left(\sum_{i=1}^{k} |S_i|\right)$.*

See Figure 6 for an example of a drawing of G_k produced by the algorithm of Lemma 7.

Constructing a good sequence. In order to use Lemma 7 to compute an ortho-radial drawing of $(\mathcal{R}, e^\star)$, we need to find a good sequence $A = (S_1, S_2, \ldots, S_k)$ with $G_k = G$. However, such a good sequence might not exist even if $(\mathcal{R}, e^\star)$ is drawable. We will show that as long as $(\mathcal{R}, e^\star)$ is drawable, we can always add some *virtual edges* to the graph so that such a good sequence exists and can be computed efficiently. The first step of the algorithm is a simple preprocessing step to ensure the following two properties:

- The facial cycle of the outer face is a horizontal segment.
- Each vertex is incident to a horizontal segment.

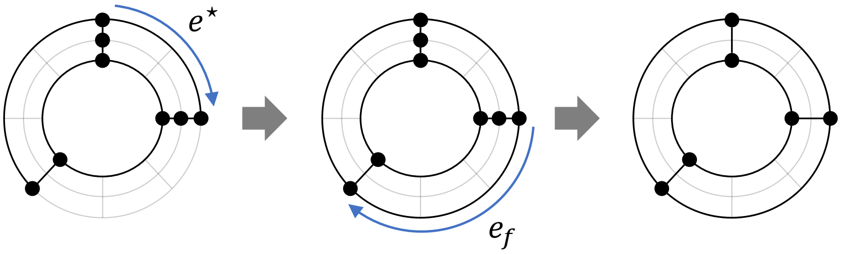


Figure 7 The preprocessing steps.

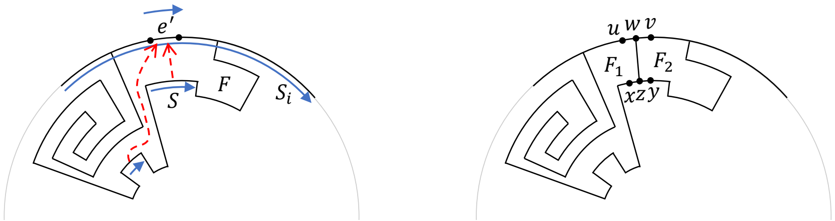


Figure 8 Adding a virtual vertical edge in a regular face.

See Figure 7 for the algorithm of the preprocessing step. The addition of the edge e_f ensures that $\overline{C_{F_\circ}}$ is a horizontal segment. To ensure that each vertex is on a horizontal segment, some degree-2 vertices are removed by smoothing.

The above two properties alone are not sufficient to guarantee the existence of a good sequence $A = (S_1, S_2, \ldots, S_k)$ with $G_k = G$, as there could be horizontal segment S such that $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ and $S \neq \overline{C_{F_\circ}}$. Such a horizontal segment S can never be added to a good sequence, as the definition of a good sequence requires all horizontal segments in the sequence to be non-empty. To deal with this issue, we consider the following eligibility criterion for adding a *virtual* vertical edge incident to such a horizontal segment S:

- Let $A = (S_1, S_2, \ldots, S_k)$ be the current good sequence. Let $S \notin A$ be a horizontal segment such that $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ and $S \neq \overline{C_{F_\circ}}$. Let F be the face such that $\overline{S}$ is a subpath of C_F. We say that S is *eligible* for adding a virtual edge if there exists an edge $e' \in C_F$ with $e' \in S_i$ for some $1 \leq i \leq k$ such that either $\mathsf{rotation}(e' \circ \cdots \circ \overline{S}) = 2$ or $\mathsf{rotation}(\overline{S} \circ \cdots \circ e') = 2$ along the cycle C_F.

See Figure 8 for an illustration of adding a virtual edge. In the figure, there are two horizontal segments along the contour of F that are eligible for adding a virtual edge due to $e' \in S_i$. The rotation criterion for eligibility is to ensure that the new faces created due to the virtual edge still satisfy (R2). The condition $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ ensures that immediately after adding the virtual edge, we may append S to the end of the sequence A.

Our algorithm to construct a good sequence is a simple greedy algorithm: We repeatedly find horizontal segments that can be appended to the current good sequence and repeatedly add virtual edges, until no further such operations can be done. A straightforward implementation of the greedy algorithm, which checks all remaining horizontal segments in each step, takes $O(n^2)$ time. In the full version of the paper, we will present a more efficient implementation that costs only $O(n \log n)$ time.

Extracting a strictly monotone cycle. In the full version of the paper, we prove that if the above greedy algorithm stops with a good sequence $A = (S_1, S_2, \ldots, S_k)$ that does not cover all horizontal segments, then a strictly monotone cycle of the original graph G, without any

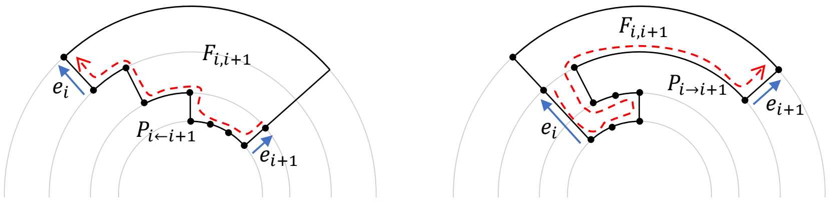


Figure 9 Face types $(*, \sqcup)$ and $(\sqcup, *)$.

virtual edges, can be found to certify the non-existence of a drawing. Let $(e_1, e_2, \ldots, e_s)$ be the circular ordering of $\mathcal{N}_{\mathsf{south}}(A)$. Note that $\{e_1, e_2, \ldots, e_s\}$ is the set of all edges connecting a vertex in G_k and a vertex not in G_k. The proof is achieved by a careful analysis of the structure of the faces involving $\{e_1, e_2, \ldots, e_s\}$. We show that the fact that no more progress can be made in the greedy algorithm forces the parts of the contours of these faces that are not in G_k to form ascending or descending patterns in a consistent manner, so we are able to extract a strictly monotone cycle in G by considering the edges in these facial cycles.

Face types. For each $1 \leq i \leq s$, we write $F_{i,i+1}$ to denote the unique face F such that C_F contains both e_i and $\overline{e_{i+1}}$. Note that $v_{s+1} = v_1$ because $(e_1, e_2, \ldots, e_s)$ is a circular ordering. Consider the face $F_{i,i+1}$, for some $1 \leq i \leq s$. We define $P_{i\leftarrow i+1}$ as the subpath of $C_{F_{i,i+1}}$ starting at $\overline{e_{i+1}}$ and ending at e_i. We write $P_{i\rightarrow i+1} = \overline{P_{i\leftarrow i+1}}$. We write $Z_{i\leftarrow i+1} = (z_1, z_2, \ldots)$ to denote the string of numbers such that z_l is the rotation of the subpath of $P_{i\leftarrow i+1}$ consisting of the first l edges. Similarly, we let $Z_{i\rightarrow i+1} = (z_1, z_2, \ldots)$ be the string of numbers such that z_l is the rotation of the subpath of $P_{i\rightarrow i+1}$ consisting of the first l edges. We define the types $(*, \sqcup)$, $(\sqcup, *)$, $(\sqcup, \sqcup)$, and $(-)$, as follows.

- $F_{i,i+1}$ is of type $(*, \sqcup)$ if $0 \circ 1^c \circ 2$, for some $c \geq 1$, is a prefix of $Z_{i\leftarrow i+1}$.
- $F_{i,i+1}$ is of type $(\sqcup, *)$ if $0 \circ (-1)^c \circ (-2)$, for some $c \geq 1$, is a prefix of $Z_{i\rightarrow i+1}$.
- $F_{i,i+1}$ is of type $(\sqcup, \sqcup)$ if $F_{i,i+1}$ is both of type $(\sqcup, *)$ and of type $(*, \sqcup)$.
- $F_{i,i+1}$ is of type $(-)$ if $Z_{i\leftarrow i+1} = 0 \circ 1^c \circ 2$ for some $c \geq 1$.

In other words, $F_{i,i+1}$ is of type $(-)$ if the subpath of the facial cycle of $F_{i,i+1}$ that connects the south endpoints of e_{i+1} and e_i is a horizontal straight line in the west direction. By considering $P_{i\rightarrow i+1} = \overline{P_{i\leftarrow i+1}}$, equivalently, $F_{i,i+1}$ is of type $(-)$ if $Z_{i\rightarrow i+1} = 0 \circ (-1)^c \circ (-2)$ for some $c \geq 1$.

Consider the good sequence $A = (S_1, S_2)$ of Figure 6 as an example, where we let $\mathcal{N}_{\mathsf{south}}(S_1, S_2) = (e_1, e_2, e_3, e_4)$, where $e_1 = (v_{3,1}, v_{1,1})$, $e_2 = (v_{3,2}, v_{2,1})$, $e_3 = (v_{3,4}, v_{2,3})$, and $e_4 = (v_{3,5}, v_{1,4})$. The facial cycle of the face $F_{1,2}$ is $(v_{3,1}, v_{1,1}, v_{2,1}, v_{3,2})$. We have $P_{1\rightarrow 2} = (v_{1,1}, v_{3,1}, v_{3,2}, v_{2,1})$ and $Z_{1\rightarrow 2} = (0, -1, -2)$, so $F_{1,2}$ is of type $(-)$.

Intuitively, the face $F_{i,i+1}$ is of type $(\sqcup, *)$ if $P_{i\rightarrow i+1}$ makes two 90° left turns before making any right turns, and the first 90° left turn is made at x_i. These two 90° left turns form a $\sqcup$-shape. Similarly, the face $F_{i,i+1}$ is of type $(*, \sqcup)$ if $P_{i\leftarrow i+1}$ makes two 90° right turns before making any left turns, and the first 90° right turn is made at x_{i+1}. These two 90° right turns form a $\sqcup$-shape. See Figure 9 for illustrations of faces of types $(*, \sqcup)$ and $(\sqcup, *)$. In the left part of the figure, we have $Z_{i\leftarrow i+1} = (0, 1, 1, 1, 2, 1, 2, 1, 2)$, so $F_{i,i+1}$ is of type $(*, \sqcup)$. In the right part of the figure, we have $Z_{i\rightarrow i+1} = (0, -1, -1, -2, -3, -3, -2, -1, -2)$, so $F_{i,i+1}$ is of type $(\sqcup, *)$. We show that one of the following holds, which intuitively implies the existence of a strictly monotone cycle.

- All faces $F_{i,i+1}$ are of type $(-)$ and $(\sqcup, *)$, and at least one face $F_{i,i+1}$ is of type $(\sqcup, *)$.
- All faces $F_{i,i+1}$ are of type $(-)$ and $(*, \sqcup)$, and at least one face $F_{i,i+1}$ is of type $(*, \sqcup)$.

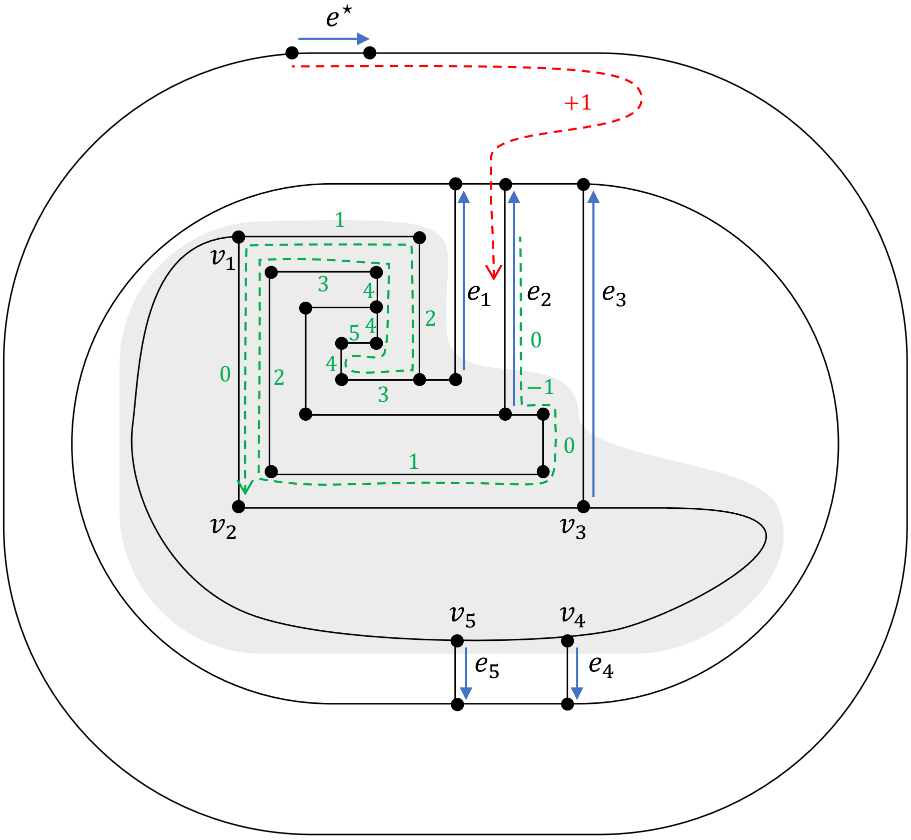


Figure 10 Extracting a strictly monotone cycle $C = (v_1, v_2, \dots, v_5)$.

Consider Figure 10 for an example of extracting a strictly monotone cycle. In the figure, the shaded part corresponds to the part of the graph that is not in G_k. In this example, $\mathcal{N}_{\mathsf{south}}(A) = (e_1, e_2, \dots, e_5)$. The faces $F_{5,1}$, $F_{1,2}$, and $F_{2,3}$ are of type $(*, \sqcup)$. The faces $F_{3,4}$ and $F_{4,5}$ are of type $(-)$. The cycle $C = (v_1, v_2, \dots, v_5)$ is strictly monotone, as it is increasing. We can calculate that $\ell_C((v_1, v_2)) = 1$ by first going from $e^\star$ to $\overline{e_2}$ via a crossing-free path P and then going from $\overline{e_2}$ to (v_1, v_2) along the path $P_{2\to3}$, as (v_1, v_2) is an intermediate edge of $P_{2\to3}$. The first part has rotation 1 and the second part has rotation 0, so the overall rotation is 1. Similarly, we can calculate that $\ell_C(e) = 0$ for each remaining edge e in C.

4 Conclusions

In this paper, we presented a near-linear time algorithm to decide whether a given ortho-radial representation is drawable, improving upon the previous quadratic-time algorithm [35]. If the representation is drawable, then our algorithm outputs an ortho-radial drawing realizing the representation. Otherwise, our algorithm outputs a strictly monotone cycle to certify the non-existence of such a drawing. Given the broad applications of the topology-shape-metric framework in orthogonal drawing, we anticipate that our new ortho-radial drawing algorithm will be relevant and useful in future research in this field.

While there has been extensive research in orthogonal drawing, much remains unknown about the computational complexity of basic optimization problems in ortho-radial drawing. In particular, the problem of finding an ortho-radial representation that minimizes the number of bends has only been addressed by a practical algorithm [34] that has no provable guarantees. It remains an intriguing open question to determine to what extent bend minimization is polynomial-time solvable for ortho-radial drawing. To the best of our knowledge, even deciding whether a given plane graph admits an ortho-radial drawing *without bends* is not known to be polynomial-time solvable.

Given an ortho-radial representation, can we find an ortho-radial drawing with the smallest number of layers (i.e., the number of concentric circles) in polynomial time? As discussed in the full version of the paper, if a good sequence is given, then our algorithm can output a layer-minimized drawing. For the general case where a good sequence might not exist, our algorithm does not have the layer-minimization guarantee, as there is some flexibility in the choice of virtual edges to add, and selecting different virtual edges results in different good sequences. There was a series of work in finding *compact* orthogonal drawings according to various complexity measures [1, 6, 12, 32, 37]. To what extent the ideas developed in these works can be applied to ortho-radial drawings?

References

1 Michael J. Bannister, David Eppstein, and Joseph A. Simons. Inapproximability of orthogonal compaction. *Journal of Graph Algorithms and Applications*, 16(3):651–673, 2012. doi:10.7155/jgaa.00263.

2 Lukas Barth, Benjamin Niedermann, Ignaz Rutter, and Matthias Wolf. Towards a Topology-Shape-Metrics Framework for Ortho-Radial Drawings. In Boris Aronov and Matthew J. Katz, editors, *33rd International Symposium on Computational Geometry (SoCG)*, volume 77 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:16, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2017.14.

3 Lukas Barth, Benjamin Niedermann, Ignaz Rutter, and Matthias Wolf. A topology-shape-metrics framework for ortho-radial graph drawing. *arXiv preprint*, 2021. arXiv:2106.05734v1.

4 Hannah Bast, Patrick Brosi, and Sabine Storandt. Metro maps on flexible base grids. In *17th International Symposium on Spatial and Temporal Databases*, pages 12–22, 2021.

5 Carlo Batini, Enrico Nardelli, and Roberto Tamassia. A layout algorithm for data flow diagrams. *IEEE Transactions on Software Engineering*, SE-12(4):538–546, 1986.

6 Michael A. Bekos, Carla Binucci, Giuseppe Di Battista, Walter Didimo, Martin Gronemann, Karsten Klein, Maurizio Patrignani, and Ignaz Rutter. On turn-regular orthogonal representations. *Journal of Graph Algorithms and Applications*, 26(3):285–306, 2022. doi:10.7155/jgaa.00595.

7 Sandeep N Bhatt and Frank Thomson Leighton. A framework for solving VLSI graph layout problems. *Journal of Computer and System Sciences*, 28(2):300–343, 1984.

8 Therese Biedl, Anna Lubiw, Mark Petrick, and Michael Spriggs. Morphing orthogonal planar graph drawings. *ACM Transactions on Algorithms (TALG)*, 9(4):1–24, 2013.

9 Thomas Bläsius, Ignaz Rutter, and Dorothea Wagner. Optimal orthogonal graph drawing with convex bend costs. *ACM Trans. Algorithms*, 12(3):33:1–33:32, 2016.

10 Ulrik Brandes, Sabine Cornelsen, Christian Fieß, and Dorothea Wagner. How to draw the minimum cuts of a planar graph. *Computational Geometry*, 29(2):117–133, 2004.

11 Ulrik Brandes and Dorothea Wagner. Dynamic grid embedding with few bends and changes. In *International Symposium on Algorithms and Computation*, pages 90–99. Springer, 1998.

12 Stina S Bridgeman, Giuseppe Di Battista, Walter Didimo, Giuseppe Liotta, Roberto Tamassia, and Luca Vismara. Turn-regularity and optimal area drawings of orthogonal representations. *Computational Geometry*, 16(1):53–93, 2000.

13 Yi-Jun Chang and Hsu-Chun Yen. On bend-minimized orthogonal drawings of planar 3-graphs. In *33rd International Symposium on Computational Geometry (SoCG 2017)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

14 Sabine Cornelsen and Andreas Karrenbauer. Accelerated bend minimization. *JGAA*, 16(3):635–650, 2012.

15 Giuseppe Di Battista, Walter Didimo, Maurizio Patrignani, and Maurizio Pizzonia. Orthogonal and quasi-upward drawings with vertices of prescribed size. In *Proceedings of the 7th International Symposium on Graph Drawing (GD)*, pages 297–310. Springer Berlin Heidelberg, 1999.

16 Giuseppe Di Battista, Giuseppe Liotta, and Francesco Vargiu. Spirality and optimal orthogonal drawings. *SIAM Journal on Computing*, 27(6):1764–1811, 1998.

17 Walter Didimo, Giuseppe Liotta, Giacomo Ortali, and Maurizio Patrignani. Optimal orthogonal drawings of planar 3-graphs in linear time. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 806–825. SIAM, 2020.

18 Walter Didimo, Giuseppe Liotta, and Maurizio Patrignani. On the complexity of HV-rectilinear planarity testing. In *International Symposium on Graph Drawing (GD)*, pages 343–354. Springer, 2014.

19 Walter Didimo, Giuseppe Liotta, and Maurizio Patrignani. Bend-minimum orthogonal drawings in quadratic time. In *International Symposium on Graph Drawing and Network Visualization (GD)*, pages 481–494. Springer, 2018.

20 Sally Dong, Yu Gao, Gramoz Goranci, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Guanghao Ye. Nested dissection meets ipms: Planar min-cost flow in nearly-linear time. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 124–153. SIAM, 2022.

21 Stephane Durocher, Stefan Felsner, Saeed Mehrabi, and Debajyoti Mondal. Drawing HV-restricted planar graphs. In *Latin American Symposium on Theoretical Informatics (LATIN)*, pages 156–167. Springer, 2014.

22 Markus Eiglsperger, Carsten Gutwenger, Michael Kaufmann, Joachim Kupke, Michael Jünger, Sebastian Leipert, Karsten Klein, Petra Mutzel, and Martin Siebenhaller. Automatic layout of UML class diagrams in orthogonal style. *Information Visualization*, 3(3):189–208, 2004.

23 Martin Fink, Magnus Lechner, and Alexander Wolff. Concentric metro maps. In *Proceedings of the Schematic Mapping Workshop (SMW)*, 2014.

24 Michael Formann, Torben Hagerup, James Haralambides, Michael Kaufmann, Frank Thomson Leighton, Antonios Symvonis, Emo Welzl, and G Woeginger. Drawing graphs in the plane with high resolution. *SIAM Journal on Computing*, 22(5):1035–1052, 1993.

25 Ashim Garg and Roberto Tamassia. A new minimum cost flow algorithm with applications to graph drawing. In *Proceedings of the Symposium on Graph Drawing (GD)*, pages 201–216. Springer Berlin Heidelberg, 1997.

26 Carsten Gutwenger, Michael Jünger, Karsten Klein, Joachim Kupke, Sebastian Leipert, and Petra Mutzel. A new approach for visualizing UML class diagrams. In *Proceedings of the 2003 ACM symposium on Software visualization*, pages 179–188, 2003.

27 Mahdieh Hasheminezhad, S Mehdi Hashemi, Brendan D McKay, and Maryam Tahmasbi. Rectangular-radial drawings of cubic plane graphs. *Computational Geometry*, 43(9):767–780, 2010.

28 Mahdieh Hasheminezhad, S Mehdi Hashemi, and Maryam Tahmasbi. Ortho-radial drawings of graphs. *Australasian Journal of Combinatorics*, 44:171–182, 2009.

29 Min-Yu Hsueh. *Symbolic layout and compaction of integrated circuits.* PhD thesis, University of California, Berkeley, 1980.

30 Steve Kieffer, Tim Dwyer, Kim Marriott, and Michael Wybrow. Hola: Human-like orthogonal network layout. *IEEE transactions on visualization and computer graphics*, 22(1):349–358, 2015.

31 Gunnar W. Klau and Petra Mutzel. Quasi-orthogonal drawing of planar graphs. Technical Report MPI-I-98-1-013, Max-Planck-Institut für Informatik, Saarbrücken, 1998.

32 Gunnar W Klau and Petra Mutzel. Optimal compaction of orthogonal grid drawings. In *Proceedings of the 7th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 304–319. Springer, 1999.

33 Robin S. Liggett and William J. Mitchell. Optimal space planning in practice. *Computer-Aided Design*, 13(5):277–288, 1981. Special Issue Design optimization. doi:10.1016/0010-4485(81)90317-1.

34 Benjamin Niedermann and Ignaz Rutter. An integer-linear program for bend-minimization in ortho-radial drawings. In *International Symposium on Graph Drawing and Network Visualization*, pages 235–249. Springer, 2020.

35 Benjamin Niedermann, Ignaz Rutter, and Matthias Wolf. Efficient Algorithms for Ortho-Radial Graph Drawing. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry (SoCG)*, volume 129 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 53:1–53:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2019.53.

36 Achilleas Papakostas and Ioannis G Tollis. Efficient orthogonal drawings of high degree graphs. *Algorithmica*, 26(1):100–125, 2000.

37 Maurizio Patrignani. On the complexity of orthogonal compaction. *Computational Geometry*, 19(1):47–67, 2001.

38 James A Storer. The node cost measure for embedding graphs on the planar grid. In *Proceedings of the twelfth annual ACM symposium on Theory of computing*, pages 201–210, 1980.

39 Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16(3):421–444, 1987.

40 Leslie G Valiant. Universality considerations in VLSI circuits. *IEEE Transactions on Computers*, 100(2):135–140, 1981.

41 Hsiang-Yun Wu, Benjamin Niedermann, Shigeo Takahashi, Maxwell J. Roberts, and Martin Nöllenburg. A survey on transit map layout – from design, machine, and human perspectives. *Computer Graphics Forum*, 39(3):619–646, 2020. doi:10.1111/cgf.14030.

42 Yingying Xu, Ho-Yin Chan, and Anthony Chen. Automated generation of concentric circles metro maps using mixed-integer optimization. *International Journal of Geographical Information Science*, pages 1–26, 2022.