Ortho-radial Drawing in Near-linear Time

Anonymous author
Anonymous affiliation

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 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 take it a step further by improving the time complexity of these tasks to near-linear time. We prove a new characterization for ortho-radial drawings based on the concept of a *good sequence*, which enables us to construct an ortho-radial drawing through a simple greedy algorithm.

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.CVIT.2016.23

Acknowledgements Anonymous acknowledgements

1 Introduction

A *plane graph* is a *planar graph* $G = (V, E)$ with a fixed *combinatorial embedding* $\mathcal{E}$, which gives a circular ordering $\mathcal{E}(v)$ of the edges incident to v, for each $v \in V$, to specify the counter-clockwise ordering of these edges surrounding v in a planar embedding. 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 [5, 20], architectural floor plan design [15], and network visualization [4, 9, 12, 14].

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 [6, 7, 8, 18, 19, 11]. In a seminal work [19], Tamassia, the *topology-shape-metric* framework was introduced 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, it was shown in [19] that an orthogonal

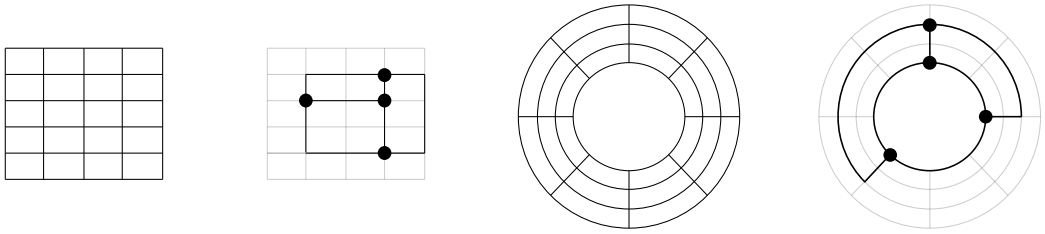


Figure 1 A grid, an orthogonal drawing, a cylindrical grid, and an ortho-radial drawing.

representation can be realized as an orthogonal drawing with zero bends if and only if the following two conditions are satisfied:

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

(O2) The summation 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 (O1) and (O2). Given a valid orthogonal representation, an orthogonal drawing realizing the orthogonal representation can be computed in linear time [13, 19]. 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 the 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 minimum cost flow [19]. Such a reduction to a flow computation is not easy to see if one thinks about orthogonal drawings directly without thinking about orthogonal representations.

1.1 Ortho-radial drawing

Barth, Niedermann, Rutter, and Wolf [1] introduced a generalization of orthogonal drawing to *cylindrical grids*, whose grid-lines consist of concentric circles and straight lines emanating from the center of the circles. 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 do not allow an edge to pass through the origin in the drawing. See Figure 1 for an ortho-radial drawing of K_4 with 2 bends.

The study of ortho-radial drawing is motivated by its applications [3, 10, 22] in network map visualization [21]. 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 [1] 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 summation of angles around each vertex is $360°$.
(R2) The summation 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 summation of angles around each vertex and each face, with one 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 [1] is to show that these three conditions are actually sufficient.

1.2 Previous methods

We briefly review the approaches taken in the previous works [1, 17] in ortho-radial drawing. The proof by Barth, Niedermann, Rutter, and Wolf [1] that (R1)–(R3) are necessary and sufficient is only existential in that it does not yield an efficient algorithm. 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 [17] addressed this gap by showing an $O(n^2)$-time algorithm to decide whether a strictly monotone cycle exists for a given reference edge $e^\star$, and 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 [1] is a reduction to the easier case where each regular face is *rectangular*. For this case, they provided a proof that (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 [1] 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 [1] does not immediately lead to a polynomial-time algorithm.

Quadratic-time algorithms This above issue was addressed in a subsequent work by Niedermann, Rutter, and Wolf [17]. 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 [1] 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 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 [1], the time complexity of the above $O(n^4)$-time drawing algorithm can be lowered to $O(n^2)$ [17]. 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 there is a strictly monotone cycle) and drawing (finding an ortho-radial drawing), the two algorithms in [17] naturally cost $O(n^2)$ time, as they both require performing left-first DFS for $O(n)$ 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. Given an ortho-radial representation satisfying (R1) and (R2) with a fixed reference edge $e^\star$, whether an edge e is a vertical edge (e is drawn as a segment of a straight line passing through the origin) or a horizontal edge (e is drawn as a circular arc of some circle centered on the origin) is determined. 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 [1, 17] 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, ~~to~~ in order to identify new virtual edges to be added, we only need to do some simple local checks such as calculating the summation 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.

1.4 Organization

In Section 2, we discuss the basic graph terminology used in this paper, review some results in previous works [1, 17], 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

In this section, we introduce some basic graph terminology and review some results from the paper [2], which is a merge of the two previous papers [1, 17] on ortho-radial drawing. 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 discuss in Section 7 of the full version of the paper, we may assume that the input graph G is *simple* and *biconnected*.

Paths and cycles Unless otherwise stated, all edges e, paths P, and cycles C are assumed to be directed. We write $\overline{e}$, $\overline{P}$, and $\overline{C}$ to denote the *reversal* of e, P, and C. 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 [2], we say that a path or a cycle is *crossing-free* if it does not contain repeated undirected edges and, for each vertex v that appears multiple times in the path or in the cycle, the ordering of the edges incident to v appearing in the path or the cycle respects the ordering $\mathcal{E}(v)$ or the reversal of $\mathcal{E}(v)$. Intuitively, 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 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 non-simple 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.

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.

I suggest using the same font as in the text.

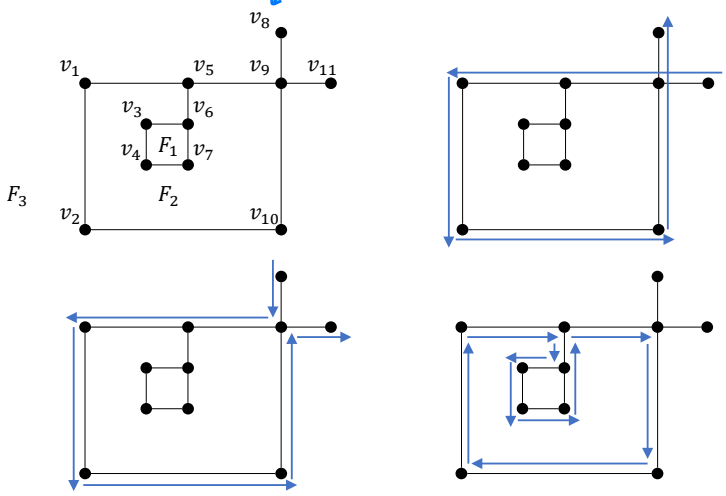


Figure 2 A non-crossing-free path, a crossing-free path, and a facial cycle.

- 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.

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. We also consider the setting where the *reference edge* $e^\star$ is fixed, where there is an additional requirement that the reference edge $e^\star$ has to lie on the outermost circular arc used in the drawing and is along 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 ∘, ∘∘, and ∘ ∘ ∘ to indicate a 90°, a 180°, and a 270° angle assigned to a corner, respectively.

It was shown in [2] 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 length-2 path $P = (u, v, w)$ that passes v such that $u \neq w$. Given the angle assignment ϕ, P is either a 90° left turn, straight line, or a 90° right turn. We define the *combinatorial rotation* of P as follows: $\mathsf{rotation}(P) = -1$ if P is a 90° left turn, $\mathsf{rotation}(P) = 0$ if P is a straight line, and $\mathsf{rotation}(P) = 1$ if P is a 90° right turn.

More formally, consider the two undirected edges $\{u, v\}$ and $\{v, w\}$. 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 undirected edges incident to v. Then the combinatorial rotation of P is defined as $\mathsf{rotation}(P) = \left(\sum_{j=1}^{k-1} \phi(e_j, e_{j+1}) - 180°\right) / 90°$.

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 traverses from u to v along the

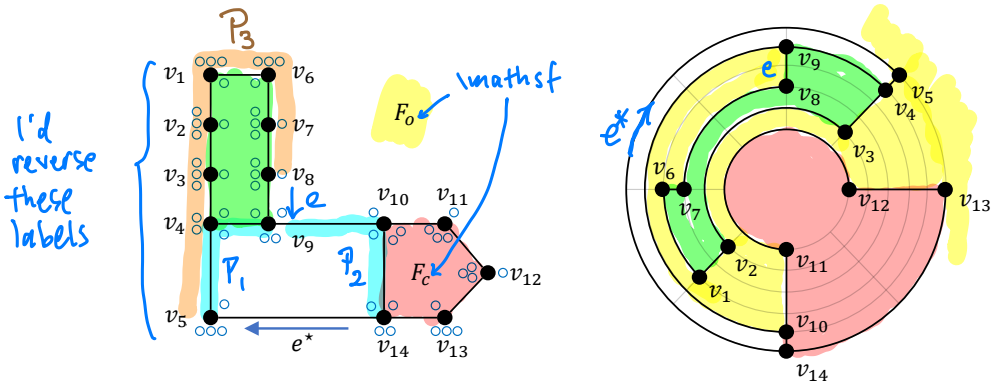


Figure 3 A drawing of an ortho-radial representation with a reference edge.

right side of e and then traverses from v to 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 summation of the combinatorial rotations of all length-2 subpaths of P. Similarly, for a cycle C of length more than 2, we define $\mathsf{rotation}(C)$ as the summation of the combinatorial rotations of all length-2 subpaths of C. Based on this notion, we may restate condition (R2) as follows.

(R2) Each face F satisfies the following: $\mathsf{rotation}(C_F) = 4$ if F is a regular face, $\mathsf{rotation}(C_F) = 0$ if F is either the central face or the outer face, but not both, and $\mathsf{rotation}(C_F) = -4$ if F is both the central face and the outer face.

Consider the ortho-radial representation in Figure 3: 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, cycle $C = (v_{10}, v_{11}, v_{12}, v_{13}, v_{14})$ is the facial cycle of the central face, and it has $\mathsf{rotation}(C) = 0$.

Interior and exterior regions of a cycle Any cycle C partitions the remaining graph into two parts. 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. 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 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 [2], 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 [2].

▶ **Lemma 1** ([2]). *Let C be a simple cycle oriented in such a way that the outer face F_o lies in its exterior, then it satisfies the following: If C is an essential cycle, then $\mathsf{rotation}(C) = 4$; otherwise $\mathsf{rotation}(C) = 0$.*

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 $\mathcal{E}(v) = (e_1, e_2, e_3)$ with $\phi(e_1, e_2) = 90°$, $\phi(e_2, e_3) = 90°$, and $\phi(e_3, e_1) = 180°$ in G. Suppose v only has the two incident 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, and 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.

Consider the subgraph H induced by $\{v_1, v_2, \ldots, v_9\}$ in the ortho-radial representation shown in Figure 3. In H, v_9 has only two incident edges $e_1 = \{v_8, v_9\}$ and $e_2 = \{v_4, v_9\}$, and the angle assignment ϕ_H for the two corners surrounding v_9 in H will be $\phi_H(e_1, e_2) = 90°$ and $\phi_H(e_2, e_1) = 270°$. The outer face and the central face of H are the same.

Measuring directions via reference paths Following [2], 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.

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

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

Consider the reference edge $e = (v_{14}, v_5)$ in Figure 3. We measure the direction of $e' = (v_8, v_9)$ from e with different choices of the reference path P. If $P_1 = (v_5, v_4, v_9)$, then $\mathsf{direction}(e, P, e') = \mathsf{rotation}(e \circ P_1 \circ \overline{e'}) - 2 = -1$. If $P_2 = (v_{14}, v_{10}, v_9)$, then we also have $\mathsf{direction}(e, P, e') = \mathsf{rotation}(\overline{e} \circ P_2 \circ \overline{e'}) = -1$. If we select $P_3 = (v_5, v_4, v_3, v_2, v_1, v_6, v_7, v_8)$, then we get a different value of $\mathsf{direction}(e, P, e') = \mathsf{rotation}(e \circ P_3 \circ e') = 3$. As we will later discuss, $\mathsf{direction}(e, P, e') \bmod 4$ is invariant of 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 the actual path, which is not crossing-free, that we want to consider is $e \circ \overline{e} \circ P \circ e'$, where we make an 180° right turn in $e \circ \overline{e}$, which contributes $+2$ in the calculation of combinatorial rotation. Similarly, the additive -2 in $\mathsf{rotation}(e \circ P \circ \overline{e'}) - 2$ is due to the fact the actual path that we want to consider is $e \circ P \circ \overline{e'} \circ e'$, where we make an 180° left turn in $\overline{e'} \circ e'$. The reason that there is no additive term in $\mathsf{rotation}(\overline{e} \circ P \circ \overline{e'})$ is due to the cancellation of the 180° right turn $e \circ \overline{e}$ and the 180° left turn $\overline{e'} \circ e'$. The reason that $e \circ \overline{e}$ 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

I suggest to use different colors for P, e, $e^\star$.

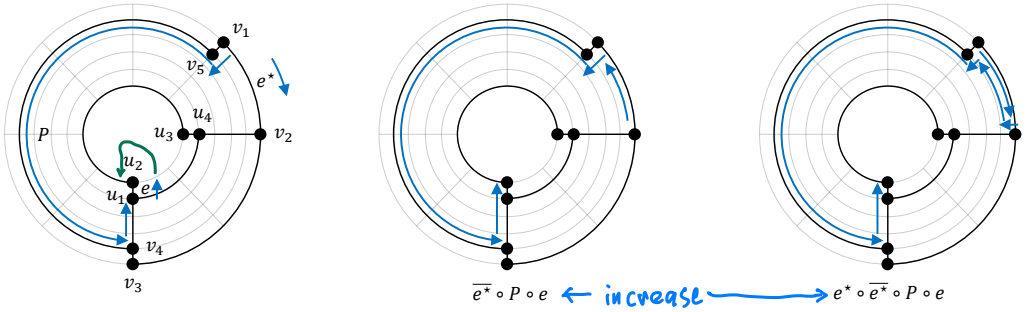


Figure 4 The calculation of $\mathsf{direction}(e^\star, P, e)$.

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 is in the *north* direction if e is drawn as a line segment of a directed straight line emanating from the origin. If e is in the north direction, then $\overline{e}$ is in the *south* direction.
- e is in the *east* direction if e is drawn as a circular arc of a circle centered at the origin in the clockwise direction. If e is in the east direction, then $\overline{e}$ is in the *west* 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 to east, and so $\overline{e^\star}$ points to 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** ([2]). *For any two edges e and e', the value of $\mathsf{direction}(e, P, e') \bmod 4$ is invariant of the choice of the reference path P.*

The above lemma shows that $\mathsf{direction}(e^\star, P, e) \bmod 4$ is invariant of 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 all edges in $\overline{C_{F_\circ}}$ is eligible to be a reference edge. To determine whether an edge $e \in \overline{C_{F_\circ}}$ is eligible to be a reference edge, we need to introduce some terminology.

Given the reference edge $e^\star$, the set of vertical edges in the north direction E_v and the set of horizontal edges in the east direction E_v 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 $S \in \mathcal{S}_{\mathsf{h}}$ a *horizontal segment*.

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 all $1 \leq i < s$. 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 index i of the endpoint v_i of e 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 $S = (v_{10}, v_9, v_4)$ has $\mathcal{N}_{\mathsf{south}}(S) = ((v_{11}, v_{10}), (v_8, v_9), (v_3, v_4))$ and $\mathcal{N}_{\mathsf{north}}(S) = ((v_{10}, v_{14}), (v_4, v_5))$.

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 be checked easily in linear time.

Spirality Intuitively, $\mathsf{direction}(e, P, e')$ quantifies the degree of *spirality* of e' with respect to e and P. Unfortunately, the Lemma 2 does not hold if we replace $\mathsf{direction}(e, P, e')$ mod 4 with $\mathsf{direction}(e, P, e')$. A crucial observation made in [2] 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** ([2]). *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'. Then $\mathsf{direction}(e, P, e')$ is invariant of 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 P can be a general path that is not crossing-free, then we could choose P in such a way that P repeatedly traverse a cycle for many times, so that $\mathsf{direction}(e, P, e')$ can be made arbitrarily large and arbitrarily small.

Setting $e = e^\star$ and $C = \overline{C_{F_\circ}}$ 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 $\overline{C_{F_\circ}}$ is the outermost essential cycle in that all other essential cycles are in the interior of C. The reason that we have to take $\overline{C_{F_\circ}}$ and not $C_{F_\circ}$ is that $F_\circ$ has to be in the exterior of C. Recall that the assumption that G is biconnected ensures that each facial cycle is a simple cycle.

Let C be an essential cycle and let e be an edge in C. In view of the above, following [2], 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 $C = \overline{C_{F_\circ}}$, 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 the actual path that we want to consider is $e \circ \overline{e} \circ P \circ e'$, where we make an 180° right turn in $e \circ \overline{e}$. The reason that $e \circ \overline{e}$ has to be a right turn is due to 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.

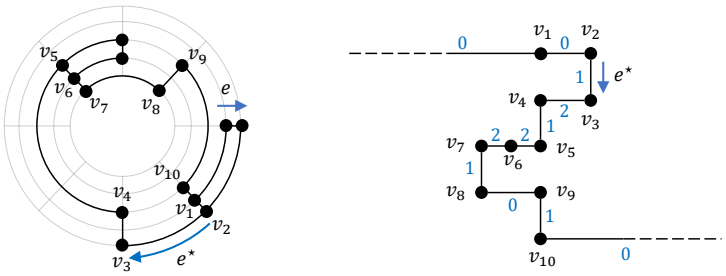


Figure 5 Changing the reference edge to e leads to a strictly monotone cycle.

Monotone cycles We are now ready to define the notion of strictly monotone cycle 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.

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 [2] 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** ([2]). *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 monotone and increasing. With respect to the reference edge e, all the edge labels ~~$\ell_C(e')$~~ 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 a 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 $e^\star$.*

The above theorem improves upon the previous algorithm of [17] that costs $O(n^2)$ time. We also extend the above theorem to the case 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 a biconnected simple graph is drawable~~, and it outputs~~. If $\mathcal{R}$ is drawable, then $\mathcal{A}$ also computes a drawing of $\mathcal{R}$.*

The proofs of Theorems 5 and 6 are in Sections 5 and 6 of the full version of the paper, respectively.

3 Technical overview

Let $A = (S_1, S_2, \ldots, S_k)$ be any sequence of k horizontal segments. In general, we do not require A to cover the set of all horizontal segments in $\mathcal{S}_{\mathsf{h}}$. We consider the following

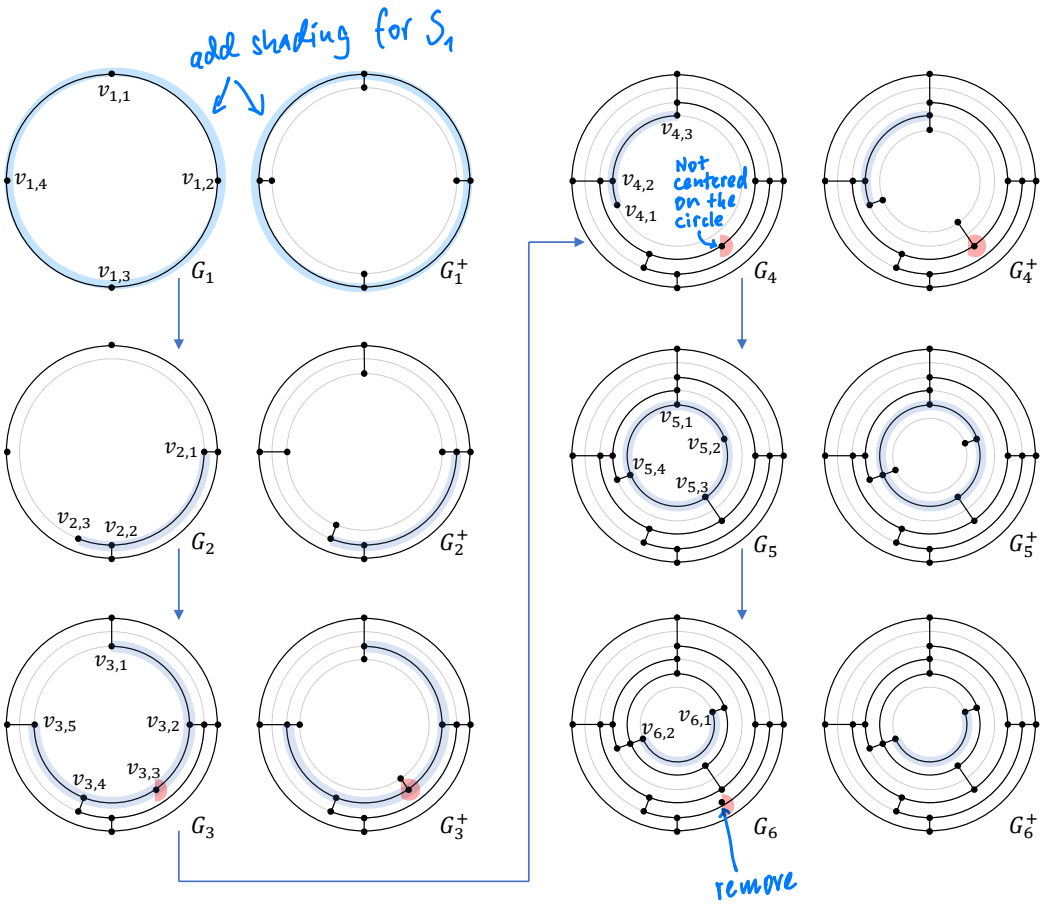


Figure 6 Constructing a good drawing for a good sequence.

terminology for each $1 \leq i \leq k$, where k is the size 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, as follows. Let $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_i)$ be the set of 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^+.

Observe that for each $e = (x, y) \in \mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_i)$, 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:

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

With respect to the sequence $A = (S_1, S_2, \ldots, S_6)$, Figure 6 shows the graphs G_i and G_{i+1}, for all $1 \leq i \leq 6$. 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 satisfies the following conditions.

(S1) $S_1 = \overline{C_{F_o}}$ is the reversal of the facial cycle of the outer face F_o.

(S2) For each $1 < i \leq k$, $\mathcal{N}_{\mathsf{north}}(S_i)$ 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 $\mathcal{N}_{\mathsf{north}}(S_i) = \mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_{i-1})$ have the same circular order.

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 $1 \leq i < k$. In general, a good sequence might not exist for a given $(\mathcal{R}, e^\star)$. In particular, in order to satisfy (S1), it is necessary that the cycle $\overline{C_{F_o}}$ 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 $S_{i+1}, S_{i+2}, \ldots, S_k$. See Section 3 of the full version of the paper for the proof of the following lemma.

▶ **Lemma 7.** *A good drawing of G_k for a given good sequence $A = (S_1, S_2, \ldots, S_k)$ 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.

See Figure 7 for the algorithm of the preprocessing step. These 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 $S \neq \overline{C_{F_o}}$. Such a horizontal segment S can never be added to a good sequence, as the definition of a good sequence requires all horizontal segments S 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:

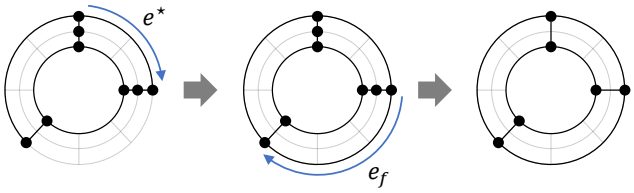
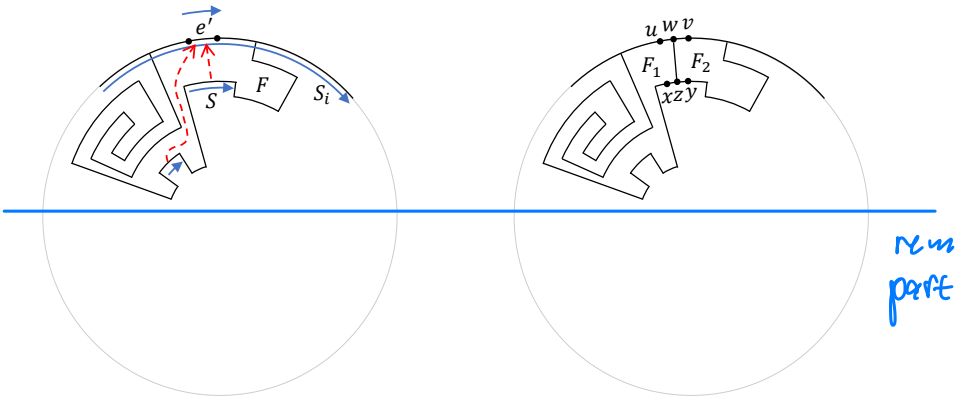


Figure 7 The preprocessing steps.



remove lower part of figure

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

Let $A = (S_1, S_2, \ldots, S_k)$ be the current good sequence. 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 $1 \le i \le k$ 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. ⇝ Similar to criterion for a "kitty corner" in orthogonal drawings; see Bridgeman et al., CGTA'00.

See Figure 8 for an illustration for 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 is to ensure that the new faces created due to the virtual edge still satisfy (R2). 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. We will present a more efficient implementation that requires only $O(n \log n)$ time. See Section 4 of the full version of the paper for details.

clickable hyperlink missing.

Extracting a strictly monotone cycle (we will 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 ~~we will prove that~~ a strictly monotone cycle of the original graph G, without any virtual edges, can be found to certify the non-existence of a drawing. Let $\{e_1, e_2, \ldots, e_s\}$ be 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 will 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 will be able to extract a strictly monotone cycle by considering the edges in these facial cycles. See Figure 9 for an illustration, where the shaded part corresponds to the part of the graph that is not in G_k. In the figure, $C = (v_1, v_2, \ldots, v_5)$ is a strictly monotone cycle,

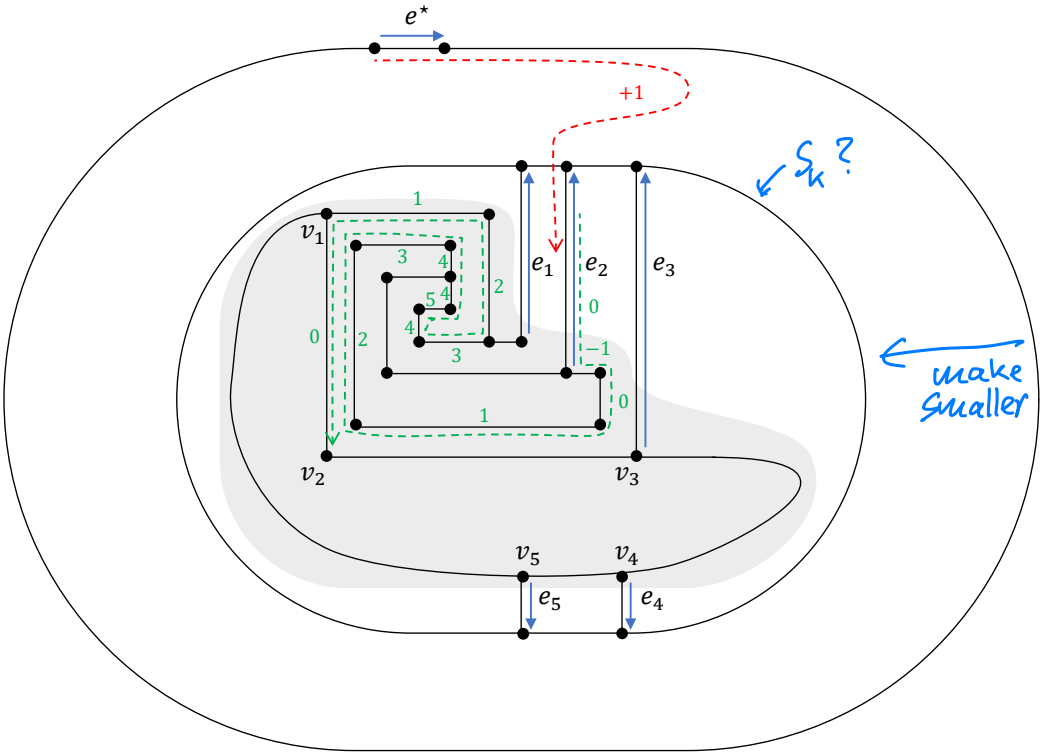


Figure 9 Extracting a strictly monotone cycle $C = (v_1, v_2, \ldots, v_5)$.

Explain the meaning of the red & green dashed paths!

as $\ell_C((v_1, v_2)) = 1$ and $\ell_C(e) = 0$ for each remaining edge e in C. See Section 5 of the full version of the paper for details.

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 [17]. 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. For example, the problem of finding an ortho-radial representation that minimizes the number of bends has only been addressed by a practical algorithm [16] 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.

lemma?

References

1. 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 fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2017/7223`, `doi:10.4230/LIPIcs.SoCG.2017.14`.
2. Lukas Barth, Benjamin Niedermann, Ignaz Rutter, and Matthias Wolf. A topology-shape-metrics framework for ortho-radial graph drawing. *arXiv preprint arXiv:2106.05734v1*, 2021. URL: `https://arxiv.org/abs/2106.05734v1`.
3. 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.
4. 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.
5. 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.
6. Sabine Cornelsen and Andreas Karrenbauer. Accelerated bend minimization. *JGAA*, 16(3):635–650, 2012.
7. Giuseppe Di Battista, Giuseppe Liotta, and Francesco Vargiu. Spirality and optimal orthogonal drawings. *SIAM Journal on Computing*, 27(6):1764–1811, 1998.
8. 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.
9. 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.
10. Martin Fink, Magnus Lechner, and Alexander Wolff. Concentric metro maps. In *Proceedings of the Schematic Mapping Workshop (SMW)*, 2014.
11. 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.
12. 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.
13. Min-Yu Hsueh. *Symbolic layout and compaction of integrated circuits*. PhD thesis, University of California, Berkeley, 1980.
14. 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.
15. Robin S. Liggett and William J. Mitchell. Optimal space planning in practice. *Computer-Aided Design*, 13(5):277–288, 1981. Special Issue Design optimization. URL: `https://www.sciencedirect.com/science/article/pii/0010448581903171`, `doi:https://doi.org/10.1016/0010-4485(81)90317-1`.
16. 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.
17. 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 fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2019/10457`, `doi:10.4230/LIPIcs.SoCG.2019.53`.

18 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.

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

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

21 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. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14030`, `doi:https://doi.org/10.1111/cgf.14030`.

22 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.

Appendix: Full version of the paper

Ortho-radial Drawing in Near-linear Time

Anonymous author
Anonymous affiliation

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 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 take it a step further by improving the time complexity of these tasks to near-linear time. We prove a new characterization for ortho-radial drawings based on the concept of a *good sequence*, which enables us to construct an ortho-radial drawing through a simple greedy algorithm.

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.CVIT.2016.23

Acknowledgements Anonymous acknowledgements

1 Introduction

A *plane graph* is a *planar graph* $G = (V, E)$ with a fixed *combinatorial embedding* $\mathcal{E}$, which gives a circular ordering $\mathcal{E}(v)$ of the edges incident to v, for each $v \in V$, to specify the counter-clockwise ordering of these edges surrounding v in a planar embedding. 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 [5, 36], architectural floor plan design [30], and network visualization [4, 20, 24, 28].

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 [11, 13, 14, 34, 35, 23]. In a seminal work [35] of Tamassia, the *topology-shape-metric* framework was introduced 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

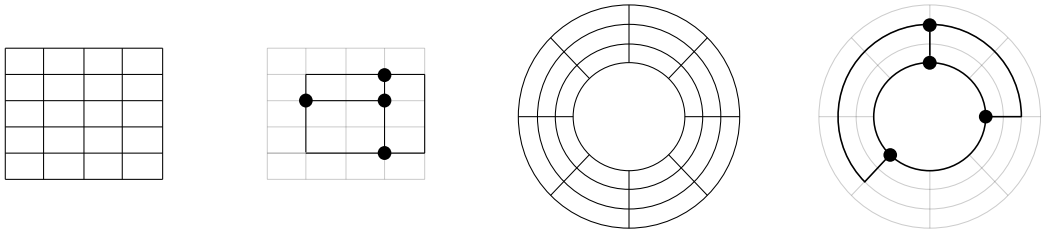


Figure 1 A grid, an orthogonal drawing, a cylindrical grid, and an ortho-radial drawing.

and a designation of the *outer face*. Specifically, it was shown in [35] that an orthogonal representation can be realized as an orthogonal drawing with zero bends if and only if the following two conditions are satisfied:

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

(O2) The summation 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 (O1) and (O2). Given a valid orthogonal representation, an orthogonal drawing realizing the orthogonal representation can be computed in linear time [27, 35]. 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 the 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 minimum cost flow [35]. Such a reduction to a flow computation is not easy to see if one thinks about orthogonal drawings directly without thinking about orthogonal representations.

1.1 Ortho-radial drawing

Barth, Niedermann, Rutter, and Wolf [1] introduced 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. 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 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 2 bends.

The study of ortho-radial drawing is motivated by its applications [3, 21, 38] in network map visualization [37]. 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 [1] 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 summation of angles around each vertex is 360°.

(R2) The summation 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 summation of angles around each vertex and each face, with one 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 [1] is to show that these three conditions are actually sufficient.

1.2 Previous methods

We briefly review the approaches taken in the previous works [1, 32] in ortho-radial drawing. The proof by Barth, Niedermann, Rutter, and Wolf [1] that (R1)–(R3) are necessary and sufficient is only existential in that it does not yield an efficient algorithm. 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 [32] addressed this gap by showing an $O(n^2)$-time algorithm to decide whether a strictly monotone cycle exists for a given reference edge $e^\star$, and 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 [1] is a reduction to the easier case where each regular face is *rectangular*. For this case, they provided a proof that (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 [1] is that they need to ensure that the addition of the edges preserve 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 [1] does not immediately lead to a polynomial-time algorithm.

Quadratic-time algorithms This above issue was addressed in a subsequent work by Niedermann, Rutter, and Wolf [32]. 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 [1] 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 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 [1], the time complexity of the above $O(n^4)$-time drawing algorithm can be lowered to $O(n^2)$ [32]. 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 there is a strictly monotone cycle) and drawing (finding an ortho-radial drawing), the two algorithms in [32] naturally cost $O(n^2)$ time, as they both require performing left-first DFS for $O(n)$ 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. Given an ortho-radial representation satisfying (R1) and (R2) with a fixed reference edge $e^\star$, whether an edge e is a vertical edge (e is drawn as a segment of a straight line passing through the origin) or a horizontal edge (e is drawn as a circular arc of some circle centered on the origin) is determined. 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 [1, 32] 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, to find out what are the new virtual edges that can be added, we only need to do some simple local checks such as calculating the summation 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.

1.4 Related work

Orthogonal drawing is a central topic in graph drawing, see [18] for a survey. The bend minimization problem for orthogonal drawings for planar graphs of maximum degree 4 without a fixed combinatorial embedding is NP-hard [22, 23]. If the combinatorial embedding is fixed, the topology-shape-metric framework of Tamassia [35] reduces the bend minimization problem to a min-cost flow computation. The algorithm of Tamassia [35] costs $O(n^2 \log n)$ time. The time complexity was later improved to $O\left(n^{7/4}\sqrt{\log n}\right)$ [23] and then to $O\left(n^{3/2} \log n\right)$ [11]. A recent $O(n \operatorname{poly} \log n)$-time planar min-cost flow algorithm [17] 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 [22, 23] can be bypassed if the first bend on each edge does not incur any cost [7] 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 [13]. After a series of improvements [10, 14, 16], we now know that a bend-minimized orthogonal drawing of a planar graph with maximum degree 3 can be computed in $O(n)$ time [14].

The topology-shape-metric framework [35] 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 [6], allowing vertices with degree greater than 4 [12, 29, 33], restricting the direction of edges [15, 19], drawing cluster graphs [8], and drawing dynamic graphs [9].

The study of ortho-radial drawing by Barth, Niedermann, Rutter, and Wolf [1, 32] extended the topology-shape-metric framework [35] to accommodate cylindrical grids. Before these works [1, 32], a combinatorial characterization of drawable ortho-radial representation is only known for paths, cycles, and theta graphs [26], and for the special case where the graph is 3-regular and each regular face in the ortho-radial representation is a rectangle [25].

1.5 Organization

In Section 2, we discuss the basic graph terminology used in this paper, review some results in previous works [1, 32], and state our main theorems. In Section 3, we introduce the notion of a good sequence and show that its existence implies a simple ortho-radial drawing algorithm. In Section 4, we present a greedy algorithm that adds virtual edges to a given ortho-radial

representation with a fixed reference edge so that a good sequence that covers the entire graph exists and can be computed efficiently. In Section 5, we show that a strictly monotone cycle, which certifies the non-existence of a drawing, exists and can be computed efficiently when the greedy algorithm fails. In Section 6, we show that our results can be extended to the setting where the reference edge is not fixed at the cost of an extra logarithmic factor in the time complexity. In Section 7, we justify our assumption that the input graph is simple and biconnected by showing a reduction from any graph to a biconnected simple graph. We conclude in Section 8 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 Section 7, 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 [2], which is a merge of the two papers [1, 32] on ortho-radial drawing.

Paths and cycles Unless otherwise stated, all edges e, paths P, and cycles C are assumed to be directed. We write $\overline{e}$, $\overline{P}$, and $\overline{C}$ to denote the *reversal* of e, P, and 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 [2], 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.

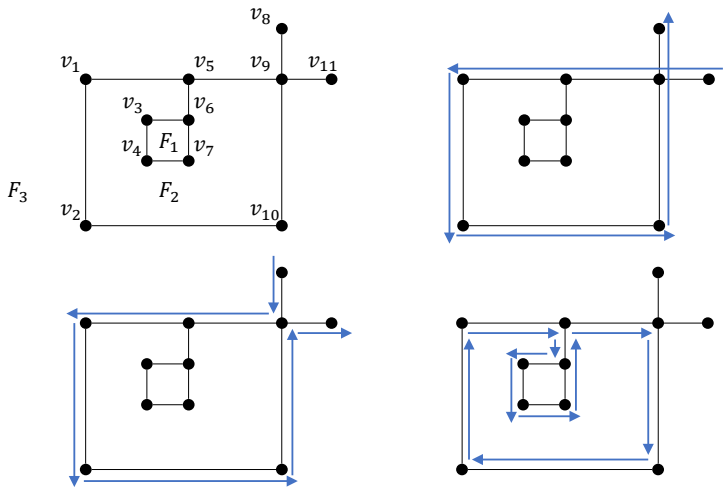


Figure 2 A non-crossing-free path, a crossing-free path, and a facial cycle.

An ortho-radial representation $\mathcal{R} = (\phi, F_c, F_o)$ 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 is along 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 [2] 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 length-2 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, consider the two undirected edges $\{u, v\}$ and $\{v, w\}$. 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 undirected edges incident to v. Then the combinatorial rotation of P is defined as $\mathsf{rotation}(P) = \left(\sum_{j=1}^{k-1} \phi(e_j, e_{j+1}) - 180^\circ\right) / 90^\circ$.

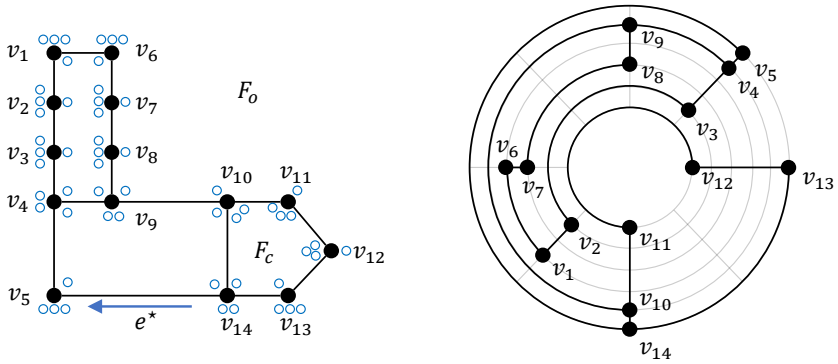


Figure 3 A drawing of an ortho-radial representation with a reference edge, where the small blue circles in the left figure denote the angles in the representation that are realized in the right figure.

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 traverses from u to v along the right side of e and then traverses from v to 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 summation of the combinatorial rotations of all length-2 subpaths of P. Similarly, for a cycle C of length more than 2, we define $\mathsf{rotation}(C)$ as the summation of the combinatorial rotations of all length-2 subpaths of C. Based on this notion, we may restate condition (R2) as follows.

(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). For example, if F is a regular face with k corners, then in the original definition of (R2), it is required that the summation 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. 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 [2], 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 [2].

▶ **Lemma 1** ([2]). *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 $\mathcal{E}(v) = (e_1, e_2, e_3)$ with $\phi(e_1, e_2) = 90°$, $\phi(e_2, e_3) = 90°$, and $\phi(e_3, e_1) = 180°$ in G. Suppose v only has the two incident 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, and 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 $\{v_1, v_2, \ldots, v_9\}$ in the ortho-radial representation shown in Figure 3. In H, v_9 has only two incident edges $e_1 = \{v_8, v_9\}$ and $e_2 = \{v_4, v_9\}$, and the angle assignment ϕ_H for the two corners surrounding v_9 in H will be $\phi_H(e_1, e_2) = 90°$ and $\phi_H(e_2, e_1) = 270°$. The outer face and the central face of H are the same.

Measuring directions via reference paths Following [2], 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 length-1 path. In the definition of $\mathsf{direction}(e, P, e')$, we allow the possibility that a reference

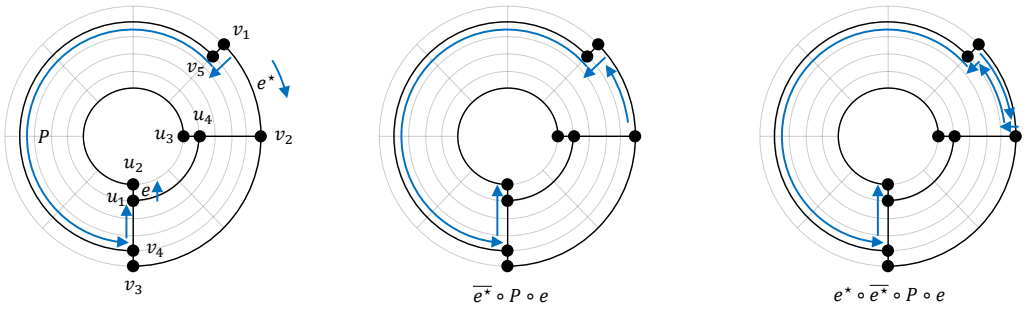


Figure 4 The calculation of $\mathsf{direction}(e^\star, P, e)$.

path P consists of a single vertex. If $v = x$ and $u \neq w$, then we may choose P to be the length-0 path consisting of a single vertex $v = x$, in which case $\mathsf{direction}(e, P, e')$ is simply the combinatorial rotation of the length-2 path (u, v, y). We do not consider the case of $e = e'$ or $e = \overline{e'}$.

Consider the reference edge $e = (v_{14}, v_5)$ 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 $P = (v_5, v_4, v_9)$, 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 $P = (v_5, v_4, v_3, v_2, v_1, v_6, v_7, v_8)$, 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 the actual path, which is not crossing-free, that we want to consider is $e \circ \overline{e} \circ P \circ e'$, where we make an 180° right turn in $e \circ \overline{e}$, which contributes $+2$ in the calculation of combinatorial rotation. Similarly, the additive -2 in $\mathsf{rotation}(e \circ P \circ \overline{e'}) - 2$ is due to the fact the actual path that we want to consider is $e \circ P \circ \overline{e'} \circ e'$, where we make an 180° left turn in $\overline{e'} \circ e'$. The reason that there is no additive term in $\mathsf{rotation}(\overline{e} \circ P \circ \overline{e'})$ is due to the cancellation of the 180° right turn $e \circ \overline{e}$ and the 180° left turn $\overline{e'} \circ e'$. The reason that $e \circ \overline{e}$ 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 from 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 to the east, and so $\overline{e^\star}$ points to the 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** ([2]). *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 all edges in $\overline{C_{F_\circ}}$ are eligible to be a reference edge. To determine whether an edge $e \in \overline{C_{F_\circ}}$ is eligible to be a reference edge, we need to introduce some terminology.

Given the reference edge $e^\star$, the set of vertical edges in the north direction E_v and the set of horizontal edges in the east direction E_h 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 $S \in \mathcal{S}_\mathsf{h}$ a *horizontal segment*.

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 all $1 \le i < s$. 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 e in $\mathcal{N}_\mathsf{north}(S)$ and $\mathcal{N}_\mathsf{south}(S)$ are ordered according to the index i of the endpoint v_i of e 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 $S = (v_{10}, v_9, v_4)$ has $\mathcal{N}_\mathsf{south}(S) = ((v_{11}, v_{10}), (v_8, v_9), (v_3, v_4))$ and $\mathcal{N}_\mathsf{north}(S) = ((v_{10}, v_{14}), (v_4, v_5))$.

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 be checked easily in linear time.

Spirality Intuitively, $\mathsf{direction}(e, P, e')$ quantifies the degree of *spirality* of e' with respect to e and P. Unfortunately, the 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 [2] 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** ([2]). *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 P can be a general path that is not crossing-free, then we could choose P in such a way that P repeatedly traverse a cycle for many times, so that $\mathsf{direction}(e, P, e')$ can be made arbitrarily large and arbitrarily small.

Setting $e = e^\star$ and $C = \overline{C_{F_\circ}}$ 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 $C = \overline{C_{F_\circ}}$ is the outermost essential cycle in that all other essential cycles are in the interior of C. The reason that we have to take $C = \overline{C_{F_\circ}}$ and not $C = C_{F_\circ}$ is that $F_\circ$ has to be in the exterior of C. Recall that the assumption that G is biconnected ensures that each facial cycle is a simple cycle.

Let C be an essential cycle and let e be an edge in C. In view of the above, following [2], 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 $C = \overline{C_{F_\circ}}$, 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 the actual path that we want to consider is $e \circ \overline{e} \circ P \circ e'$, where we make an $180°$ right turn in $e \circ \overline{e}$. The reason that $e \circ \overline{e}$ has to be a right turn is due to 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 cycle 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.

Intuitively, an increasing cycle is like a loop of descending stair, and a decreasing cycle is like a loop of ascending stairs, so they are not drawable. It was proved in [2] 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** ([2]). *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 monotone and increasing. With respect to the reference edge e, all the edge labels $\ell_C(e')$ on the cycle C are non-negative, with some of them being positive.

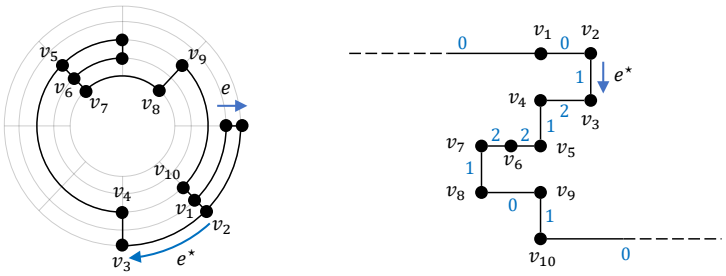


Figure 5 Changing the reference edge to e leads to a strictly monotone cycle.

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 a 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 $e^\star$.*

The above theorem improves upon the previous algorithm of [32] that 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, due to 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 a biconnected simple graph is drawable, and it outputs. If $\mathcal{R}$ is drawable, then $\mathcal{A}$ also computes a drawing of $\mathcal{R}$.*

The proof of Theorem 5 is in Section 5, and the proof of Theorem 6 is in Section 6.

3 Ortho-radial drawings via good sequences

In this section, we introduce the notion of a good sequence, whose existence guarantees that a simple linear-time drawing algorithm.

Sequences of horizontal segments Let $A = (S_1, S_2, \ldots, S_k)$ be any sequence of k horizontal segments. In general, we do not require A to cover the set of all horizontal segments in $\mathcal{S}_{\mathsf{h}}$. We consider the following terminology for each $1 \leq i \leq k$, where k is the size 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, as follows. Let $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_i)$ be the set of 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^+.

Observe that for each $e = (x, y) \in \mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_i)$, 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:

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

With respect to the sequence $A = (S_1, S_2, \ldots, S_6)$, Figure 6 shows the graphs G_i and G_{i+1}, 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 satisfies the following conditions.

(S1) $S_1 = \overline{C_{F_\circ}}$ is the reversal of the facial cycle of the outer face $F_\circ$.

(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 $\mathcal{N}_{\mathsf{north}}(S_i) = \mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_{i-1})$ have the same circular order.

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 might not exist for a given $(\mathcal{R}, e^\star)$. In particular, in order to satisfy (S1), it is necessary that the cycle $\overline{C_{F_\circ}}$ is a horizontal segment. The sequence $A = (S_1, S_2, \ldots, S_6)$ shown in Figure 6 is a good sequence.

Good drawings Throughout the paper, we use the polar coordinate system, where (r, θ) denotes the point given by $x = r\cos\theta$ and $y = r\sin\theta$ in the Cartesian coordinate system. We always have $r \ge 0$. Let $A = (S_1, S_2, \ldots, S_k)$ be a good sequence of k horizontal segments. For notational simplicity, we may also write $\mathcal{N}_{\mathsf{south}}(A) = \mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_k)$. Let $(e_1, e_2, \ldots, e_s)$ be the circular ordering of $\mathcal{N}_{\mathsf{south}}(A)$, and let $e_j = (x_j, y_j)$, for each $1 \le j \le s$, where s is the size of $\mathcal{N}_{\mathsf{south}}(A)$. We say that an ortho-radial drawing of G_k with zero bends is *good* if the drawing satisfies the following properties. Here we let (r_j, θ_j) denote the position of vertex y_j in the drawing.

(D1) For each $1 \le j \le s$, the drawing does not use any point in $\{(r, \theta) \mid 0 \le r < r_j \text{ and } \theta = \theta_j\}$.

(D2) The clockwise circular ordering of $e_1, e_2, \ldots, e_s$ given by $\theta_1, \theta_2, \ldots, \theta_s$ in the drawing is the same as the circular ordering given by $\mathcal{N}_{\mathsf{south}}(A)$.

In Figure 6, the drawing of the graph G_i, for each $1 \le i \le 6$, is a good drawing for the good sequence $(S_1, S_2, \ldots, S_i)$. It is intuitively to see that (D1) implies (D2), and we prove it in the following lemma. We include both (D1) and (D2) in the definition of a good drawing for the sake of convenience.

▶ **Lemma 7.** *If an ortho-radial drawing of G_k for a good sequence $A = (S_1, S_2, \ldots, S_k)$ satisfies (D1), then the drawing also satisfies (D2).*

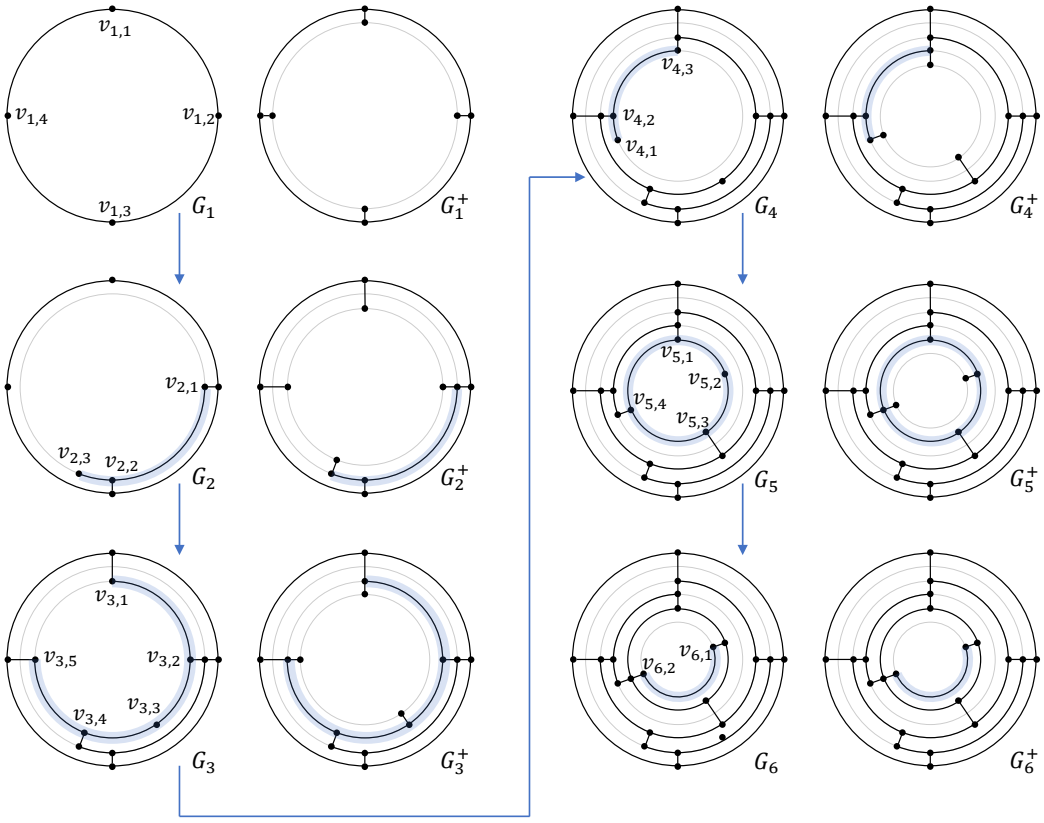


Figure 6 Constructing a good drawing for a good sequence.

Proof. Suppose (D2) is not satisfied. Then we can find three indices a, b, and c such that the clockwise ordering (e_c, e_b, e_a) given by their θ-values is in the opposite direction of their circular ordering (e_a, e_b, e_c) in $\mathcal{N}_{\mathsf{south}}(A) = (e_1, e_2, \ldots, e_s)$.

Let G_k^* be the graph resulting from identifying the south endpoints of all the edges in $\mathcal{N}_{\mathsf{south}}(A) = (e_1, e_2, \ldots, e_s)$ into a vertex v^*. A planar drawing of G_k^* can be found by extending the given ortho-radial drawing of G_k by placing v^* at the origin and drawing all the edges in $\mathcal{N}_{\mathsf{south}}(A) = (e_1, e_2, \ldots, e_s)$ as straight lines. By (D1), the drawing of G_k^* is crossing-free. Assuming that (D2) is not satisfied, we will derive a contradiction by showing that this drawing cannot be crossing-free, so (D2) must be satisfied.

For any $1 \leq i \leq s$ and $1 \leq j \leq s$, we write $P_{i,j}$ to denote the subpath of C_k^+ starting at e_i and ending at $\overline{e_j}$. Any such a path in G_k^* is a cycle, as it starts and ends at the same vertex v^*.

Consider the cycle $P_{a,b}$ in G_k^*. Our assumption on the θ-values for $\{e_a, e_b, e_c\}$ implies that e_c lies in the interior of the cycle $P_{a,b}$ in the above drawing of G_k^*. Now consider the path $P_{c,a}$, which starts at e_c and ends at $\overline{e_a}$. Let v be the first vertex of $P_{a,b} - \{v^*\}$ that $P_{c,a}$ visits. Since $P_{c,a}$ ends at $\overline{e_a}$, such a vertex exists. Let e be the edge incident to v from which $P_{c,a}$ enters v. Since C_k^+ is a facial cycle of G_k^+, the circular ordering of the incident edges of v in C_k^+ must respect the counter-clockwise ordering given by $\mathcal{E}(v)$, so e must be an edge in the exterior of the cycle $P_{a,b}$. Therefore, there exist an edge of $P_{c,a}$ and an edge of $P_{a,b}$ crossing each other, since otherwise $P_{c,a}$ cannot go from the interior of $P_{a,b}$ to the exterior of $P_{a,b}$. See Figure 7 for an illustration. ◀

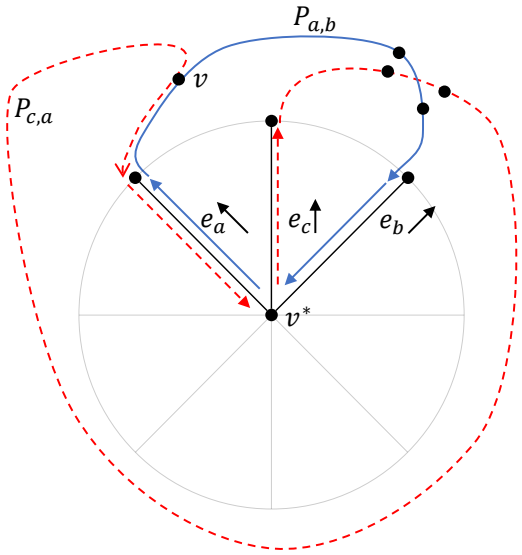


Figure 7 Illustration for the proof of Lemma 7.

We show an efficient algorithm computing a good drawing of G_k for a given good sequence $A = (S_1, S_2, \ldots, S_k)$. The time complexity of the algorithm is linear in the size of G_k. For the special case that $G_k = G$, this gives a linear-time algorithm for computing an ortho-radial drawing realizing the given $(\mathcal{R}, e^\star)$.

▶ **Lemma 8.** *A good drawing of G_k for a given good sequence $A = (S_1, S_2, \ldots, S_k)$ can be constructed in time $O\left(\sum_{i=1}^{k} |S_i|\right)$.*

Proof. The lemma is proved by an induction on the length k of the sequence A. Refer to Figure 6 for an illustration of the algorithm described in the proof.

Base case For the base case of $k = 1$, a good drawing of $G_1 = S_1$ can be constructed, as follows. By (S1), $S_1 = \overline{C_{F_o}}$, which is the outermost essential cycle. Let $S_1 = (v_1, v_2, \ldots, v_t)$, where $t = |S_1|$ is the number of vertices in the cycle S_1. Then we may draw $G_1 = S_1$ on the unit circle by putting v_j on the coordinate $\left(1, -\frac{j}{t} \cdot 2\pi\right)$, for each $1 \leq j \leq t$. The minus sign is due to the fact that S_1 is oriented in the clockwise direction. The construction of the drawing takes $O(|S_1|)$ time as we need to write down these coordinates. Condition (D1) is satisfied because the drawing does not use any point (r, θ) with $0 \leq r < 1$. By Lemma 7, (D2) follows from (D1).

Inductive step For the inductive step, given that we have a good drawing of G_{k-1}, we will extend this drawing to a good drawing of G_k by spending $O(|S_k|)$ time to properly assigning the coordinates to the vertices in S_k. We select $r^* > 0$ to be any number that is smaller than the r-value of the coordinate of all vertices in G_{k-1} in the given drawing, so the circle $r = r^*$ is strictly contained in the central face F_{k-1} of G_{k-1} in the given drawing. Let $S_k = (v_1, v_2, \ldots, v_t)$, where $t = |S_k|$ is the number of vertices in S_k. We will draw S_k on the circle $r = r^*$.

Step 1: vertices with neighbors in the given drawing Let $(e_1, e_2, \ldots, e_s)$ be the circular ordering of $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_{k-1})$, and let $e_j = (x_j, y_j)$, for each $1 \leq j \leq s$, where s is the

size of $\mathcal{N}_{\mathsf{south}}(A)$. We let (r_j, θ_j) denote the position of vertex y_j in the given drawing.

By (S2), $\mathcal{N}_{\mathsf{north}}(S_k)$ is a subset of $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_{k-1})$. For each vertical edge $e_j = (x_j, y_j) \in \mathcal{N}_{\mathsf{north}}(S_k)$, We assign the coordinate (r^*, θ_j) to x_j. By (D1), the given drawing does not use any point in $\{(r, \theta) \mid 0 \leq r < r_j \text{ and } \theta = \theta_j\}$, so we may draw e_j as a straight line connecting x_j and y_j.

By (D2), for the vertices in S_k that we have drawn, that is, the set of vertices in S_k that have incident edges in $\mathcal{N}_{\mathsf{north}}(S_k)$, the clockwise ordering of their θ-values respect the ordering of the horizontal segment $S_k = (v_1, v_2, \ldots, v_t)$. If S_k is a cycle, then $(v_1, v_2, \ldots, v_t)$ is seen as a circular ordering.

Step 2: the two endpoints For the case S_k is a path, we draw the two endpoints v_1 and v_t of $S_k = (v_1, v_2, \ldots, v_t)$, as follows. By (S2), in this case, $\mathcal{N}_{\mathsf{north}}(S_k)$ is a contiguous subsequence of $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_{k-1})$. Let j_1 and j_2 be the indices such that the subsequence starts at e_{j_1} and ends at e_{j_2}. Let $\epsilon = \min_{1 \leq j \leq s}(\theta_j - \theta_{j-1})/3$. If v_1 does not have an incident edge in $\mathcal{N}_{\mathsf{north}}(S_k)$, then we assign the coordinate $(r^*, \theta_{j_1} + \epsilon)$ to v_1. Similarly, if v_t does not have an incident edge in $\mathcal{N}_{\mathsf{north}}(S_k)$, then we assign the coordinate $(r^*, \theta_{j_2} - \epsilon)$ to v_t. Our choice of ϵ ensures that the range $[\theta_{j_2} - \epsilon, \theta_{j_1} + \epsilon]$ of radians does not overlap with θ_j, for any $e_j \in \mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_{k-1}) \setminus \mathcal{N}_{\mathsf{north}}(S_k)$.

Step 3: remaining vertices We draw the remaining vertices of S_k as follows. Let $(v_a, \ldots, v_b)$ be any maximal-length contiguous subsequence of S_k consisting of vertices that have not been drawn yet. We may simply draw them by placing them in between v_{a-1} and v_{b+1} on the circle $r = r^*$. Formally, let θ_{west} be the θ-value of the coordinate of v_{a-1}, and let θ_{east} be the θ-value of the coordinate of v_{b+1}. For each $a \leq j \leq b$, the coordinate of v_j is assigned to be $\left(r^*, \theta_{\mathsf{west}} - (j - a + 1) \cdot \frac{\theta_{\mathsf{east}} - \theta_{\mathsf{west}}}{b - a + 2}\right)$.

In general, it is possible to have $v_{a-1} = v_{b+1}$ when S_k is a cycle and $|\mathcal{N}_{\mathsf{north}}(S_k)| = 1$, in which case $v_{a-1} = v_{b+1}$ is the vertex in S_k incident to the only edge in $\mathcal{N}_{\mathsf{north}}(S_k)$. Note that this case cannot occurs when the underlying graph G is biconnected. For this case, we should let the θ-value of the coordinate of v_{b+1} to be the θ-value of the coordinate of v_{a-1} minus 2π in the above calculation.

Validity of the drawing For the drawing of G_k that we construct, we verify that condition (D1) is satisfied, and then (D2) follows from (D1) by Lemma 7. Consider any vertex v in G_k that has an incident edge in $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_k)$. Suppose that its coordinate in our drawing is (r_v, θ_v). To prove that (D1) is satisfied, we just need to verify that our drawing does not use any point (r, θ) with $0 \leq r < r_v$ and $\theta = \theta_v$. For the case v is in S_k, we have $r_v = r^*$, and our choice of r^* implies that our drawing does not use any point whose r-value is smaller than r^*.

Now suppose v is not in S_k. Then $v = y_j$ for some $e_j = (x_j, y_j) \in \mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_{k-1}) \setminus \mathcal{N}_{\mathsf{north}}(S_k)$. Note that this case is possible only when S_k is a path. By the induction hypothesis, the given drawing of G_{k-1} does not use any point (r, θ) with $0 \leq r < r_v$ and $\theta = \theta_v$, so we just need to verify that when we draw the horizontal segment S_k, the circular arc used to draw S_k does not cross the line $\{(r, \theta) \mid 0 \leq r < r_v \text{ and } \theta = \theta_v\}$. Indeed, the θ-values of this circular arc are confined to the range $[\theta_{j_2} - \epsilon, \theta_{j_1} + \epsilon]$, and as discussed earlier, this range does not overlap with θ_j, for any $e_j \in \mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_{k-1}) \setminus \mathcal{N}_{\mathsf{north}}(S_k)$, so such a crossing is impossible. ◀

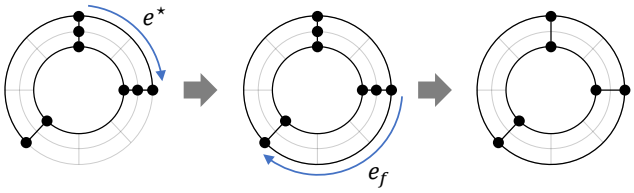


Figure 8 The preprocessing steps.

4 Constructing a good sequence

Given an ortho-radial representation $\mathcal{R}$ of G with a reference edge $e^\star$ such that the horizontal segment $S^\star \in \mathcal{S}_\mathsf{h}$ with $e^\star \in S^\star$ satisfies $\mathcal{N}_\mathsf{north}(S^\star) = \emptyset$, in this section we describe an algorithm that achieves the following. If $(\mathcal{R}, e^\star)$ is drawable, then the algorithm adds virtual edges to $\mathcal{R}$ so that a good sequence $A = (S_1, S_2, \ldots, S_k)$ such that $G_k = G$ exists and can be computed efficiently, and then a drawing of $(\mathcal{R}, e^\star)$ can be constructed using the drawing algorithm in the previous section. If $(\mathcal{R}, e^\star)$ is not drawable, then the algorithm returns a strictly monotone cycle to certify that $(\mathcal{R}, e^\star)$ is not drawable. Recall that we require $e^\star$ to be placed on the outermost circular arc in the drawing of $(\mathcal{R}, e^\star)$. A necessary condition for such a drawing to exist is $\mathcal{N}_\mathsf{north}(S^\star) = \emptyset$.

Preprocessing step 1: the outer face To ensure that a non-empty good sequence exists, by (S1), it is required that $S = \overline{C_{F_\circ}}$ is a horizontal segment in $\mathcal{S}_\mathsf{h}$. If this requirement is not met, then we will add virtual edges to $\mathcal{R}$ to satisfy this requirement. Let $S^\star \in \mathcal{S}_\mathsf{h}$ be the horizontal segment that contains the reference edge $e^\star$, and we have $\mathcal{N}_\mathsf{north}(S^\star) = \emptyset$. We add a virtual horizontal edge e_f that connects the two endpoints of $S^\star$, so $S^\star$ together with e_f becomes the new contour of the outer face and is a horizontal segment in $\mathcal{S}_\mathsf{h}$. See Figure 8 for an illustration.

Observe that the addition of a virtual edge, in general, does not change the value of edge label $\ell_C(e)$ of any edge e in any essential cycle C that already exists in the original graph, as long as the addition of the virtual edge does not destroy (R1) or (R2). The reason is that the calculation of $\ell_C(e)$ is invariant under the choice of the reference path P in the calculation of $\ell_C(e)$, and there is always a reference path P that already exists in the original graph and does not involve any virtual edge.

Preprocessing step 2: smoothing As our goal is to find a good sequence $A = (S_1, S_2, \ldots, S_k)$ such that $G_k = G$, it is necessary that A contains all the horizontal segments in $\mathcal{S}_\mathsf{h}$ and each vertex $v \in V$ is incident to a horizontal segment in $\mathcal{S}_\mathsf{h}$. As we assume that the underlying graph is biconnected, the only possibility that a vertex $v \in V$ is not incident to any horizontal segment is that $\deg(v) = 2$ and v is incident to two vertical edges (u, v) and (v, w). We may get rid of any such a vertex v by *smoothing* it, that is, we replace (u, v) and (v, w) with a single vertical edge (u, w). See Figure 8 for an illustration. It is straightforward to see that smoothing does not affect the drawability of $(\mathcal{R}, e^\star)$, and a drawing of the graph after smoothing can be easily transformed into a drawing of the graph before smoothing. From now on, we assume that each vertex $v \in V$ is incident to a horizontal segment in $\mathcal{S}_\mathsf{h}$, and so all we need to do is to find a good sequence that covers all the horizontal segments.

Eligibility for adding virtual edges Let $S \in \mathcal{S}_\mathsf{h}$ 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 (S2) requires $\mathcal{N}_\mathsf{north}(S)$

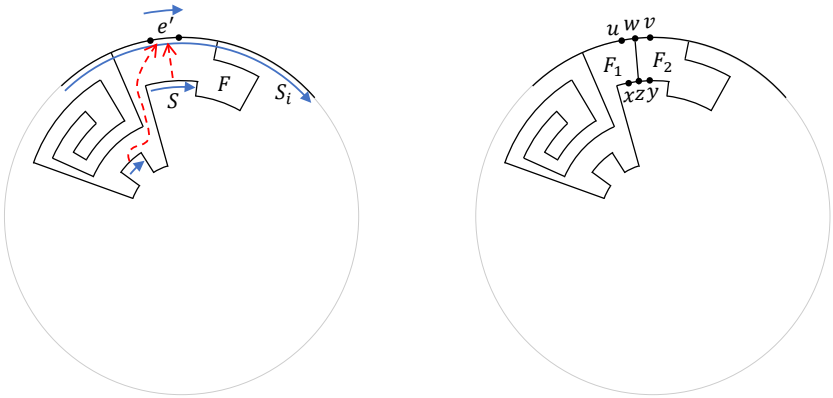


Figure 9 Adding a virtual vertical edge in a regular face.

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 a good sequence. Let $S \notin A$ be a horizontal segment such that $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ and $S \neq \overline{C_{F_o}}$. 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.

For the case that F is a regular face, $\mathsf{rotation}(C_F) = 4$, so $\mathsf{rotation}(e' \circ \cdots \circ \overline{S}) = 2$ if and only if $\mathsf{rotation}(\overline{S} \circ \cdots \circ e') = 2$. We also allow F to be the central face, in which case at most one of $\mathsf{rotation}(e' \circ \cdots \circ \overline{S}) = 2$ and $\mathsf{rotation}(\overline{S} \circ \cdots \circ e') = 2$ can be true.

We argue that if S is eligible for adding a virtual edge with respect to the current good sequence A, then we may add a virtual vertical edge $e_f = (z, w)$ connecting a middle point z of a horizontal edge (x, y) in S and a middle point w of the horizontal edge $e' = (u, v)$, as follows.

$$\begin{array}{ccccc} u & - & v & & \\ & & & \rightarrow & \\ x & - & y & & \end{array} \qquad \begin{array}{ccccc} u & - & w & - & v \\ & & | & & \\ x & - & z & - & y \end{array}$$

See Figure 9 for an illustration. In the figure, F is a regular face, and there are two horizontal segments along the contour of F that are eligible for adding a virtual edge due to $e' \in S_i$.

We argue that the addition of $e_f = (z, w)$ does not destroy (R1) and (R2). The verification of (R1) is straightforward. We verify (R2) for the case $\mathsf{rotation}(\overline{S} \circ \cdots \circ e') = 2$, as the other case $\mathsf{rotation}(e' \circ \cdots \circ \overline{S}) = 2$ is similar. The addition of e_f decompose F into two new faces F_1 and F_2. Let F_1 be the one whose facial cycle contains (u, w, z, x) as a subpath, and let F_2 be the one whose facial cycle contains (y, z, w, v) as a subpath.

We first consider F_1. The rotation of the facial cycle of F_1 equals $\mathsf{rotation}(\overline{S} \circ \cdots \circ e') = 2$ plus the rotation of the path (u, w, z, x), which is also 2, as (u, w, z, x) consists of two right turns. Therefore, the rotation of this facial cycle is 4, meaning that F_1 is a regular face. Now consider the other face F_2. The rotation of the facial cycle of F_2 is identical to $\mathsf{rotation}(C_F)$ before the addition of e_f. The reason is that the rotation of the subpath from y to v is both 2 in C_F and in C_{F_2}, as we assume that $\mathsf{rotation}(\overline{S} \circ \cdots \circ e') = 2$. If F is a regular face, then the summation of rotations is 4 for both F and F_2, so F_2 is also a regular face. If F is a central face, then the summation of rotations is 0 for both F and F_2, so F_2 is also a central face.

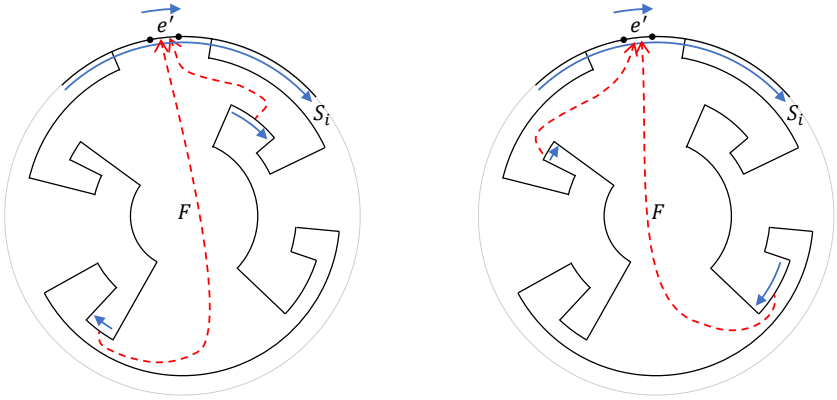


Figure 10 Eligible horizontal segments in the contour of the central face.

Consider Figure 10 as an example: There are four horizontal segments in the contour of the central face F that are eligible for adding a virtual vertical edge due to $e' \in S_i$. The two horizontal segments highlighted in the left part of the figure are eligible due to $\mathsf{rotation}(e' \circ \cdots \circ \overline{S}) = 2$ along the cycle C_F. The two horizontal segments highlighted in the right part of the figure are eligible due to $\mathsf{rotation}(\overline{S} \circ \cdots \circ e') = 2$ along the cycle C_F.

A greedy algorithm Assuming that $C_{F_\circ} \in \mathcal{S}_\mathsf{h}$ and each $v \in V$ is incident to a horizontal segment, our algorithm for constructing a good sequence is as follows. We start with the trivial good sequence $A = (S_1)$, where $S = \overline{C_{F_\circ}}$, and then we repeatedly do the following two operations, until no further such operations can be done.

- Find a horizontal segment $S \in \mathcal{S}_\mathsf{h}$ such that appending S to the end of the current sequence A results in a good sequence, and then extend A by adding S to the end of A.
- Find a horizontal segment $S \in \mathcal{S}_\mathsf{h}$ that is eligible for adding a virtual edge with respect to the current good sequence A, and then add a virtual vertical edge incident to S as discussed above.

There are two possible outcomes of the algorithm. If we obtain a good sequence that covers all horizontal segments $\mathcal{S}_\mathsf{h}$, then we may use Lemma 8 to compute a drawing of $(\mathcal{R}, e^\star)$. Otherwise, the algorithm stops with a good sequence that does not cover all horizontal segments $\mathcal{S}_\mathsf{h}$, and no more progress can be made, in which case in the next section we will show that a strictly monotone cycle can be found.

A straightforward implementation of the greedy algorithm, which checks all horizontal segments in each step, takes $O(n^2)$ time. In the following lemma, we present a more efficient implementation that requires only $O(n \log n)$ time.

▶ **Lemma 9.** *The greedy algorithm can be implemented to run in $O(n \log n)$ time.*

Proof. Let $A = (S_1, S_2, \ldots, S_k)$ denote the current good sequence, which is initialized to an empty set $\emptyset$. During the algorithm, we maintain the circular ordering $\mathcal{N}_\mathsf{south}(S_1, S_2, \ldots, S_k)$ as a circular doubly linked list. Whenever a path $S \in \mathcal{S}_\mathsf{h}$ is inserted to A, this circular doubly linked list is updated by replacing the contiguous subsequence $\mathcal{N}_\mathsf{north}(S)$ of $\mathcal{N}_\mathsf{south}(S_1, S_2, \ldots, S_k)$ with $\mathcal{N}_\mathsf{south}(S)$. Whenever a cycle $S \in \mathcal{S}_\mathsf{h}$ is inserted to A, we have $\mathcal{N}_\mathsf{south}(S_1, S_2, \ldots, S_k) = \mathcal{N}_\mathsf{north}(S)$, so the circular doubly linked list is updated to the circular ordering of $\mathcal{N}_\mathsf{south}(S)$.

Horizontal segments Throughout the algorithm, we maintain a set W containing all horizontal segments $S \in \mathcal{S}_{\mathsf{h}}$ such that inserting S to A results in a good sequence. The set W is initialized as $W = \{\overline{C_{F_\circ}}\}$ and is maintained as follows. First, consider any $S \in \mathcal{S}_{\mathsf{h}}$ with $|\mathcal{N}_{\mathsf{north}}(S)| = 1$. Let $\mathcal{N}_{\mathsf{north}}(S) = \{e\}$. If S is a path, then we add S to W when e is added to $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_k)$. If S is a cycle, then we add S to W when $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_k) = \{e\}$. We note that the case S is a cycle with $|\mathcal{N}_{\mathsf{north}}(S)| = 1$ is not possible when G is biconnected.

Next, consider any $S \in \mathcal{S}_{\mathsf{h}}$ with $|\mathcal{N}_{\mathsf{north}}(S)| \geq 2$. For any two vertical edges e and e' such that e' immediately follows e in the ordering $\mathcal{N}_{\mathsf{north}}(S)$, we maintain an indicator $X_{e,e'} \in \{0, 1\}$ such that $X_{e,e'} = 1$ if e' also immediately follows e in $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_k)$. Initially, all $X_{e,e'}$ are set to 0. For each update to $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_k)$, we check and update $X_{e,e'}$ for all edges e and e' that could be affected. For example, if an edge $\tilde{e}$ is removed from $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_k)$, we check if the two edges immediately preceding and following $\tilde{e}$ in $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_k)$ belong to $\mathcal{N}_{\mathsf{north}}(S)$ for the same horizontal segment S. If the answer is yes, then we set the corresponding indicator $X_{e,e'} = 1$ for S. By (S2), S can be inserted to A if and only if the value of each of its indicators is 1. Therefore, we can decide whether S should join W by checking the summation X_S of all its indicators $X_{e,e'}$. This summation X_S is updated and checked whenever we update the value of an indicator $X_{e,e'}$ for S.

The data structure and algorithm described above costs $O(n)$ time. Each insertion of a horizontal segment S to the good sequence $A = (S_1, S_2, \ldots, S_k)$ incurs a number of insertions and deletions to the circular doubly linked list $\mathcal{N}_{\mathsf{south}}(S_1, S_2, \ldots, S_k)$, and each of these updates gives rise to $O(1)$ operations. The total time complexity is $O(n)$, as the number of updates is linear in $|E_{\mathsf{v}}| = O(n)$.

Virtual edges Next, we consider the task of adding virtual vertical edges. Whenever a horizontal segment $S' \in \mathcal{S}_{\mathsf{h}}$ is inserted to the good sequence $A = (S_1, S_2, \ldots, S_k)$, we check each edge $e' \in S'$ to see if e' causes some horizontal segment $S \in \mathcal{S}_{\mathsf{h}}$ to become eligible for adding virtual edges with respect to $(S_1, S_2, \ldots, S_k, S')$. For all such S, we add a virtual edge e_f incident to S and then add S to W, as the addition of e_f causes the insertion of S to result in a good sequence.

In the subsequent discussion, we fix an $e' \in S'$ and consider the task of finding an $S \in \mathcal{S}_{\mathsf{h}}$ that are eligible for adding virtual edges with respect to $(S_1, S_2, \ldots, S_k, S')$ due to e', if such an S exists. Let F be the face where $e' \in C_F$. We only need to check the set of all $S \in \mathcal{S}_{\mathsf{h}}$ such that $\overline{S}$ is a subpath of C_F and $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$. After finding such an S and adding a virtual edge e_f incident to S, the face F is divided into two faces F_1 and F_2, and the edge e' is also divided into two edges e'_1 and e'_2, where $e'_1 \in C_{F_1}$ and $e'_2 \in C_{F_2}$. We will recursively apply the algorithm for both e'_1 and e'_2. Therefore, by applying the algorithm for all $e' \in S'$, we can ensure that, at the end of this recursive process, no more virtual edges can be added.

A straightforward algorithm of the above task involves checking all $S \in \mathcal{S}_{\mathsf{h}}$ such that $\overline{S}$ is a subpath of C_F and $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$, which costs $O(n)$ time in the worst case. In what follows, we present a more efficient algorithm that costs only $O(\log n)$ time. The number of times we perform this task is $O(n)$, so the total time complexity is $O(n \log n)$.

Regular faces We first focus on the case where the face F with $e' \in C_F$ is a regular face. As discussed earlier, our task is to find a horizontal segment $S \in \mathcal{S}_{\mathsf{h}}$ with $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ such that $\overline{S}$ is a subpath of C_F and 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, if such an S exists. As F is a regular face, $\mathsf{rotation}(C_F) = 4$, so $\mathsf{rotation}(e' \circ \cdots \circ \overline{S}) = 2$ and $\mathsf{rotation}(\overline{S} \circ \cdots \circ e') = 2$ are equivalent.

We write $C_F = (e_1, e_2, \ldots, e_s)$, where s is the number of edges of C_F. To facilitate

the calculation of rotation of a subpath of C_F, we calculate and store the value of $r_i = \mathsf{rotation}(e_1 \circ \cdots \circ e_i)$ for each $1 \le i \le s$. Then the rotation from e_i to e_j is $r_j - r_i$ if $1 \le i \le j \le s$ and is $r_j - r_i + 4$ if $1 \le j < i \le s$. Let i be the index such that $e' = e_i$. In our application, as we want to find S such that $\mathsf{rotation}(e' \circ \cdots \circ \overline{S}) = 2$, we may limit our search space to e_j such that the rotation from e_i to e_j is 2, so we only need to consider e_j such that $r_j = r_i + 2$ or $r_j = r_i - 2$.

Based on the above idea, we organize the set of all $S \in \mathcal{S}_{\mathsf{h}}$ such that $\overline{S}$ is a subpath of C_F and $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ into buckets $B_1, B_2, \ldots$ such that S is added to bucket B_j if the rotation from e_1 to the first edge e_S of $\overline{S}$ is j. Each bucket is stored as an array, sorted according to the indices of e_S in $C_F = (e_1, e_2, \ldots, e_s)$.

Given this data structure, for a given $e' = e_i \in C_F$, we may limit our search space to the two buckets B_{r_i-2} and B_{r_i+2}. For each bucket, we can do a simple $O(1)$-time search to find a desired $S \in \mathcal{S}_{\mathsf{h}}$, if such an S exists. For example, the reason that the bucket B_{r_i-2} is considered is that for the case $1 \le j < i \le s$, the rotation from e_i to e_j is 2 if and only if $r_j - r_i + 4 = 2$, which is equivalent to $r_j = r_i - 2$. Therefore, the search space for this bucket will be any indices within the range $[1, i-1]$, so all we need to do is to check the first element of the array B_{r_i-2} and see if its index lies in the range $[1, i-1]$.

In case such an S is found, as discussed earlier, the face F will be divided into two faces F_1 and F_2. If we rebuild the above data structure for both faces, then the reconstruction costs $O(n)$ time in the worst case, which we cannot afford. The key observation here is that both F_1 and F_2 can be seen as a result of replacing a subpath of F of rotation 2 with a new path of rotation 2, and this new path is irrelevant for the future searches. The portion of F_1 and F_2 inherited from F can be described as a subinterval of the circular ordering $C_F = (e_1, e_2, \ldots, e_s)$. For any two edges $e_{i'}$ and $e_{j'}$ in this subinterval, the rotation from $e_{i'}$ to $e_{j'}$ in a new face can still be computed using the same formula from $r_{i'}$ and $r_{j'}$ defined with respect to the old face F.

In view of the above, instead of rebuilding the data structure for the two new faces F_1 and F_2, we simply store the two subintervals of $C_F = (e_1, e_2, \ldots, e_s)$ corresponding to the F_1 and F_2. If we subsequently need to search for an eligible horizontal segment in F_1 or F_2, we can just use the r-values computed for the old face F to similarly restrict our search space to two buckets of F, and for each bucket we may similarly perform a search to find a desired S, if such an S exists. The only difference is that here the search space will be further constrained by the given subinterval. Searching in a bucket can be done in $O(\log n)$ time using a binary search, as each bucket is sorted.

The two faces F_1 and F_2 can still be partitioned recursively, so there will be multiple faces corresponding to a face F in the original graph G. To implement our approach, for any given $e' \in C_F$, we need to be able to efficiently obtain the subinterval of $C_F = (e_1, e_2, \ldots, e_s)$ corresponding to the current face F' such that $e' \in C_F$. This can be achieved by building a binary search tree to store all these subintervals for each F in the original graph G. Whenever a face is partitioned into two faces, we perform one deletion and two additions to the binary search tree. Whenever we want to query the subinterval for a given $e' \in C_F$, we perform a search in the binary search tree. Each of the operations on the binary search tree costs $O(\log n)$ time.

The central face Now consider the remaining case where the face F with $e' \in C_F$ is the central face. In this case, the two conditions $\mathsf{rotation}(e' \circ \cdots \circ \overline{S}) = 2$ and $\mathsf{rotation}(\overline{S} \circ \cdots \circ e') = 2$ are not equivalent, as $\mathsf{rotation}(C_F) = 0$. However, we can still search for an eligible horizontal segment based on the same approach by considering both two conditions.

Specifically, here we want to find S such that either $\mathsf{rotation}(e' \circ \cdots \circ \overline{S}) = 2$ or $\mathsf{rotation}(\overline{S} \circ \cdots \circ e') = 2$, from the set of all $S \in \mathcal{S}_\mathsf{h}$ with $\mathcal{N}_\mathsf{north}(S) = \emptyset$ such that $\overline{S}$ is a subpath of C_F. Again, we write $C_F = (e_1, e_2, \ldots, e_s)$, let $e_i = e'$, and let e_j be an edge in $\overline{S}$. Then the rotation from e_i to e_j is $r_j - r_i$ for all i and j, so we only need to consider e_j such that $r_j = r_i + 2$ or $r_j = r_i - 2$. Any horizontal segment in the two buckets B_{r_i-2} and B_{r_i+2} are eligible. Of course, same as the case of regular faces, if F is formed using a virtual edge during a recursive call, then we need to restrict the search space to the subinterval corresponding to F, so a binary search is needed.

When a virtual edge is inserted, the face F will be divided into two faces F_1 and F_2. Unlike the case of regular faces, we cannot reuse the r-values for both F_1 and F_2. Here, only one $F^* \in \{F_1, F_2\}$ of the two new faces can be seen as the result of replacing a subpath of F of rotation 2 with a new path of rotation 2, so the rotation of the facial cycle of F^* is still 0, F^* will be the new central face, and the old r-values computed for F can still be used for F^*. For the other new face $F' \in \{F_1, F_2\} \setminus \{F^*\}$, it is the result of replacing a subpath of F of rotation -2 with a new path of rotation 2, so the rotation of the facial cycle of F' will be 4, F' will be a regular face, and the old r-values computed for F cannot be used for F'.

To deal with the above issue, we may simply construct the data structure of F' from scratch, where the time spent is linear in the number of edges in the facial cycle of F'. The total cost for the reconstruction throughout the algorithm is upper bounded by $O(s) = O(n)$, where s is the number of edges in C_{F_c} in the original graph G. ◀

5 Extracting a strictly monotone cycle

In this section, we consider the scenario where the greedy algorithm in the previous section stops with a good sequence $A = (S_1, S_2, \ldots, S_k)$ that does not cover the set of all horizontal segments $\mathcal{S}_\mathsf{h}$, and our goal is to show that in this case a strictly monotone cycle of the original graph G can be computed in $O(n)$ time.

We introduce the terminology that will be used throughout the section. Given a good sequence $A = (S_1, S_2, \ldots, S_k)$ of size k, let $(e_1, e_2, \ldots, e_s)$ be the circular ordering of $\mathcal{N}_\mathsf{south}(A)$, and let $e_j = (x_j, y_j)$, for each $1 \le j \le s$, where s is the size of $\mathcal{N}_\mathsf{south}(A)$. We write $\tilde{G}$ to denote the graph resulting from running the greedy algorithm. That is, $\tilde{G}$ includes all the virtual edges added during the greedy algorithm. Both G_k and G_k^+ are seen as subgraphs of $\tilde{G}$. For each $1 \le i \le s$, we write $F_{i,i+1}$ to denote the unique face F of $\tilde{G}$ 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. Since we assume that G is biconnected, we cannot have $s = |\mathcal{N}_\mathsf{south}(A)| = 1$. We assume that $A = (S_1, S_2, \ldots, S_k)$ and $\tilde{G}$ are the end results of our greedy algorithm in that A cannot be further extended to a longer good sequence and no more virtual edges can be added to $\tilde{G}$.

Face types Consider the face $F_{i,i+1}$, for some $1 \le i \le s$. 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 $c \ge 1$, 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 $c \ge 1$, 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 $c \ge 1$.

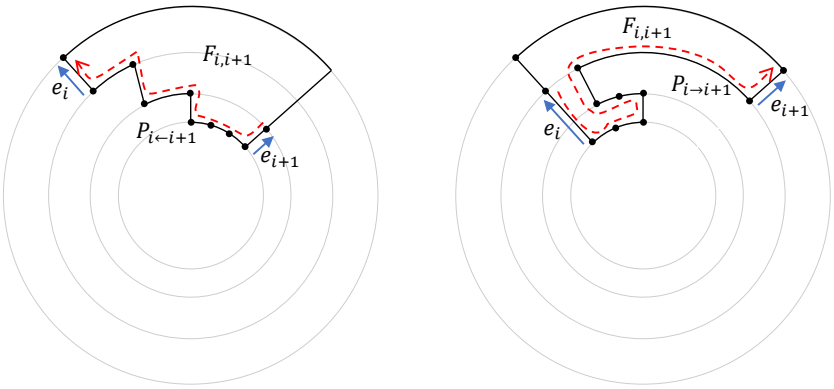


Figure 11 Face types $(*, \sqcup)$ and $(\sqcup, *)$.

In other words, $F_{i,i+1}$ is of type $(-)$ if the subpath $(x_{i+1}, \ldots, x_i)$ of the facial cycle of $F_{i,i+1}$ is a horizontal straight line in the west direction. By considering $P_{i\to i+1} = \overline{P_{i\leftarrow i+1}}$, equivalently, $F_{i,i+1}$ is of type $(-)$ if $Z_{i\to i+1} = 0 \circ (-1)^c \circ (-2)$ 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 $P_{1\to 2} = (v_{1,1}, v_{3,1}, v_{3,2}, v_{2,1})$ and $Z_{1\to 2} = (0, -1, -2)$, so $F_{1,2}$ is of type $(-)$.

Intuitively, the face $F_{i,i+1}$ is of type $(\sqcup, *)$ if $P_{i\to i+1}$ 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 11 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 $Z_{i\to i+1} = (0, -1, -1, -2, -3, -3, -2, -1, -2)$, so $F_{i,i+1}$ is of type $(\sqcup, *)$.

Structural properties We analyze the structural properties of the edges $(e_1, e_2, \ldots, e_s)$ and the their incident faces $F_{i,i+1}$. The following lemma proves the intuitive fact that the rotation from the reference edge $e^\star$ to any $\overline{e_i}$ must be 90° via any crossing-free path P in G_k^+. We note that such a path P must exist.

▶ **Lemma 10.** *Let P be any crossing-free path in G_k^+ starting at the reference edge $e^\star$ and ending at $\overline{e_i}$, for some $1 \leq i \leq s$. Then* $\mathsf{rotation}(P) = 1$.

Proof. Consider the ortho-radial representation $\mathcal{R}'$ resulting from connecting the south endpoints $x_1, x_2, \ldots, x_s$ of the edges $e_1, e_2, \ldots, e_s$ in G_k^+ into a cycle $C = (x_1, x_2, \ldots, x_s)$ in such a way that the rotation from $\overline{e_i}$ to the new edge (x_i, x_{i+1}) is a 90° left turn, the rotation from $\overline{e_i}$ to the new edge (x_i, x_{i-1}) is a 90° right turn, and any subpath of C is a straight line. Observe that C is a horizontal segment and is the facial cycle of the central face of $\mathcal{R}'$. By (D1) and (D2), it is straightforward to convert a good drawing of G_k into an ortho-radial drawing of $\mathcal{R}'$, so $\mathcal{R}'$ is drawable. Since C is a horizontal segment, the edge label $\ell_C(e)$ of all edges $e \in C$ must be the same. Since C cannot be a strictly monotone cycle, the only possibility is that C is a monotone cycle, meaning that $\ell_C(e) = 0$ for all edges $e \in C$. Now consider the path P in the lemma statement and the edge $e = (x_i, x_{i+1})$ in C.

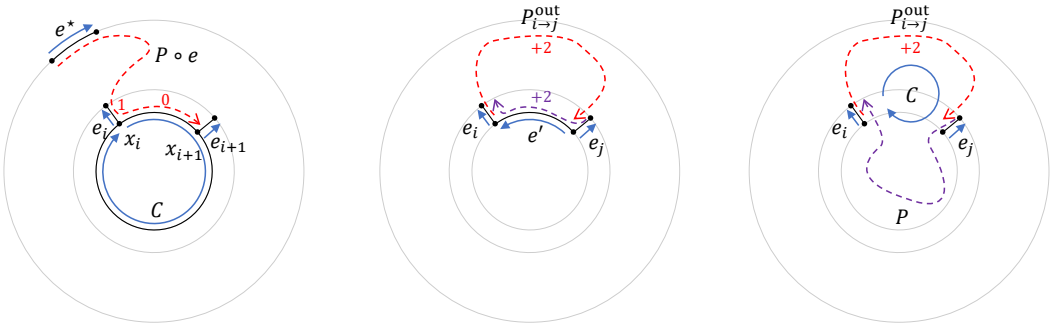


Figure 12 Illustration for the proof of Lemmas 10–12.

Since $\ell_C(e) = 0$, we have $\mathsf{rotation}(P \circ e) = 0$. Since $P \circ e$ makes a 90° left turn at x_i, we have $\mathsf{rotation}(P) = 1$. See the left drawing of Figure 12 for an illustration of the proof. ◀

In the subsequent discussion, we let $P^{\text{out}}_{i\to j}$ denote the subpath of C^+_k starting at e_i and ending at $\overline{e_j}$, for any $1 \le i \le s$ and $1 \le j \le s$. For the special case that $j = i+1$, $P^{\text{out}}_{i\to j}$ is a subpath of the facial cycle of $F_{i,i+1}$. The following lemma proves the intuitive fact that the rotation of $P^{\text{out}}_{i\to j}$ is 2.

▶ **Lemma 11.** *For any $1 \le i \le s$ and $1 \le j \le s$ with $i \ne j$, we have $\mathsf{rotation}(P^{\text{out}}_{i\to j}) = 2$.*

Proof. Consider the ortho-radial representation $\mathcal{R}'$ resulting from connecting the south endpoints x_i and x_j of the edges e_i and e_j in G^+_k into a horizontal edge $e' = (x_j, x_i)$ in such a way that the path $\overline{e_j} \circ e' \circ e_i$ makes two 90° right turns. Similar to the proof of Lemma 10, $\mathcal{R}'$ is drawable by extending a good drawing of G_k. The path $P^{\text{out}}_{i\to j}$ together with the path $\overline{e_j} \circ e' \circ e_i$ forms a facial cycle of a regular face. By (R2), we must have $\mathsf{rotation}(P^{\text{out}}_{i\to j}) = 2$, as the summation of rotations for a regular face has to be 4. See the middle drawing of Figure 12 for an illustration of the proof. ◀

The following lemma considers the rotation of a simple path traversing from $\overline{e_j}$ to e_i that does not use any edges in G_k. The lemma shows that the rotation is either -2 or 2, depending on whether the path, together with $P^{\text{out}}_{i\to j}$, encloses the central face of $\tilde{G}$.

▶ **Lemma 12.** *Let $1 \le i \le s$ and $1 \le j \le s$ with $i \ne j$. Let P be any simple path in $\tilde{G}$ starting at $\overline{e_j}$ and ending at e_i satisfying the following conditions.*

- *P lies in the interior of C^+_k.*
- *P does not contain any vertex in $P^{\text{out}}_{i\to j} \setminus \{x_i, y_i, x_j, y_j\}$.*

Let C be the cycle resulting from combining P with $P^{\text{out}}_{i\to j}$. If the central face lies in the interior of C, then $\mathsf{rotation}(P) = -2$. Otherwise, $\mathsf{rotation}(P) = 2$.

Proof. Consider subgraph H of $\tilde{G}$ induced by G^+_k and all edges in P. Observe that C is a facial cycle of H. If the central face of $\tilde{G}$ lies in the interior of C, then C is the facial cycle of the central face of H, so $\mathsf{rotation}(C) = 0$ by (R2). By Lemma 11, $\mathsf{rotation}(P^{\text{out}}_{i\to j}) = 2$, so we must have $\mathsf{rotation}(P) = -2$. If the central face of $\tilde{G}$ lies in the exterior of C, then C is the facial cycle of a regular face of H, so $\mathsf{rotation}(C) = 4$ by (R2). Therefore, Lemma 11 implies that $\mathsf{rotation}(P) = 2$. See the right drawing of Figure 12 for an illustration of the proof. ◀

The above lemma with $j = i+1$ and $P = P_{i\leftarrow i+1}$ implies that the last element in the sequence of numbers $Z_{i\leftarrow i+1} = (z_1, z_2, \ldots)$ is either -2 or 2, depending on whether $F_{i,i+1}$ is the central face or a regular face. We will later show that $F_{i,i+1}$ must be a regular face.

▶ **Lemma 13.** *Consider the face $F_{i,i+1}$ for any $1 \le i \le s$. The facial cycle C of $F_{i,i+1}$ must contain an edge e' from some horizontal segment S_l in $A = (S_1, S_2, \ldots, S_k)$ such that* $\mathsf{rotation}(e_i \circ \cdots \circ e') = 1$ *and* $\mathsf{rotation}(e' \circ \cdots \circ \overline{e_{i+1}}) = 1$ *along the cycle C.*

Proof. Consider the path $P^{\mathsf{out}}_{i \to i+1}$, which is a subpath of C starting at e_i and ending at $\overline{e_{i+1}}$. By Lemma 11, $\mathsf{rotation}(P^{\mathsf{out}}_{i \to i+1}) = 2$, so there exists an edge e' in $P^{\mathsf{out}}_{i \to i+1}$ such that $\mathsf{rotation}(e_i \circ \cdots \circ e') = 1$ and $\mathsf{rotation}(e' \circ \cdots \circ \overline{e_{i+1}}) = 1$ along the path $P^{\mathsf{out}}_{i \to i+1}$, or equivalently along the cycle C. Since e_i and e_{i+1} are vertical, such an edge e' must be horizontal. Since e' is in G_k, e' belongs to some horizontal segment S_l in $A = (S_1, S_2, \ldots, S_k)$. See Figure 13 for an illustration. ◀

Combining the above lemma with the assumption that no more virtual edges can be added, we show that the strings $Z_{i \leftarrow i+1}$ and $Z_{i \to i+1}$ must satisfy some structural properties.

▶ **Lemma 14.** *Consider any $1 \le i \le s$. If $F_{i,i+1}$ is of type $(*, \sqcup)$, then all numbers in the string $Z_{i \leftarrow i+1}$ are at least 1, except for the first number of the string. If $F_{i,i+1}$ is of type $(\sqcup, *)$, then all numbers in the string $Z_{i \to i+1}$ are at most -1, except for the first number of the string.*

Proof. Our assumption that no more virtual edges can be added, together with Lemma 13, implies that we cannot have a horizontal segment $S \in \mathcal{S}_{\mathsf{h}}$ with $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ satisfying the following conditions:

- S is a subpath of $\overline{C_{F_{i,i+1}}}$.
- For each edge $e \in \overline{S}$, we have $\mathsf{rotation}(\overline{e_{i+1}} \circ \cdots \circ e) = 1$ or $\mathsf{rotation}(e \circ \cdots \circ e_i) = 1$ along the cycle $\overline{C_{F_{i,i+1}}}$.

If such a horizontal segment S with $\mathsf{rotation}(\overline{e_{i+1}} \circ \cdots \circ e) = 1$ for all $e \in \overline{S}$ exists, then S is eligible for adding a virtual edge, due to the edge e' in Lemma 13, as

$$\mathsf{rotation}(e' \circ \cdots \circ e) = \mathsf{rotation}(e' \circ \cdots \circ \overline{e_{i+1}}) + \mathsf{rotation}(\overline{e_{i+1}} \circ \cdots \circ e) = 1 + 1 = 2$$

along the cycle $\overline{C_{F_{i,i+1}}}$. For the remaining case that $\mathsf{rotation}(e \circ \cdots \circ e_i) = 1$ for all $e \in \overline{S}$, for a similar reason, S is also eligible for adding a virtual edge, due to the edge e' in Lemma 13.

Now suppose $F_{i,i+1}$ is of type $(*, \sqcup)$ and some number z_l in the string $Z_{i \leftarrow i+1}$ is 0 and $l \neq 1$. The type $(*, \sqcup)$ guarantees that the string $Z_{i \leftarrow i+1}$ starts with $0 \circ 1^{c'} \circ 2$, for some $c' \ge 1$. Between this number 2 and the above number $z_l = 0$, there must exist a substring $2 \circ 1^c \circ 0$ in $Z_{i \leftarrow i+1}$, for some $c \ge 1$. The reversal of the subpath of $P_{i \leftarrow i+1}$ corresponding to the substring 1^c is a horizontal segment S such that $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ and $\mathsf{rotation}(\overline{e_{i+1}} \circ \cdots \circ e) = 1$ for all $e \in \overline{S}$. Such a horizontal segment S cannot exist, due to the above discussion. Therefore, all numbers in the string $Z_{i \leftarrow i+1}$ must be at least 1, except for the first number, which is always 0. See Figure 13 for an illustration.

The proof for the second statement of the lemma is similar. Suppose $F_{i,i+1}$ is of type $(\sqcup, *)$ and some number z_l in the string $Z_{i \to i+1}$ is at least 0 and $l \neq 1$. Then we can find a substring $(-2) \circ (-1)^c \circ 0$, for some $c \ge 1$, of $Z_{i \to i+1}$, and then we obtain a contradiction, as the horizontal segment corresponding to the substring $(-1)^c$ cannot exist. ◀

Intuitively, if a face $F_{i,i+1}$ is of type $(\sqcup, \sqcup)$, then a $\sqcap$-shape must exist in the middle of $P_{i \leftarrow i+1}$, and the horizontal segment corresponding to the middle part of the $\sqcap$-shape must be eligible for adding a virtual edge, so $F_{i,i+1}$ cannot be of type $(\sqcup, \sqcup)$. In the following lemma, we prove this intuitive observation formally, by combining Lemmas 12 and 14.

▶ **Lemma 15.** *Consider any $1 \le i \le s$. Suppose $F_{i,i+1}$ is of type $(\sqcup, *)$ or $(*, \sqcup)$. Then $F_{i,i+1}$ must be a regular face and $F_{i,i+1}$ cannot be of type $(\sqcup, \sqcup)$.*

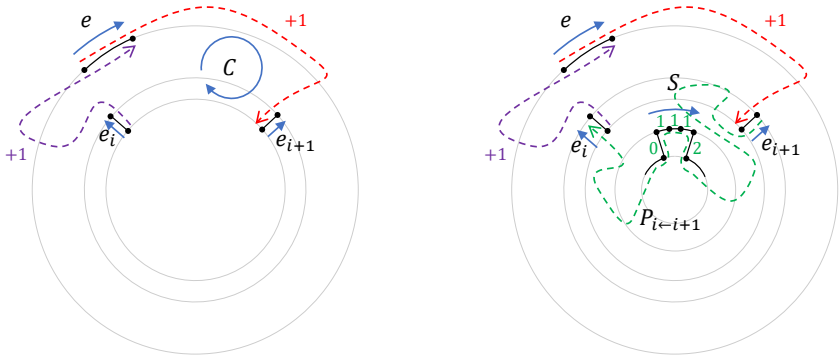


Figure 13 Illustration for the proof of Lemmas 13 and 14.

Proof. By Lemma 12, the rotation of the path $P_{i\leftarrow i+1}$ is -2 if $F_{i,i+1}$ is the central face, and it is 2 if $F_{i,i+1}$ is a regular face. Suppose $F_{i,i+1}$ is of type $(*, \sqcup)$. Then Lemma 14 implies that the rotation of $P_{i\leftarrow i+1}$ is at least 1, so $F_{i,i+1}$ must be a regular face and the rotation of $P_{i\leftarrow i+1}$ is exactly 2, meaning that the string $Z_{i\leftarrow i+1}$ ends with the number 2. As a result, if $F_{i,i+1}$ is also of type $(\sqcup, *)$, then $0 \circ 1^c \circ 2$, for some $c \geq 1$ will be a suffix of $Z_{i\leftarrow i+1}$, violating Lemma 14. Therefore, $F_{i,i+1}$ cannot be of type $(\sqcup, \sqcup)$.

To finish the proof, we just need to show that when $F_{i,i+1}$ is of type $(\sqcup, *)$, $F_{i,i+1}$ also has to be a regular face. Again, Lemma 12 implies that if $F_{i,i+1}$ is the central face, then the rotation of the path $P_{i\rightarrow i+1} = \overline{P_{i\leftarrow i+1}}$ is 2. This contradicts Lemma 14, since it requires the rotation of $P_{i\rightarrow i+1}$ to be at most -1. Therefore, $F_{i,i+1}$ is a regular face. ◀

As discussed in the previous section, we may assume that each vertex in $\tilde{G}$ is incident to a horizontal segment. Consider the horizontal segment S incident to the south endpoint x_i of some $e_i \in \mathcal{N}_{\mathsf{south}}(A)$. In view of (S2), there are two possible reasons for why adding S to the current good sequence A does not result in a good sequence. The first possible reason is that $\mathcal{N}_{\mathsf{north}}(S)$ contains an edge that is not in $\mathcal{N}_{\mathsf{south}}(A)$. The second possible reason is that there exist two edges e and e' such that e' immediately follows e in the ordering of $\mathcal{N}_{\mathsf{north}}(S)$ and e' does not immediately follows e in the ordering of $\mathcal{N}_{\mathsf{south}}(A)$. We show that the second reason is not possible.

▶ **Lemma 16.** *Let S be any horizontal segment in $\tilde{G}$, and let e and e' be any two edges such that e' immediately follows e in the ordering of $\mathcal{N}_{\mathsf{north}}(S)$. If both e and e' are in $\mathcal{N}_{\mathsf{south}}(A)$, then e' also immediately follows e in the circular ordering of $\mathcal{N}_{\mathsf{south}}(A)$.*

Proof. Suppose that the lemma statement does not hold. Then there exist two edges e_i and e_j in $\mathcal{N}_{\mathsf{south}}(A)$ with $j \neq i+1$ and a horizontal segment $S \in \mathcal{S}_{\mathsf{h}}$ such that e_j immediately follows e_i in the ordering of $\mathcal{N}_{\mathsf{north}}(S)$ and all the edges $e_{i+1}, e_{i+2}, \ldots, e_{j-1}$ are not in $\mathcal{N}_{\mathsf{north}}(S)$. Suppose such e_i, e_j, and S exist. We select such e_i, e_j, and S to minimize the number of edges after e_i and before e_j in the circular ordering $\mathcal{N}_{\mathsf{south}}(A)$.

Our choice of e_i, e_j, and S implies that for each horizontal segment S' such that $\mathcal{N}_{\mathsf{north}}(S')$ contains an edge in $(e_{i+1}, e_{i+2}, \ldots, e_{j-1})$, the intersection of $\mathcal{N}_{\mathsf{north}}(S')$ and $\mathcal{N}_{\mathsf{south}}(A)$ must be a contiguous subsequence of $(e_{i+1}, e_{i+2}, \ldots, e_{j-1})$, since otherwise we should select S' and not select S. We partition $(e_{i+1}, e_{i+2}, \ldots, e_{j-1})$ into groups according to the horizontal segment S' incident to the the south endpoint x_l of each edge $e_l = (x_l, y_l)$. In other words, if the south endpoints of two edges of $(e_{i+1}, e_{i+2}, \ldots, e_{j-1})$ are both incident to the same

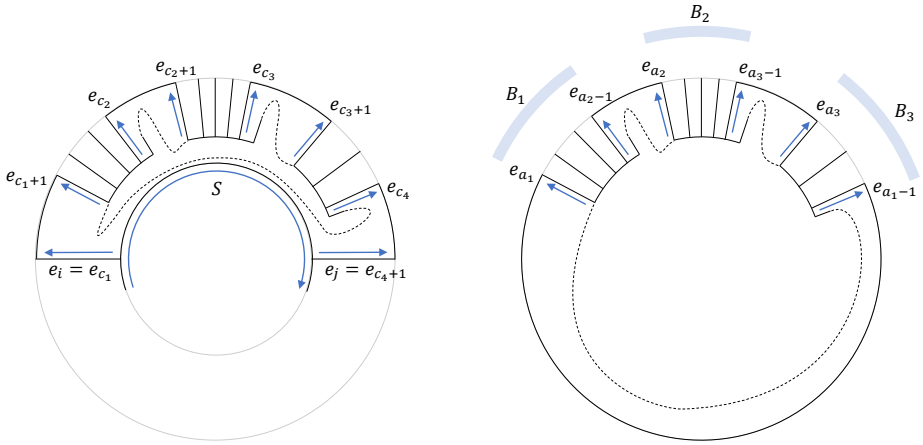


Figure 14 Illustration for the proof of Lemmas 16 and 17.

horizontal segment S', then these two edges are in the same group corresponding to S'. Note that each group corresponds to a contiguous subsequence of $(e_{i+1}, e_{i+2}, \ldots, e_{j-1})$.

Consider a group $(e_a, e_{a+1}, \ldots, e_b)$, and let S' be its corresponding horizontal segment, so the intersection of $\mathcal{N}_{\mathsf{north}}(S')$ and $\mathcal{N}_{\mathsf{south}}(A)$ is precisely the set of edges in the group. Since we assume that adding S' to A does not lead to a good sequence, $\mathcal{N}_{\mathsf{north}}(S')$ must contain some edges that are not in $\mathcal{N}_{\mathsf{south}}(A)$, so either e_a is not the first element of $\mathcal{N}_{\mathsf{north}}(S')$, in which case the face $F_{a-1,a}$ is of type $(*, \sqcup)$, or e_b is not the last element of $\mathcal{N}_{\mathsf{north}}(S')$, in which case the face $F_{b,b+1}$ is of type $(\sqcup, *)$, or both.

We let $i = c_1 < c_2 < \cdots < c_t = j - 1$ be the set of all indices a such that e_a is the last edge of a group or e_{a+1} is the first edge of a group. Our choice of e_i, e_j, and S implies that $F_{i,i+1} = F_{c_1,c_1+1}$ must be of type $(\sqcup, *)$, because the path $P_{i \to i+1}$ will first make a 90° left turn at x_i, go straight along S, and then make another 90° left turn at x_j to enter e_j.

By Lemma 15, $F_{i,i+1} = F_{c_1,c_1+1}$ cannot be also of type $(*, \sqcup)$, as this forces the face to be of type $(\sqcup, \sqcup)$. In view of the discussion above, the fact that F_{c_1,c_1+1} cannot be of type $(*, \sqcup)$ forces the type of F_{c_2,c_2+1} to be $(\sqcup, *)$. Similarly, we can argue that the types of $F_{c_3,c_3+1}, F_{c_4,c_4+1}, \ldots$ must be $(\sqcup, *)$, implying that the type of $F_{c_t,c_t+1} = F_{j-1,j}$ is also $(\sqcup, *)$. The same argument for showing that $F_{i,i+1}$ is of type $(\sqcup, *)$ can also be used to show that $F_{j-1,j}$ must be of type $(*, \sqcup)$. Therefore, $F_{j-1,j}$ is of type $(\sqcup, \sqcup)$, which is impossible due to Lemma 15, so the lemma statement holds. See the left drawing of Figure 14 for an illustration for the proof with $t = 4$. ◀

In view of the above lemma, for each horizontal segment S that is incident to the south endpoint x_i of some e_i, S must be a path, and $\mathcal{N}_{\mathsf{north}}(S)$ must contain an edge e that is not in $\mathcal{N}_{\mathsf{south}}(A)$. Also by the above lemma, such an edge e can only be the first edge or the last edge of the sequential ordering of $\mathcal{N}_{\mathsf{north}}(S)$. If the first edge of $\mathcal{N}_{\mathsf{north}}(S)$ is not in $\mathcal{N}_{\mathsf{south}}(A)$, then the second edge of $\mathcal{N}_{\mathsf{north}}(S)$ must be some edge $e_i \in \mathcal{N}_{\mathsf{south}}(A)$ such that the face $F_{i-1,i}$ is of the type $(*, \sqcup)$. Similarly, if the last edge of $\mathcal{N}_{\mathsf{north}}(S)$ is not in $\mathcal{N}_{\mathsf{south}}(A)$, then the second last edge of $\mathcal{N}_{\mathsf{north}}(S)$ must be some edge $e_i \in \mathcal{N}_{\mathsf{south}}(A)$ such that the face $F_{i,i+1}$ is of the type $(\sqcup, *)$. In the following lemma, we prove that we cannot simultaneously have a face of type $(\sqcup, *)$ and another face of type $(*, \sqcup)$.

▶ **Lemma 17.** *One of the following holds.*

- *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 $(*,\sqcup)$.*

Proof. Similar to the proof of Lemma 16, we partition $\mathcal{N}_{\mathsf{south}}(A) = (e_1, e_2, \ldots, e_s)$ into groups according to the horizontal segment S' incident to the the south endpoint x_l of each edge $e_l = (x_l, y_l)$. By Lemma 16, each group corresponds to a contiguous subsequence of the circular ordering $(e_1, e_2, \ldots, e_s)$.

Let t be the number of groups, and let $B_1, B_2, \ldots, B_t$ denote the contiguous subsequences for these t groups, circularly ordered according to their positions in the circular ordering $(e_1, e_2, \ldots, e_s)$. Let a_i denote the index such that the first edge of B_i is e_{a_i}, so the last edge of B_{i-1} is e_{a_i-1}. See the right drawing of Figure 14 for an illustration with $t = 3$.

Let S' be the horizontal segment corresponding to the group for the contiguous subsequence B_i. As discussed earlier, S' must be a path, and $\mathcal{N}_{\mathsf{north}}(S')$ must contain an edge e that is not in B_i. By Lemma 16, such an edge e can only be the first edge or the last edge of the sequential ordering of $\mathcal{N}_{\mathsf{north}}(S')$. If the first edge of $\mathcal{N}_{\mathsf{north}}(S')$ is not in B_i, then the second edge of $\mathcal{N}_{\mathsf{north}}(S')$ must be e_{a_i}, which is the first edge of B_i. In this case, F_{a_i-1,a_i} is of type $(*,\sqcup)$. Similarly, if the last edge of $\mathcal{N}_{\mathsf{north}}(S')$ is not in B_i, then the second last edge of $\mathcal{N}_{\mathsf{north}}(S')$ must be $e_{a_{i+1}-1}$, which is the last edge of B_i. In this case, $F_{a_{i+1}-1,a_{i+1}}$ is of type $(\sqcup,*)$.

By Lemma 15, the face F_{a_i-1,a_i} cannot be of type $(\sqcup,\sqcup)$, for each $1 \le i \le t$. Therefore, the only possibility is that they are all of type $(*,\sqcup)$ or they are all of type $(\sqcup,*)$.

Now consider any face $F_{l,l+1}$ that is not of the form F_{a_i-1,a_i} for some $1 \le i \le t$. That is, e_l and e_{l+1} are in the same group, meaning that their south endpoints are both incident to the same horizontal segment S', and e_{l+1} immediately follows e_l in the sequential ordering $\mathcal{N}_{\mathsf{north}}(S')$, so the face $F_{l,l+1}$ must be of type $(-)$. ◀

Informally, the above lemma, together with Lemma 14, implies that either all of $P_{i\to i+1}$ are monotonically ascending or all of of $P_{i\leftarrow i+1}$ are monotonically ascending, so we should be able to extract a strictly monotone cycle by considering the edges in these paths. Before we do that, we first show that all faces $F_{i,i+1}$ are distinct regular faces.

▶ **Lemma 18.** *For each $1 \le i \le s$, $F_{i,i+1}$ is a regular face.*

Proof. If $F_{i,i+1}$ is of type $(\sqcup,*)$ or $(\sqcup,*)$, then Lemma 15 implies that $F_{i,i+1}$ is a regular face. By Lemma 16, the only remaining case is when $F_{i,i+1}$ is of type $(-)$. Suppose $F_{i,i+1}$ is of type $(-)$, and let C be the facial cycle of $F_{i,i+1}$. Then $\mathsf{rotation}(C)$ equals the summation of $\mathsf{rotation}(P^{\mathsf{out}}_{i\to i+1})$ and $\mathsf{rotation}(P_{i\leftarrow i+1})$. By the definition of the type $(-)$, we know that $\mathsf{rotation}(P_{i\leftarrow i+1}) = 2$. By Lemma 11, we know that $\mathsf{rotation}(P^{\mathsf{out}}_{i\to i+1}) = 2$. Therefore, $\mathsf{rotation}(C) = 4$, so C is a regular face in view of (R2). ◀

▶ **Lemma 19.** *Any two faces $F_{i,i+1}$ and $F_{i',i'+1}$ with $i \neq i'$ must be distinct.*

Proof. Suppose $F_{i,i+1}$ and $F_{i',i'+1}$ are the same face F. Let C be the facial cycle of F. We know that both $P^{\mathsf{out}}_{i\to i+1}$, which starts at e_i and ends at $\overline{e_{i+1}}$, and $P^{\mathsf{out}}_{i'\to i'+1}$, which starts at $e_{i'}$ and ends at $\overline{e_{i'+1}}$, are subpaths of C. We may assume that $e_{i+1} \neq e_{i'}$ and $e_{i'+1} \neq e_i$, since otherwise C is not a simple cycle, implying that the underlying graph is not biconnected.

We define P_1 as the subpath of C starting at $\overline{e_{i'+1}}$ and ending at e_i and define P_2 as the subpath of C starting at $\overline{e_{i+1}}$ and ending at $e_{i'}$. Therefore, $\mathsf{rotation}(C)$ is the summation of the rotation of the four paths $P^{\mathsf{out}}_{i\to i+1}$, $P^{\mathsf{out}}_{i'\to i'+1}$, P_1, and P_2. By Lemma 18, F is a regular face, so $\mathsf{rotation}(C) = 4$ due to (R2).

The two paths $\overline{P_1}$ and $P^{\mathsf{out}}_{i'+1\to i}$ form a cycle. The two paths $\overline{P_2}$ and $P^{\mathsf{out}}_{i+1\to i'}$ form another cycle. Observe that the central face of $\tilde{G}$ lies in the interior of one of these two cycles. By

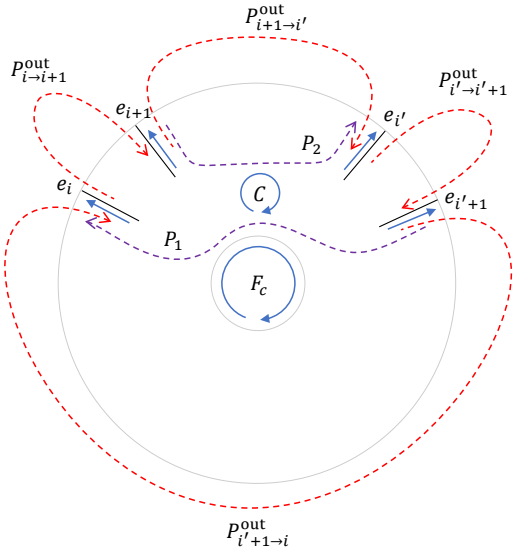


Figure 15 Illustration for the proof of Lemma 19.

symmetry, we may assume the central face lies in the interior of the cycle formed by $\overline{P_1}$ and $P^{\text{out}}_{i'+1\to i}$. In case the central face lies in the interior of the other cycle, we swap i and i'. See Figure 15 for an illustration.

By Lemma 12 with $P = \overline{P_2}$ and $P^{\text{out}}_{i+1\to i'}$, we obtain that $\mathsf{rotation}(\overline{P_2}) = 2$. Similarly, by Lemma 12 with $P = \overline{P_1}$ and $P^{\text{out}}_{i'+1\to i}$, we obtain that $\mathsf{rotation}(\overline{P_1}) = -2$. Therefore, we have $\mathsf{rotation}(P_1) = 2$ and $\mathsf{rotation}(P_2) = -2$. There is one subtle issue in applying Lemma 12: P_2 might contain vertices in $P^{\text{out}}_{i+1\to i'} \setminus \{x_{i+1}, y_{i+1}, x_{i'}, y_{i'}\}$ and P_1 might contain vertices in $P^{\text{out}}_{i'+1\to i} \setminus \{x_{i'+1}, y_{i'+1}, x_i, y_i\}$, so the condition for applying Lemma 12 is not met. This issue can be overcome by selecting i and i' with $F_{i,i+1} = F_{i',i'+1}$ in such a way that the number of edges after e_{i+1} and before $e_{i'}$ in the circular ordering $(e_1, e_2, \ldots, e_s)$ is minimized. This forces P_2 to not contain any edges in $(e_{i+2}, \ldots, e_{i'-1})$ and their reversal, meaning that P_2 cannot contain any vertex in $P^{\text{out}}_{i+1\to i'} \setminus \{x_{i+1}, y_{i+1}, x_{i'}, y_{i'}\}$. Being able to apply Lemma 12 to one of $\overline{P_1}$ and $\overline{P_2}$ is enough, since we already know that $\mathsf{rotation}(C) = 4$, $\mathsf{rotation}(P^{\text{out}}_{i\to i+1}) = 2$ and $\mathsf{rotation}(P^{\text{out}}_{i'\to i'+1}) = 2$ by Lemma 11. These rotation numbers force $\mathsf{rotation}(P_2) = -\mathsf{rotation}(P_1)$.

By Lemma 12, the rotation of $P_{i'\to i'+1}$ is -2, so the last number of the string $Z_{i'\to i'+1}$ is -2. Observe that $\overline{P_1}$ is a suffix of the path $P_{i'\to i'+1}$ and $\overline{P_1} \neq P_{i'\to i'+1}$, so $\mathsf{rotation}(\overline{P_1}) = -2$ implies that $Z_{i'\leftarrow i'+1}$ contains a number 0 that is not the first number of the string. Consequently, $F_{i',i'+1}$ cannot be of type $(-)$. Also, $F_{i',i'+1}$ cannot be of type $(\sqcup, *)$, since a requirement for type $(\sqcup, *)$ due to Lemma 14 is that all numbers in the string $Z_{i'\to i'+1}$ are at most -1, except for the first number of the string. Lemma 17 forces $F_{i',i'+1}$ to be of type $(*, \sqcup)$.

By a similar argument that considers the suffix P_1 of the path $P_{i\leftarrow i+1}$, we may also force $F_{i,i+1}$ to be of type $(\sqcup, *)$. The fact that both types $(*, \sqcup)$ and $(\sqcup, *)$ exist is a contradiction to Lemma 17, so $F_{i,i+1}$ and $F_{i',i'+1}$ cannot be the same face. ◀

Strictly monotone cycles As a consequence of the above lemma, all paths $P_{i\to i+1}$ cannot use any edge that is in G_k. Although the starting and the ending edges of $P_{i\to i+1}$ might be virtual edges, the remaining edges of $P_{i\to i+1}$ must appear in the original graph G. We let H

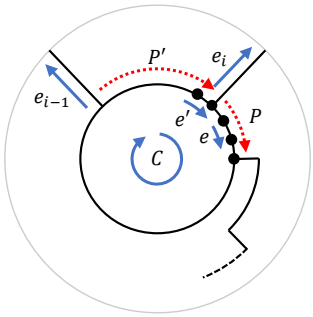


Figure 16 Showing that C is strictly monotone.

denote the subgraph of G induced by the set of the undirected version of all edges of $P_{i\to i+1}$, except for the first edge $\overline{e_i}$ and the last edges e_{i+1}, over all $1 \le i \le s$. We now show that a strictly monotone cycle of G can be found by considering the central face of H.

▶ **Lemma 20.** *The facial cycle C of the central face of H is a strictly monotone cycle of G.*

Proof. By Lemmas 18 and 19, the cycle formed by concatenating the paths $P_{i\to i+1}$, excluding the first edge $\overline{e_i}$ and the last edges e_{i+1}, over all $1 \le i \le s$, separate the central face of $\tilde{G}$ from the outer face of $\tilde{G}$ and all faces $F_{i,i+1}$, for all $1 \le i \le s$. Therefore, the central face and the outer face of H are distinct, where the central face of H contains the central face of $\tilde{G}$, and the outer face of H contains the outer face of $\tilde{G}$ and also the faces $F_{i,i+1}$, for all $1 \le i \le s$.

Let C be the facial cycle of the central face of H. We claim that C must be a simple cycle, so the above discussion implies that C is an essential cycle of $\tilde{G}$. To see this, consider any edge $e \in C$. Since the undirected version of e is in H, at least one of e and $\overline{e}$ is an edge in a path $P_{i\to i+1}$, for some $1 \le i \le s$. We cannot have $\overline{e} \in P_{i'\to i'+1}$ for any $1 \le i' \le s$, since implies that the face $F_{i',i'+1}$ is incident to e from the right, meaning that $F_{i',i'+1}$ is contained in the central face of H, which is impossible. Therefore, for each $e \in C$, we have $e \in P_{i\to i+1}$ for some $1 \le i \le s$. From this discussion, we infer that we cannot simultaneously have $e \in C$ and $\overline{e} \in C$, and this forces C to a simple cycle, as C is a facial cycle.

Edge labels Next, consider any $e \in C$, We calculate $\ell_C(e)$ with respect to the reference edge $e^\star$ in $\tilde{G}$. By Lemma 17, either all faces $F_{i,i+1}$ are of the type $(-)$ and $(\sqcup, *)$ or all faces $F_{i,i+1}$ are of the type $(-)$ and $(*, \sqcup)$. We claim that C is monotonically decreasing for the first case, and C is monotonically increasing for the second case.

Suppose all faces $F_{i,i+1}$ are of the type $(-)$ and $(\sqcup, *)$. Fix an $e \in C$, and let $e_i \in \mathcal{N}_{\mathsf{south}}(A)$ be chosen such that $e \in P_{i\to i+1}$. We calculate $\ell_C(e)$ by consider any crossing-free path $P = (e^\star, \ldots, \overline{e_i})$ of G_k from $e^\star$ to $\overline{e_i}$ and consider the subpath $P' = \overline{e_i} \circ \cdots \circ e$ of $P_{i\to i+1}$. Then $\ell_C(e) = \mathsf{rotation}(P) + \mathsf{rotation}(P')$. By Lemma 10, we have $\mathsf{rotation}(P) = 1$. By Lemma 14, we have $\mathsf{rotation}(P') \le -1$, as we cannot have $e = e_i$. Therefore, $\ell_C(e) \le 0$, so C is monotonically decreasing.

Suppose all faces $F_{i,i+1}$ are of the type $(-)$ and $(*, \sqcup)$. Similarly, by combining Lemma 14 with the fact that $F_{i,i+1}$ is a regular face, we obtain that the rotation from $\overline{e_i}$ to any intermediate edge e of $P_{i\to i+1}$ along the path $P_{i\to i+1}$ is at least -1. Therefore, by the same argument as above, we obtain that $\ell_C(e) \ge 0$ for all $e \in C$, so C is monotonically increasing.

Strict monotonicity To finish the proof, we still need to show that C is either strictly monotonically decreasing or strictly monotonically increasing. That is, we need to show that we cannot have $\ell_C(e) = 0$ for all $e \in C$. Now suppose $\ell_C(e) = 0$ for all $e \in C$, and we will show that this implies that all faces $F_{i,i+1}$ are of the type $(-)$, contradicting Lemma 17. See Figure 16 for an illustration for this part of the proof.

Again, here we assume that all faces $F_{i,i+1}$ are of the type $(-)$ and $(\sqcup, *)$. The proof for the other case where all faces $F_{i,i+1}$ are of the type $(-)$ and $(*, \sqcup)$ is similar. Consider any $e \in C$. We know that $e \in P_{i\to i+1}$ for some $1 \le i \le s$. We extend e into a subpath P of C such that P is a subpath of $P_{i\to i+1}$, and both the edge immediately preceding P and the edge immediately follows P in C are not in $P_{i\to i+1}$. Since the edge labels $\ell_C(\tilde{e})$ of all edges $\tilde{e}$ of P are zero, the rotation from $\overline{e_i}$ to any edge $\tilde{e}$ of P along the path $P_{i\to i+1}$ must be -1. By also considering the two edges of $P_{i\to i+1}$ immediately before and after P, we obtain a sequence of rotation numbers $0 \circ (-1)^c \circ (-2)$ that is a substring of $Z_{i\to i+1}$. Such a substring corresponds to a $\sqcup$-shape in the drawing.

There are two cases. If $F_{i,i+1}$ is of type $(-)$, then the above string $0 \circ (-1)^c \circ (-2)$ must be the entire string $Z_{i\to i+1}$. If $F_{i,i+1}$ is of type $(\sqcup, *)$, then in view of Lemma 14 the above string $0 \circ (-1)^c \circ (-2)$ must be a prefix of $Z_{i\to i+1}$. In either case, the edge in $P_{i\to i+1}$ immediately before P must be $\overline{e_i}$.

Now consider the edge e' in C immediately before P. This edge e' must be the second last edge in $P_{i-1\to i}$, that is, e' is immediately before the last edge e_i of $P_{i-1\to i}$. By repeating the above analysis again with e' instead of e, then we obtain a path P' that contains e'. Because here e' is second last edge in $P_{i-1\to i}$, the path P' must cover all intermediate edges of $P_{i-1\to i}$. In other words, $P_{i-1\to i}$ must have the following structure: After the first edge $\overline{e_{i-1}}$, make a 90° left turn to enter C, go straight along C, and then make another 90° left turn from e' to e_i, so $F_{i-1,i}$ is of type $(-)$. We may repeat the same analysis for $F_{i-2,i-1}$, $F_{i-3,i-2}$, and so on, to infer that all of them are of type $(-)$. ◀

Consider Figure 17 for an example of extracting a strictly monotone cycle. In the figure, H is the subgraph induced by the vertices in the shaded area. The cycle $C = (v_1, v_2, \ldots, v_5)$ is the facial cycle of the central face of H. In this example, $\mathcal{N}_{\mathsf{south}}(A) = (e_1, e_2, \ldots, e_5)$. 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 is strictly monotonically increasing. We can calculate $\ell_C((v_1, v_2)) = 1$ by first traverse from $e^\star$ to $\overline{e_2}$ via a crossing-free path P and then traverse from $\overline{e_2}$ to (v_1, v_2) along the path $P_{2\to 3}$, as (v_1, v_2) is an intermediate edge of $P_{2\to 3}$. 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. We summarize the discussion of this section as a lemma.

▶ **Lemma 21.** *Suppose the greedy algorithm outputs a good sequence $A = (S_1, S_2, \ldots, S_k)$ that does not cover the set of all horizontal segments $\mathcal{S}_{\mathsf{h}}$. Then a strictly monotone cycle of the original graph G can be computed in $O(n)$ time.*

Proof. By Lemma 20, the facial cycle C of the central face of H is a strictly monotone cycle of G. The computation of H and C can be done in $O(n)$ time. ◀

We are now ready to prove Theorem 5.

Proof of Theorem 5. We run the $O(n \log n)$-time greedy algorithm of Lemma 9 for the given input $(\mathcal{R}, e^\star)$ such that $\mathcal{N}_{\mathsf{north}}(S) = \emptyset$ for the horizontal segment $S \in \mathcal{S}_{\mathsf{h}}$ that contains $e^\star$. There are two possible outcomes of the algorithm. If we obtain a good sequence that covers

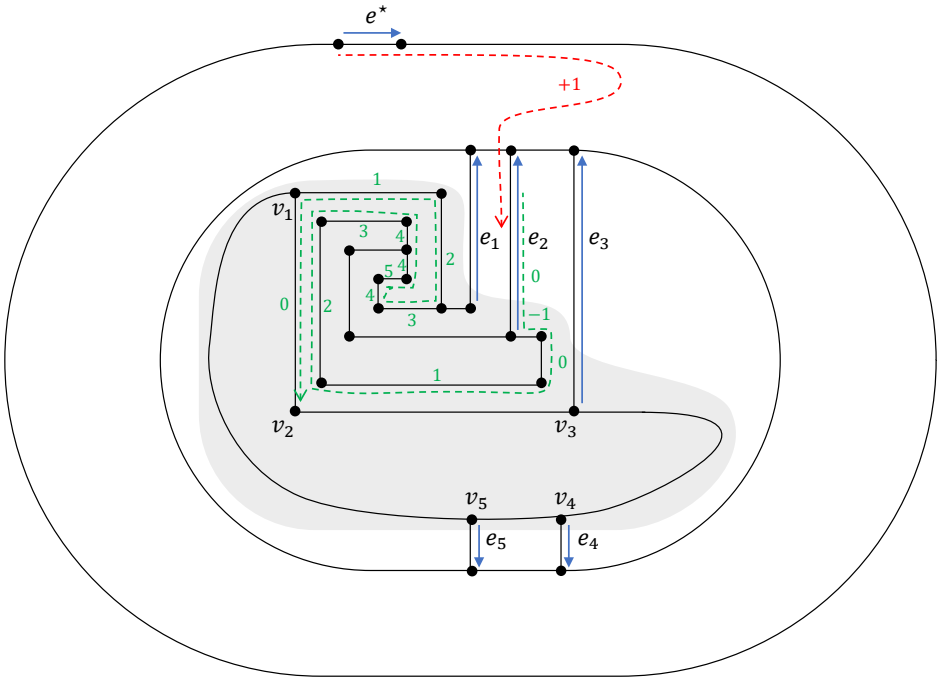


Figure 17 Extracting a strictly monotone cycle $C = (v_1, v_2, \ldots, v_5)$.

all horizontal segments $\mathcal{S}_\mathsf{h}$, then we may use Lemma 8 to compute a drawing of $(\mathcal{R}, e^\star)$. Otherwise, the algorithm stops with a good sequence that does not cover all horizontal segments $\mathcal{S}_\mathsf{h}$, and no more progress can be made, in which case a strictly monotone cycle can be computed in $O(n)$ time by Lemma 21. ◀

6 Searching for the reference edge

In this section, we show that our drawing algorithm of Theorem 5 for an ortho-radial representation $\mathcal{R}$ with a fixed reference edge $e^\star$ can be extended to a drawing algorithm for an ortho-radial representation $\mathcal{R}$ without a fixed reference edge, at the cost of an $O(\log n)$ factor overhead in the time complexity. The idea is that we may to a binary search to find a reference edge $e^\star$ such that $(\mathcal{R}, e^\star)$ is drawable, if such an edge $e^\star$ exists.

Throughout this section, we write $\ell_C^{e^\star}(e)$ to denote the edge label of e with respect to C using $e^\star$ as the reference edge. We first show the following lemma. We note that a similar lemma as been proved in [2]. We still include a proof for the sake of completeness.

▶ **Lemma 22.** *For any two choices of the edges $e_1^\star$ and $e_2^\star$ in $\overline{C_{F_o}}$, for any essential cycle C and any edge e in C, we have $\ell_C^{e_2^\star}(e) = \ell_C^{e_1^\star}(e) + \ell_{\overline{C_{F_o}}}^{e_2^\star}(e_1^\star)$.*

Proof. In the proof, we may assume that the three edges e, $e_1^\star$, and $e_2^\star$ are distinct. If $e_1^\star = e_2^\star$, then the equality is trivial as $\ell_C^{e_2^\star}(e) = \ell_C^{e_1^\star}(e)$ and $\ell_{\overline{C_{F_o}}}^{e_2^\star}(e_1^\star) = 0$. If $e = e_1^\star$, then the equality is trivial as $\ell_C^{e_1^\star}(e) = 0$ and $\ell_C^{e_2^\star}(e) = \ell_{\overline{C_{F_o}}}^{e_2^\star}(e_1^\star)$, as the reference path P used to calculate $\ell_{\overline{C_{F_o}}}^{e_2^\star}(e_1^\star)$ can also be used to calculate $\ell_C^{e_2^\star}(e)$. If $e = e_2^\star$, then $\ell_C^{e_2^\star}(e) = 0$ and $\ell_C^{e_1^\star}(e) + \ell_{\overline{C_{F_o}}}^{e_2^\star}(e_1^\star) = 0$, because $\ell_C^{e_1^\star}(e) = \ell_{\overline{C_{F_o}}}^{e_1^\star}(e_2^\star)$ implies $\ell_C^{e_1^\star}(e) = -\ell_{\overline{C_{F_o}}}^{e_2^\star}(e_1^\star)$. We also observe that e cannot be $\overline{e_1^\star}$

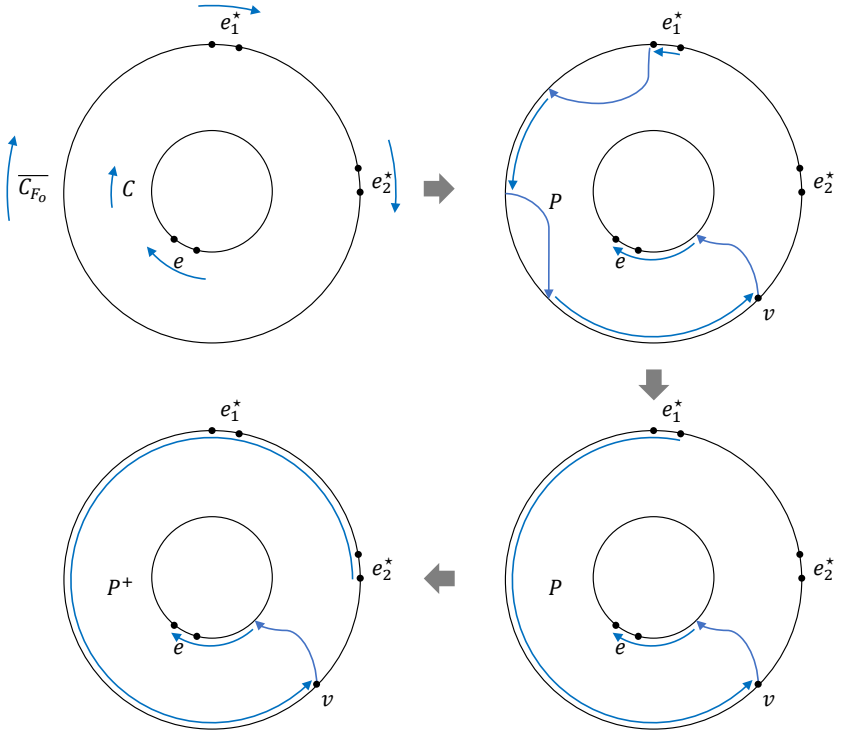


Figure 18 Finding a suitable reference path in the proof of Lemma 22.

and $\overline{e_2^\star}$ as there is no essential cycle that contains $\overline{e_1^\star}$ or $\overline{e_2^\star}$.

Finding a suitable reference path We pick a path P satisfying the following conditions.

1. P is a simple path residing in the exterior of C, starting at $e_1^\star$ or $\overline{e_1^\star}$ and ending at e or $\bar{e}$.
2. Once P leaves $\overline{C_{F_o}}$ it never visit any vertex of $\overline{C_{F_o}}$ again. More formally, if we write $P = e_1 \circ e_2 \circ \cdots \circ e_s$, then the requirement is that there exists an index i such that $e_1 \circ e_2 \circ \cdots \circ e_i$ is a subpath of $\overline{C_{F_o}}$ and the only vertex of $e_{i+1} \circ e_{i+2} \circ \cdots \circ e_s$ that belongs to $\overline{C_{F_o}}$ is its first endpoint.
3. P does contain $e_2^\star$ and $\overline{e_2^\star}$.

Such a path P can be found as follows. First, we select P as any path satisfying the first condition. To make it satisfy the remaining two conditions, we let v be the last vertex of $\overline{C_{F_o}}$ used in P, and we decompose P in such a way that $P = P_s \circ P_t$, where P_s ends at v and P_t starts at v.

For the case $P_t = \emptyset$, we know that e or $\bar{e}$ is on $\overline{C_{F_o}}$, so naturally there are two choices of subpaths of $\overline{C_{F_o}}$ connecting $e_1^\star$ or $\overline{e_1^\star}$ to e or $\bar{e}$, and one of them does not contain $e_2^\star$ and $\overline{e_2^\star}$, so this gives us a desired path P.

For the case $P_t \neq \emptyset$, we may replace P_s with a subpath from either $e_1^\star$ or $\overline{e_1^\star}$ to v to satisfy the second condition, and one of these two choices satisfies the third condition.

Let $P = \tilde{e_1^\star} \circ \cdots \circ \tilde{e}$ be a path satisfying the above conditions. Here $\tilde{e_1^\star}$ is either $e_1^\star$ or $\overline{e_1^\star}$, and $\tilde{e}$ is either e or $\bar{e}$. We let $P^+ = \tilde{e_2^\star} \circ \cdots \circ \tilde{e_1^\star} \circ \cdots \circ \tilde{e}$ be a simple path that extending P in such a way that $\tilde{e_2^\star}$ is either $e_2^\star$ or $\overline{e_2^\star}$, and the part $\tilde{e_2^\star} \circ \cdots \circ \tilde{e_1^\star}$ lies on $\overline{C_{F_o}}$. Such an extension is possible due to the requirements of P specified above. See Figure 18 for an illustration of the construction of P and P^+.

Edge label calculation We may use $P^+ = \tilde{e_2^\star} \circ \cdots \circ \tilde{e_1^\star} \circ \cdots \circ \tilde{e}$, excluding the two endpoints, as a reference path for the calculation of $\ell_C^{e_2^\star}(e)$. By the formula of direction, we may write

$$\ell_C^{e_2^\star}(e) = \mathsf{rotation}(P^+) + 2b_2 - 2b,$$

where $b_2 \in \{0,1\}$ is the indicator of whether $\tilde{e_2^\star} = \overline{e_2^\star}$, and similarly $b \in \{0,1\}$ is the indicator of whether $\tilde{e} = \overline{e}$. We break the calculation of $\mathsf{rotation}(P^+)$ into two parts:

$$\mathsf{rotation}(P^+) = \mathsf{rotation}(\tilde{e_2^\star} \circ \cdots \circ \tilde{e_1^\star}) + \mathsf{rotation}(\tilde{e_1^\star} \circ \cdots \circ \tilde{e}).$$

Similarly, $\tilde{e_2^\star} \circ \cdots \circ \tilde{e_1^\star}$, excluding the two endpoints, can be used as a reference path for the calculation of $\ell_{\overline{C_{F_\circ}}}^{e_2^\star}(e_1^\star)$, so we can infer that

$$\ell_{\overline{C_{F_\circ}}}^{e_2^\star}(e_1^\star) = \mathsf{rotation}(\tilde{e_2^\star} \circ \cdots \circ \tilde{e_1^\star}) + 2b_2 - 2b_1,$$

where $b_1 \in \{0,1\}$ is the indicator of whether $\tilde{e_1^\star} = \overline{e_1^\star}$. We may also write $\ell_C^{e_1^\star}(e)$ in terms of $\mathsf{rotation}(\tilde{e_1^\star} \circ \cdots \circ \tilde{e})$, as follows:

$$\ell_C^{e_1^\star}(e) = \mathsf{rotation}(\tilde{e_1^\star} \circ \cdots \circ \tilde{e}) + 2b_1 - 2b.$$

Combining these formulas, we obtain the desired equality

$$\ell_C^{e_2^\star}(e) = \ell_C^{e_1^\star}(e) + \ell_{\overline{C_{F_\circ}}}^{e_2^\star}(e_1^\star),$$

as all $\pm 2b$, $\pm 2b_1$, and $\pm 2b_2$ cancel out. ◀

In particular, if $\ell_{\overline{C_{F_\circ}}}^{e_2^\star}(e_1^\star) = 0$, then the above lemma implies that the edge label $\ell_C(e)$ is the same regardless of $e^\star = e_1^\star$ or $e^\star = e_2^\star$.

In the following discussion, we fix an edge e' in $\overline{C_{F_\circ}}$, and let I be the range of possible values of $\ell_C^{e'}(e)$ over all e in $\overline{C_{F_\circ}}$. Then I is a contiguous sequence of integers with $0 \in I$. For each essential cycle C, let I_C be the range of possible values of $\ell_C^{e'}(e)$ over all e in C. Then I_C is also contiguous sequence of integers.

Suppose the reference edge in use is $e^\star$. Recall that an essential cycle C is increasing if the edge labels $\ell_C^{e^\star}(e)$, for all $e \in C$, are non-negative, and there exists $e \in C$ such that $\ell_C^{e^\star}(e) \geq 1$. Then Lemma 22 implies that C is an increasing cycle if and only if the following condition is met:

$$\ell_{\overline{C_{F_\circ}}}^{e^\star}(e') + \min I_C \geq 0, \qquad \text{if } |I_C| \geq 2,$$
$$\ell_{\overline{C_{F_\circ}}}^{e^\star}(e') + \min I_C \geq 1, \qquad \text{if } |I_C| = 1.$$

Similarly, by Lemma 22, C is a decreasing cycle if and only if the following condition is met:

$$\ell_{\overline{C_{F_\circ}}}^{e^\star}(e') + \max I_C \leq 0, \qquad \text{if } |I_C| \geq 2,$$
$$\ell_{\overline{C_{F_\circ}}}^{e^\star}(e') + \max I_C \leq -1, \qquad \text{if } |I_C| = 1.$$

In view of the above, there exist two integers L_C and U_C, depending only on I_C, such that C is an increasing cycle if and only if $\ell_{\overline{C_{F_\circ}}}^{e^\star}(e') > U_C$, and C is a decreasing cycle if and only if $\ell_{\overline{C_{F_\circ}}}^{e^\star}(e') < L_C$. Therefore, C is not a strictly monotone cycle if and only if

$$\ell_{\overline{C_{F_\circ}}}^{e^\star}(e') \in [L_C, U_C].$$

In particular, (R3) is satisfied if and only if the intersection of $[L_C, U_C]$ over all essential cycles C is non-empty. In other words, (R3) is satisfied if and only if $L^* \leq U^*$, where we define $L^* = \max L_C$ and $\min U_C = U^*$, where the range of minimum and maximum is over all essential cycles C.

Binary search Given that (R1)–(R3) are satisfied for the given ortho-radial representation $\mathcal{R}$, we may use a binary search to find a reference edge $e^\star$ such that $(\mathcal{R}, e^\star)$ is drawable, as follows. We fix any edge e' in $\overline{C_{F_\circ}}$ and compute the interval I defined above. We start the binary search with the interval I. In each step of the binary search, we let x be a median of the current interval, and then we run the algorithm of Theorem 5 with a choice of the reference edge $e^\star$ such that $\ell^{e^\star}_{\overline{C_{F_\circ}}}(e') = x$.

If $x > U^*$, then there exists an increasing cycle, and the algorithm must return an increasing cycle, as there is no decreasing cycle, because $x > U^* \geq L^*$. In this case, we update the upper bound of the current interval to $x - 1$ as we learn that $x > U^*$. If $x < L^*$, then there exists a decreasing cycle, and the algorithm must return a decreasing cycle, as there is no increasing cycle, because $x < L^* \leq U^*$. In this case, we update the lower bound of the current interval to $x + 1$ as we learn that $x < L^*$. If $x \in [L^*, U^*]$, then there are no increasing cycles and no decreasing cycles, so the algorithm will return a drawing of $(\mathcal{R}, e^\star)$.

Finding a suitable reference edge There is still one remaining issue in implementing the above approach: The algorithm of Theorem 5 requires that the reference edge $e^\star$ lies on a horizontal segment $S \in \mathcal{S}_\mathsf{h}$ with $\mathcal{N}_\mathsf{north}(S) = \emptyset$. Equivalently, a choice of the reference edge $e^\star \in \overline{C_{F_\circ}}$ satisfies this requirement if and only if $e^\star \in S$ such that S meets one of the following requirements.

- $S = \overline{C_{F_\circ}}$ and $\ell_{\overline{C_{F_\circ}}}(e) = 0$ for all edges e in S.
- S is a subpath of $\overline{C_{F_\circ}}$ such that the following holds.

$$\ell_{\overline{C_{F_\circ}}}(e) = \begin{cases} -1, & \text{for the edge } e \text{ immediately preceding } S \text{ in } \overline{C_{F_\circ}}. \\ 0, & \text{for all edges } e \text{ in } S, \\ 1, & \text{for the edge } e \text{ immediately following } S \text{ in } \overline{C_{F_\circ}}. \end{cases}$$

We claim that for any $x \in \left[L_{\overline{C_{F_\circ}}}, U_{\overline{C_{F_\circ}}}\right]$ we may find an edge $e^\star$ with $\ell^{e^\star}_C(e') = x$ and satisfying the above requirement, so we may use this edge $e^\star$ as the reference edge when we run the algorithm of Theorem 5 during the binary search. Note that if $x \notin \left[L_{\overline{C_{F_\circ}}}, U_{\overline{C_{F_\circ}}}\right]$, then the cycle $\overline{C_{F_\circ}}$ is already a strictly monotone cycle under any choice of $e^\star$ with $\ell^{e^\star}_C(e') = x$, so there is no need to run the algorithm of Theorem 5.

▶ **Lemma 23.** *For each $x \in \left[L_{\overline{C_{F_\circ}}}, U_{\overline{C_{F_\circ}}}\right]$ there exists an edge $e^\star \in \overline{C_{F_\circ}}$ with $\ell^{e^\star}_{\overline{C_{F_\circ}}}(e') = x$ such that if $e^\star$ is used as the reference edge, then $e^\star$ belongs to a horizontal segment $S \in \mathcal{S}_\mathsf{h}$ with $\mathcal{N}_\mathsf{north}(S) = \emptyset$.*

Proof. For notational simplicity, in this proof we write $C = \overline{C_{F_\circ}}$. We first start with any choice of $e^\star \in C$ with $\ell^{e^\star}_C(e') = x$ and use $e^\star$ as the reference edge. Since $x \in [L_C, U_C]$, C is not a strictly monotone cycle. If $\ell_C(e) = 0$ for all $e \in C$, then $S = C$ itself is already a horizontal segment $S \in \mathcal{S}_\mathsf{h}$ with $\mathcal{N}_\mathsf{north}(S) = \emptyset$, so we are done. Otherwise, C contains edges with labels -1 and 1. We select two edges $e^- \in C$ and $e^+ \in C$ in such a way that $\ell_C(e^-) = -1$, $\ell_C(e^+) = 1$, and the length of the subpath P of C starting at e^- and ending at e^+ is minimized. Our choice of P implies that all intermediate edges $\tilde{e}$ in P have $\ell_C(\tilde{e}) = 0$, so they form a desired horizontal segment S. If $e^\star \in S$, then we are done. Otherwise, since $\ell^{\tilde{e}}_C = 0$ for any $\tilde{e} \in S$, Lemma 22 implies that all edge labels remain unchanged even if we change the reference edge from $e^\star$ to $\tilde{e}$. Therefore, S is still a horizontal segment $S \in \mathcal{S}_\mathsf{h}$ with $\mathcal{N}_\mathsf{north}(S) = \emptyset$ even if the underlying reference edge is any $\tilde{e} \in S$. Any such an edge $\tilde{e}$ satisfies the requirement of the lemma, as $\ell^{\tilde{e}}_C(e') = \ell^{e^\star}_C(e') - \ell^{e^\star}_C(\tilde{e}) = x - 0 = x$ by Lemma 22 with $e = e'$, $e^\star_1 = \tilde{e}$, and $e^\star_2 = e^\star$. ◀

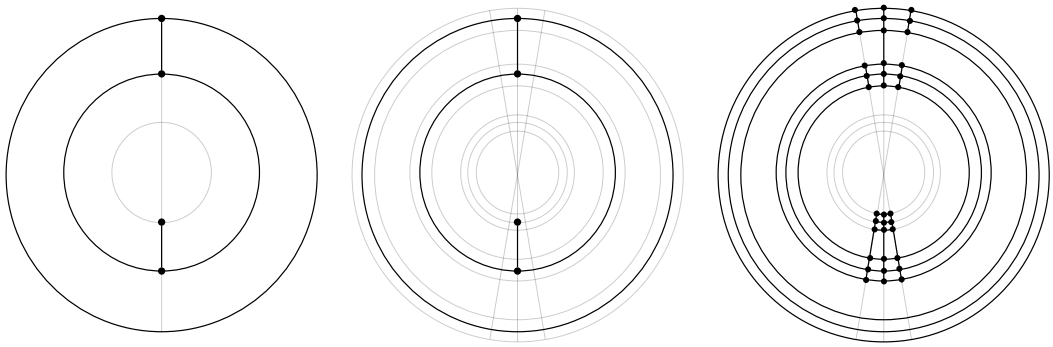


Figure 19 Reduction to biconnected simple graphs by thickening.

We are now ready to prove Theorem 6.

Proof of Theorem 6. We just need to show that for the case where the given ortho-radial representation $\mathcal{R}$ is drawable, an ortho-radial drawing realizing $\mathcal{R}$ can be computed in $O(n \log^2 n)$ time. In this case, using Lemma 23, the binary search algorithm discussed above finds a reference edge $e^\star$ such that the $O(n \log n)$-time algorithm of Theorem 5 outputs an ortho-radial drawing realizing $(\mathcal{R}, e^\star)$. This drawing is also an ortho-radial drawing realizing $\mathcal{R}$. The number of iterations of the binary search is $O(\log |I|) = O(\log n)$, so the overall time complexity is $O(n \log^2 n)$, as the cost per iteration is $O(n \log n)$, due to Theorem 5. ◀

7 A reduction to biconnected simple graphs

In this section, we show that, without loss of generality, we may assume that the input graph is simple and biconnected. Specifically, given a planar graph $G = (V, E)$, a combinatorial embedding $\mathcal{E}$ of G, and an ortho-radial representation $\mathcal{R}$ of $(G, \mathcal{E})$ satisfying (R1) and (R2), we will construct a biconnected simple planar graph $G' = (V', E')$, a combinatorial embedding $\mathcal{E}'$ of G', and an ortho-radial representation $\mathcal{R}'$ of $(G', \mathcal{E}')$ satisfying (R1) and (R2) such that $\mathcal{R}$ is drawable if and only if $\mathcal{R}'$ is drawable. See Figure 19 for an illustration of the reduction. Moreover, the reduction costs only linear time in the following sense:

- The construction of G', $\mathcal{E}'$, and $\mathcal{R}'$ takes linear time.
- Given an ortho-radial drawing of $\mathcal{R}'$, an ortho-radial drawing of $\mathcal{R}$ can be found in linear time.

The reduction Throughout this section, all edges are undirected, and we allow the graph G to have multi-edges and self-loops. The idea of the reduction is to improve the connectivity by thickening the graph. For each vertex $v \in V$, we replace it with a grid consisting of 3 horizontal lines and 3 vertical lines. Specifically, let

$$X_v = \{v_{i,j} \mid i \in \{-1, 0, 1\} \text{ and } j \in \{-1, 0, 1\}\} \quad \text{and} \quad V' = \bigcup_{v \in V} X_v.$$

For the construction of the edge set E', we first add an edge between $v_{i,j}$ and $v_{i',j'}$ if

$|i - i'| + |j - j'| = 1$, so X_v becomes a grid graph, as follows.

$$\begin{array}{ccccc} v_{-1,1} & - & v_{0,1} & - & v_{1,1} \\ | & \star & | & \star & | \\ v_{-1,0} & - & v_{0,0} & - & v_{1,0} \\ | & \star & | & \star & | \\ v_{-1,-1} & - & v_{0,-1} & - & v_{1,-1} \end{array}$$

The interior angles in the four 4-cycles in the grid graph, highlighted by $\star$, are all set to 90° in $\mathcal{R}'$ to ensure that X_v must be drawn as a grid. We write the four boundary paths of the grid as follows.

$$P^v_{1,0} = (v_{1,1}, v_{1,0}, v_{1,-1}), \qquad P^v_{0,-1} = (v_{1,-1}, v_{0,-1}, v_{-1,-1}),$$
$$P^v_{-1,0} = (v_{-1,-1}, v_{-1,0}, v_{-1,1}), \qquad P^v_{0,1} = (v_{-1,1}, v_{0,1}, v_{1,1}).$$

The concatenation of these four paths traverses the boundary of the grid in the clockwise direction.

We associate each edge incident to v to a distinct neighbor of $v_{0,0}$ in such a way that is consistent with the given counter-clockwise circular ordering $\mathcal{E}(v)$ and the angle assignment ϕ to the corners surrounding v in the given ortho-radial representation $\mathcal{R}$. By symmetry, there are five cases. In the first case, v only has one incident edge e_1, and we associate e_1 with $v_{1,0}$. In the second case, v has two incident edges e_1 and e_2 such that $\phi(e_1, e_2) = 90^\circ$ in $\mathcal{R}$, and we associate e_1 with $v_{1,0}$ and associate e_2 with $v_{0,1}$. The remaining cases are similar.

Case 1: $v - e_1 \quad : \quad e_1 \leftarrow v_{1,0}.$

Case 2:
$$\begin{array}{ccccc} & & e_2 & & \\ & & | & & \\ & & v & - & e_1 \end{array} \quad : \quad e_1 \leftarrow v_{1,0}, \quad e_2 \leftarrow v_{0,1}.$$

Case 3: $e_2 - v - e_1 \quad : \quad e_1 \leftarrow v_{1,0}, \quad e_2 \leftarrow v_{-1,0}.$

Case 4:
$$\begin{array}{ccccccc} & & & e_2 & & & \\ & & & | & & & \\ e_3 & - & v & - & e_1 & & \end{array} \quad : \quad e_1 \leftarrow v_{1,0}, \quad e_2 \leftarrow v_{0,1}, \quad e_3 \leftarrow v_{-1,0}.$$

Case 5:
$$\begin{array}{ccccccc} & & & e_2 & & & \\ & & & | & & & \\ e_3 & - & v & - & e_1 & & \\ & & & | & & & \\ & & & e_4 & & & \end{array} \quad : \quad e_1 \leftarrow v_{1,0}, \quad e_2 \leftarrow v_{0,1}, \quad e_3 \leftarrow v_{-1,0}, \quad e_4 \leftarrow v_{0,-1}.$$

For each $e = \{u, v\} \in E$, we add edges to E' to connect X_u and X_v, as follows. Suppose $e \leftarrow u_{i,j}$ and $e \leftarrow v_{k,l}$ in the above assignment. Then we connect X_u and X_v by adding the three edges $e_1 = \{x_1, y_1\}$, $e_2 = \{x_2, y_2\}$, and $e_3 = \{x_3, y_3\}$ to connect $P^u_{i,j} = (x_1, x_2, x_3)$ and $\overline{P^v_{k,l}} = (y_1, y_2, y_3)$. For example, if $e \leftarrow u_{1,0}$ and $e \leftarrow v_{0,1}$, then X_u and X_v are connected as follows.

$$\begin{array}{ccccccccccc} u_{-1,1} & - & u_{0,1} & - & u_{1,1} & - & v_{1,1} & - & v_{1,0} & - & v_{1,-1} \\ | & & | & & | & \star & | & & | & & | \\ u_{-1,0} & - & u_{0,0} & - & u_{1,0} & - & v_{0,1} & - & v_{0,0} & - & v_{0,-1} \\ | & & | & & | & \star & | & & | & & | \\ u_{-1,-1} & - & u_{0,-1} & - & u_{1,-1} & - & v_{-1,1} & - & v_{-1,0} & - & v_{1,-1} \end{array}$$

The addition of the three edges e_1, e_2, and e_3 create two 4-cycles, highlighted by $\star$. Similarly, the interior angles in these two 4-cycles are all set to 90° in $\mathcal{R}'$ to ensure that they must be drawn as rectangles.

This finishes the construction of $(G', \mathcal{E}')$, which is a biconnected simple plane graph. All remaining angles in $\mathcal{R}'$ are set in such a way that the summation of angles surrounding each vertex is 360°.

Validity of the reduction To prove that the reduction is valid, we need to show that $\mathcal{R}$ is drawable if and only if $\mathcal{R}'$ is drawable. We start with the direction $\mathcal{R} \to \mathcal{R}'$. Suppose we are given an ortho-radial drawing realizing $\mathcal{R}$. Our goal is to find an ortho-radial drawing realizing $\mathcal{R}'$. For each $v \in V$, let (r_v, θ_v) denotes its coordinate in the given ortho-radial drawing. We define

$$B_\epsilon(v) = \{(r, \theta) \mid r \in [r_v - \epsilon, r_v + \epsilon] \text{ and } \theta \in [\theta_v - \epsilon, \theta_v + \epsilon]\}.$$

For each edge $e = \{u, v\} \in E$, it is either drawn as a horizontal line (a circular arc of some circle centered at the origin) or a vertical line (a line segment of some straight line passing through the origin). That is, the drawing of e is can be described by one of the following two functions, for some choices of the parameters $(r_e, \theta_{e,1}, \theta_{e,2})$ or $(r_{e,1}, r_{e,2}, \theta_e)$:

$$\{(r, \theta) \mid r = r_e \text{ and } \theta \in [\theta_{e,1}, \theta_{e,2}]\}, \qquad \text{if } e \text{ is drawn as a horizontal line,}$$
$$\{(r, \theta) \mid r \in [r_{e,1}, r_{e,2}] \text{ and } \theta = \theta_e\}, \qquad \text{if } e \text{ is drawn as a vertical line.}$$

We define $B_\epsilon(e)$ as follows:

$$B_\epsilon(e) = \begin{cases} \{(r, \theta) \mid r \in [r_e - \epsilon, r_e + \epsilon] \text{ and } \theta \in (\theta_{e,1} + \epsilon, \theta_{e,2} - \epsilon)\} & \text{if } e \text{ is horizontal,} \\ \{(r, \theta) \mid r \in (r_{e,1} + \epsilon, r_{e,2} - \epsilon) \text{ and } \theta \in [\theta_e - \epsilon, \theta_e + \epsilon]\} & \text{if } e \text{ is vertical.} \end{cases}$$

By selecting $\epsilon > 0$ to be small enough, we can make sure that the sets $B_\epsilon(v)$ and $B_\epsilon(e)$ are non-empty and disjoint, over all $v \in V$ and $e \in E$. For each $v \in V$, we draw the grid X_v in $B_\epsilon(v)$ by drawing the following grid-lines:

$$\{(r, \theta) \mid r = r_v + s \cdot \epsilon \text{ and } \theta \in [\theta_v - \epsilon, \theta_v + \epsilon]\}, \qquad \text{for } s \in \{-1, 0, 1\},$$
$$\{(r, \theta) \mid r \in [r_v - \epsilon, r_v + \epsilon] \text{ and } \theta = \theta_v + s \cdot \epsilon\}, \qquad \text{for } s \in \{-1, 0, 1\}.$$

For each $e = \{u, v\} \in E$, we draw the three edges connecting X_u and X_v by drawing the following three lines in $B_\epsilon(e)$:

$$\{(r, \theta) \mid r = r_e + s \cdot \epsilon \text{ and } \theta \in (\theta_{e,1} + \epsilon, \theta_{e,2} - \epsilon)\}, \quad \text{for } s \in \{-1, 0, 1\}, \text{ if } e \text{ is horizontal,}$$
$$\{(r, \theta) \mid r \in (r_{e,1} + \epsilon, r_{e,2} - \epsilon) \text{ and } \theta = \theta_e + s \cdot \epsilon\}, \quad \text{for } s \in \{-1, 0, 1\}, \text{ if } e \text{ is vertical.}$$

The validity of this drawing of $\mathcal{R}'$ follows from the disjointness of the sets $B_\epsilon(v)$ and $B_\epsilon(e)$, over all $v \in V$ and $e \in E$.

Next, we consider the other direction $\mathcal{R} \to \mathcal{R}'$. Suppose we are given an ortho-radial drawing realizing $\mathcal{R}'$. Our goal is to find an ortho-radial drawing realizing $\mathcal{R}$. For each $v \in V$, we put v at the position of $v_{0,0}$ in the given drawing of $\mathcal{R}'$. To draw each edge $e = \{u, v\} \in E$, consider the assignment $e \leftarrow u_{i,j}$ and $e \leftarrow v_{k,l}$ described in the reduction. The path $P = (u_{0,0}, u_{i,j}, v_{k,l}, v_{0,0})$ must be drawn as a straight line, due to the angle assignment of $\mathcal{R}'$ described in our reduction. That is, in the given drawing of $\mathcal{R}'$, P is drawn as either a circular arc of some circle centered at the origin or a line segment of some straight line passing through the origin. Therefore, we may use the drawing of P to embed e, and this gives us a desired drawing of $\mathcal{R}$.

8 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 [32]. 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. For example, the problem of finding an ortho-radial representation that minimizes the number of bends has only been addressed by a practical algorithm [31] 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.

References

1 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 fuer Informatik. URL: `http://drops.dagstuhl.de/opus/volltexte/2017/7223`, `doi:10.4230/LIPIcs.SoCG.2017.14`.

2 Lukas Barth, Benjamin Niedermann, Ignaz Rutter, and Matthias Wolf. A topology-shape-metrics framework for ortho-radial graph drawing. *arXiv preprint arXiv:2106.05734v1*, 2021. URL: `https://arxiv.org/abs/2106.05734v1`.

3 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.

4 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.

5 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.

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

7 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.

8 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.

9 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.

10 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 fuer Informatik, 2017.

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

12 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.

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

14 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.

15 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.

16 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.

17 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.

18 Christian A. Duncan and Michael T. Goodrich. Planar orthogonal and polyline drawing algorithms. In Roberto Tamassia, editor, *Handbook of Graph Drawing and Visualization*, chapter 8. CRC Press, 2013.

19 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.

20 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.

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

22 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.

23 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.

24 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.

25 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.

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

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

28 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.

29 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.

30 Robin S. Liggett and William J. Mitchell. Optimal space planning in practice. *Computer-Aided Design*, 13(5):277–288, 1981. Special Issue Design optimization. URL: `https://`

www.sciencedirect.com/science/article/pii/0010448581903171, doi:https://doi.org/10.1016/0010-4485(81)90317-1.

31 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.

32 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 fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2019/10457, doi:10.4230/LIPIcs.SoCG.2019.53.

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

34 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.

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

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

37 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. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14030, doi:https://doi.org/10.1111/cgf.14030.

38 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.