

Efficient Algorithms for Ortho-Radial Graph Drawing

Benjamin Niedermann 

University of Bonn, Germany
niedermann@uni-bonn.de

Ignaz Rutter 

University of Passau, Germany
rutter@fim.uni-passau.de

Matthias Wolf 

Karlsruhe Institute of Technology, Germany
matthias.wolf@kit.edu

Abstract

Orthogonal drawings, i.e., embeddings of graphs into grids, are a classic topic in Graph Drawing. Often the goal is to find a drawing that minimizes the number of bends on the edges. A key ingredient for bend minimization algorithms is the existence of an *orthogonal representation* that allows to describe such drawings purely combinatorially by only listing the angles between the edges around each vertex and the directions of bends on the edges, but neglecting any kind of geometric information such as vertex coordinates or edge lengths.

Barth et al. [2] have established the existence of an analogous *ortho-radial representation* for *ortho-radial drawings*, which are embeddings into an ortho-radial grid, whose gridlines are concentric circles around the origin and straight-line spokes emanating from the origin but excluding the origin itself. While any orthogonal representation admits an orthogonal drawing, it is the circularity of the ortho-radial grid that makes the problem of characterizing valid ortho-radial representations all the more complex and interesting. Barth et al. prove such a characterization. However, the proof is existential and does not provide an efficient algorithm for testing whether a given ortho-radial representation is valid, let alone actually obtaining a drawing from an ortho-radial representation.

In this paper we give quadratic-time algorithms for both of these tasks. They are based on a suitably constrained left-first DFS in planar graphs and several new insights on ortho-radial representations. Our validity check requires quadratic time, and a naive application of it would yield a quartic algorithm for constructing a drawing from a valid ortho-radial representation. Using further structural insights we speed up the drawing algorithm to quadratic running time.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Graph Drawing, Ortho-Radial Graph Drawing, Ortho-Radial Representation, Topology-Shape-Metrics, Efficient Algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2019.53

Related Version A full version of the paper is available at <https://arxiv.org/abs/1903.05048>.

Funding *Matthias Wolf*: Matthias Wolf was funded by the Helmholtz Program Storage and Cross-linked Infrastructures, Topic 6 Superconductivity, Networks and System Integration.

1 Introduction

Grid drawings of graphs embed graphs into grids such that vertices map to grid points and edges map to internally disjoint curves on the grid lines that connect their endpoints. Orthogonal grids, whose grid lines are horizontal and vertical lines, are popular and widely used in graph drawing. Among others, orthogonal graph drawings are applied in VLSI design (e.g., [31, 6]), diagrams (e.g., [4, 19, 14, 33]), and network layouts (e.g., [27, 23]). They have



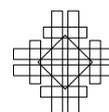
© Benjamin Niedermann, Ignaz Rutter, and Matthias Wolf;
licensed under Creative Commons License CC-BY
35th International Symposium on Computational Geometry (SoCG 2019).

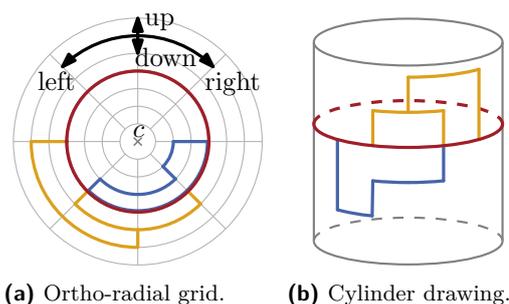
Editors: Gill Barequet and Yusu Wang; Article No. 53; pp. 53:1–53:14

Leibniz International Proceedings in Informatics

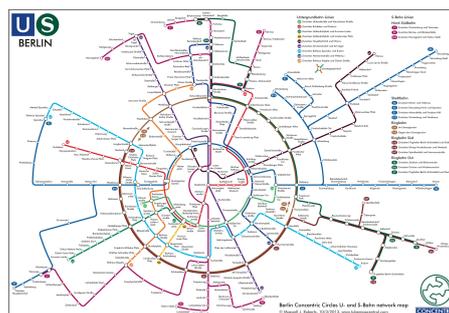


LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** An ortho-radial drawing of a graph on a grid (a) and its equivalent interpretation as an orthogonal drawing on a cylinder (b).



■ **Figure 2** Metro map of Berlin using an ortho-radial layout¹. Image copyright by Maxwell J. Roberts. Reproduced with permission.

been extensively studied with respect to their construction and properties (e.g., [30, 7, 8, 26, 1]). Moreover, they have been generalized to arbitrary planar graphs with degree higher than four (e.g., [29, 18, 9]).

Ortho-radial drawings are a generalization of orthogonal drawings to grids that are formed by concentric circles and straight-line spokes from the center but excluding the center. Equivalently, they can be viewed as graphs drawn in an orthogonal fashion on the surface of a standing cylinder, see Figure 1, or a sphere without poles. Hence, they naturally bring orthogonal graph drawings to the third dimension.

Among other applications, ortho-radial drawings are used to visualize network maps; see Figure 2. Especially, for metro systems of metropolitan areas they are highly suitable. Their inherent structure emphasizes the city center, the metro lines that run in circles as well as the metro lines that lead to suburban areas. While the automatic creation of metro maps has been extensively studied for other layout styles (e.g., [22, 25, 32, 17]), this is a new and wide research field for ortho-radial drawings.

Adapting existing techniques and objectives from orthogonal graph drawings is a promising step to open up that field. One main objective in orthogonal graph drawing is to minimize the number of bends on the edges. The core of a large fraction of the algorithmic work on this problem is the *orthogonal representation*, introduced by Tamassia [28], which describes orthogonal drawings listing (i) the angles formed by consecutive edges around each vertex and (ii) the directions of bends along the edges. Such a representation is *valid* if (I) the angles around each vertex sum to 360° , and (II) the sum of the angles around each face with k vertices is $(k - 2) \cdot 180^\circ$ for internal faces and $(k + 2) \cdot 180^\circ$ for the outer face. The necessity of the first condition is obvious and the necessity of the latter follows from the sum of inner/outer angles of any polygon with k corners. It is thus clear that any orthogonal drawing yields a valid orthogonal representation, and Tamassia [28] showed that the converse holds true as well; for a valid orthogonal representation there exists a corresponding orthogonal drawing that realizes this representation. Moreover, the proof is constructive and allows the efficient construction of such a drawing, a process that is referred to as *compaction*.

Altogether this enables a three-step approach for computing orthogonal drawings, the so-called *Topology-Shape-Metrics Framework*, which works as follows. First, fix a *topology*, i.e., combinatorial embedding of the graph in the plane (possibly planarizing it if it is non-planar);

¹ Note that ortho-radial drawings exclude the center of the grid, which is slightly different to the concentric circles maps by Maxwell J. Roberts.

second, determine the *shape* of the drawing by constructing a valid orthogonal representation with few bends; and finally, compactify the orthogonal representation by assigning suitable vertex coordinates and edge lengths (*metrics*). As mentioned before, this reduces the problem of computing an orthogonal drawing of a planar graph with a fixed embedding to the purely combinatorial problem of finding a valid orthogonal representation, preferably with few bends. The task of actually creating a corresponding drawing in polynomial time is then taken over by the framework. It is this approach that is at the heart of a large body of literature on bend minimization algorithms for orthogonal drawings (e.g., [5, 15, 13, 16, 10, 11, 12]).

Very recently Barth et al. [2] proposed a generalization of orthogonal representations to ortho-radial drawings, called *ortho-radial* representations, with the goal of establishing an ortho-radial analogue of the TSM framework for ortho-radial drawings. They show that a natural generalization of the validity conditions (I) and (II) above is not sufficient, and introduce a third, less local condition that excludes so-called *monotone cycles*, which do not admit an ortho-radial drawing. They show that these three conditions together fully characterize ortho-radial drawings. Before that, characterizations for bend-free ortho-radial drawings were only known for paths, cycles and theta graphs [21]. Further, for the special case that each internal face is a rectangle, a characterization for cubic graphs was known [20].

With the result by Barth et al. finding an ortho-radial drawing for a planar graph with fixed-embedding reduces to the purely combinatorial problem of finding a valid ortho-radial representation. In particular, since bends can be seen as additionally introduced vertices subdividing edges, finding an ortho-radial drawing with minimum number of bends reduces to finding a valid ortho-radial representation with minimum number of such additionally introduced vertices. In this sense, the work by Barth et al. constitutes a major step towards computing ortho-radial drawings with minimum number of bends.

Yet, it is here where their work still contains a major gap. While the work of Barth et al. shows that valid ortho-radial representations fully characterize ortho-radial drawings, it is unclear if it can be checked efficiently whether a given ortho-radial representation is valid. Moreover, while their existential proof of a corresponding drawing is constructive, it needs to repeatedly test whether certain ortho-radial representations are valid.

Contribution and Outline. We develop such a test running in quadratic time, thus implementing the compaction step of the TSM framework with polynomial running time. While this does not yet directly allow us to compute ortho-radial drawings with few bends, our result paves the way for a purely combinatorial treatment of bend minimization in ortho-radial drawings, thus enabling the same type of tools that have proven highly successful in minimizing bends in orthogonal drawings.

At the core of our validity testing algorithm are several new insights into the structure of ortho-radial representations. The algorithm itself is a left-first DFS that uses suitable constraints to determine candidates for monotone cycles in such a way that if a given ortho-radial representation contains a monotone cycle, then one of the candidates is monotone. While it may be obvious to use a DFS for finding cycles in general, it is far from clear how such a search works for monotone cycles in ortho-radial representations. Plugging this test as a black box into the drawing algorithm of Barth et al. yields an $O(n^4)$ -time algorithm for computing a drawing from a valid ortho-radial representation, where n is the number of vertices. Using further structural insights on the augmentation process we improve the running time of this algorithm to $O(n^2)$. Hence, our result is not only of theoretical interest, but the algorithm can be actually deployed. We believe that the algorithm is a useful intermediate step for providing initial network layouts to map designers and layout algorithms such as force directed algorithms; see also Section 5.

In Section 2 we present preliminaries that are used throughout the paper. First we formally define ortho-radial representations and recall the most important results from [2]. Afterwards we show that for the purpose of validity checking and determining the existence of a monotone cycle, we can restrict ourselves to so-called *normalized instances*. In Section 3 we give a validity test for ortho-radial representations that runs in $O(n^2)$ time. Afterwards, in Section 4, we revisit the rectangulation procedure from [2] and show that using the techniques from Section 3 it can be implemented to run in $O(n^2)$ time, improving over a naive application which would yield running time $O(n^4)$. Together with [2] this enables a purely combinatorial treatment of ortho-radial drawings. We conclude with a summary and some open questions in Section 5.

2 Preliminaries

We first formally introduce ortho-radial drawings and ortho-radial representations. Afterwards we present two transformations that we use to simplify the discussion of symmetric cases.

2.1 Ortho-Radial Drawings and Representations

We use the same definitions and conventions on ortho-radial drawings as presented by Barth et al. [2]; for the convenience of the reader we briefly repeat them here. In particular, we only consider drawings and representations without bends on the edges. As argued in [2], this is not a restriction, since it is always possible to transform a drawing/representation with bends into one without bends by subdividing edges so that a vertex is placed at each bend.

We are given a planar 4-graph $G = (V, E)$ with n vertices and fixed embedding, where a graph is a 4-graph if it has only vertices with degree at most four. We define that a path P in G is always simple, while a cycle C may contain vertices multiple times but may not cross itself. All cycles are oriented clockwise, so that their interiors are locally to the right. A cycle is part of its interior and exterior. We denote the subpath of P from u to v by $P[u, v]$ assuming that u and v are included. For any path $P = v_1, \dots, v_k$ its *reverse* is $\bar{P} = v_k, \dots, v_1$. The concatenation of two paths P_1 and P_2 is written as $P_1 + P_2$. For a cycle C in G that contains any edge at most once, the subpath $C[e, e']$ between two edges e and e' on C is the unique path on C that starts with e and ends with e' . If the start vertex u of e is contained in C only once, we also write $C[u, e']$, because then e is uniquely defined by u . Similarly, if the end vertex v of e' is contained in C only once, we also write $C[e, v]$. We also use this notation to refer to subpaths of simple paths.

In an ortho-radial drawing Δ of G each edge is directed and drawn either clockwise, counter-clockwise, towards the center or away from the center. Hence, using the metaphor of a cylinder, the edges point *right*, *left*, *down* or *up*, respectively. Moreover, *horizontal edges* point left or right, while *vertical edges* point up or down; see Figure 1.

We distinguish two types of simple cycles. If the center of the grid lies in the interior of a simple cycle, the cycle is *essential* and otherwise *non-essential*. Further, there is an unbounded face in Δ and a face that contains the center of the grid; we call the former the *outer face* and the latter the *central face*; in our drawings we mark the central face using a small “x”. All other faces are *regular*.

For two edges uv and vw incident to the same vertex v , we define the *rotation* $\text{rot}(uvw)$ as 1 if there is a right turn at v , 0 if uvw is straight and -1 if there is a left turn at v . In the special case that $u = w$, we have $\text{rot}(uvw) = -2$.

The rotation of a path $P = v_1, \dots, v_k$ is the sum of the rotations at its internal vertices, i.e., $\sum_{i=2}^{k-1} \text{rot}(v_{i-1}v_iv_{i+1})$. Similarly, for a cycle $C = v_1, \dots, v_k, v_1$, its rotation is the sum of the rotations at all its vertices (where we define $v_0 = v_k$ and $v_{k+1} = v_1$), i.e.,

$\text{rot}(C) = \sum_{i=1}^k \text{rot}(v_{i-1}v_iv_{i+1})$. We observe that $\text{rot}(P) = \text{rot}(P[s, e]) + \text{rot}(P[e, t])$ for any path P from s to t and any edge e on P . Further, we have $\text{rot}(\overline{P}) = -\text{rot}(P)$. For a face f we use $\text{rot}(f)$ to denote the rotation of the facial cycle that bounds f (oriented such that f lies on the right side of the cycle).

As introduced by Barth et al. [2], an *ortho-radial representation* Γ of a 4-planar graph G fixes the central and outer face of G as well as a reference edge e^* on the outer face such that the outer face is locally to the left of e^* . Following the convention established by Barth et al. [2] the reference edge always points right. Further, Γ specifies for each face f of G a list $H(f)$ that contains for each edge e of f the pair (e, a) , where $a \in \{90^\circ, 180^\circ, 270^\circ, 360^\circ\}$. The interpretation of (e, a) is that the edge e is directed such that the interior of f locally lies to the right of e and a specifies the angle inside f from e to the following edge. The notion of rotations can be extended to these descriptions since we can compute the angle at a vertex v enclosed by edges uv and vw by summing the corresponding angles in the faces given by the a -values. For such a description to be an ortho-radial representation, two local conditions need to be satisfied:

1. The angle sum of all edges around each vertex given by the a -fields is 360° .
2. For each face f , we have

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

These conditions ensure that angles are assigned correctly around vertices and inside faces, which implies that all properties of rotations mentioned above hold. An ortho-radial representation Γ of a graph G is *drawable* if there is a drawing Δ of G embedded as specified by Γ such that the corresponding angles in Δ and Γ are equal and the reference edge e^* points to the right. Unlike for orthogonal representations the two conditions do not guarantee that the ortho-radial representation is drawable. Therefore, Barth et al. [2] introduced a third condition, which is formulated in terms of labelings of essential cycles.

For a simple, essential cycle C in G and a path P from the target vertex s of the reference edge e^* to a vertex v on C the *labeling* ℓ_C^P assigns to each edge e on C the label $\ell_C^P(e) = \text{rot}(e^* + P + C[v, e])$. In this paper we always assume that P is *elementary*, i.e., P intersects C only at its endpoints. For these paths the labeling is independent of the actual choice of P , which was shown by Barth et al. [2]. We therefore drop the superscript P and write $\ell_C(e)$ for the labeling of an edge e on an essential cycle C . We call an essential cycle *monotone* if either all its labels are non-negative or all its labels are non-positive. A monotone cycle is a *decreasing* cycle if it has at least one strictly positive label, and it is an *increasing* cycle if C has at least one strictly negative label. An ortho-radial representation is *valid* if it contains neither decreasing nor increasing cycles. The validity of an ortho-radial representation ensures that on each essential cycle with at least one non-zero label there is at least one edge pointing up and one pointing down. The main theorem of Barth et al. [2] can be stated as follows.²

► **Proposition 1** (Reformulation of Theorem 5 in [3]). *An ortho-radial representation is drawable if and only if it is valid.*

² In the following we refer to the full version [3] of [2], when citing lemmas and theorems.

To that end, Barth et al. [3] prove the following results among others. Since we use them throughout this paper, we restate them for the convenience of the reader. Both assume ortho-radial representations that are not necessarily valid.

► **Proposition 2** (Lemma 12 in [3]). *Let C_1 and C_2 be two essential cycles and let $H = C_1 + C_2$ be the subgraph of G formed by these two cycles. For any common edge vw of C_1 and C_2 where v lies on the central face of H , the labels of vw are equal, i.e., $\ell_{C_1}(vw) = \ell_{C_2}(vw)$.*

► **Proposition 3** (Lemma 16 in [3]). *Let C and C' be two essential cycles that have at least one common vertex. If all edges on C are labeled with 0, C' is neither increasing nor decreasing.*

Proposition 2 is a useful tool for comparing the labels of two interwoven essential cycles. For example, if C_1 is decreasing, we can conclude for all edges of C_2 that also lie on C_1 and that are incident to the central face of H that they have non-negative labels. Proposition 3 is useful in the scenario where we have an essential cycle C with non-negative labels, and a decreasing cycle C' that shares a vertex with C . We can then conclude that C is also decreasing. In particular, these two propositions together imply that the central face of the graph H formed by two decreasing cycles is bounded by a decreasing cycle.

2.2 Symmetries and Normalization

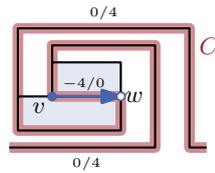
In our arguments we frequently exploit certain symmetries. For an ortho-radial representation Γ we introduce two new ortho-radial representations, its *flip* $\bar{\Gamma}$ and its *mirror* $\hat{\Gamma}$. Geometrically, viewed as a drawing on a cylinder, a flip corresponds to rotating the cylinder by 180° around a line perpendicular to the axis of the cylinder so that it is upside down, whereas mirroring corresponds to mirroring it at a plane that is parallel to the axis of the cylinder. Intuitively, the first transformation exchanges left/right and top/bottom, and thus preserves monotonicity of cycles, while the second transformation exchanges left/right but not top/bottom, and thus maps increasing cycles to decreasing ones and vice versa. This intuition is indeed true with the correct definitions of $\bar{\Gamma}$ and $\hat{\Gamma}$, but due to the non-locality of the validity condition for ortho-radial representations and the dependence on a reference edge this requires some care.

Moreover, in the following sections we restrict ourselves to instances with minimum degree 2, which we call *normalized* instances. To this end we replace each degree-1 vertex v by a 4-cycle forming a rectangle. Since v is not part of any simple essential cycle, this does not affect the validity of the representation. Normalized instances do not contain 360° turns at vertices, which simplifies our arguments. See the full version of this paper for details [24].

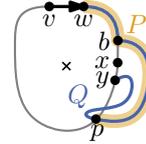
3 Finding Monotone Cycles

The two conditions for ortho-radial representations are local and checking them can easily be done in linear time. We therefore assume in this section that we are given a planar 4-graph G with an ortho-radial representation Γ . The condition for validity however references all essential cycles of which there may be exponentially many. We present an algorithm that checks whether Γ contains a monotone cycle and computes such a cycle if one exists. The main difficulty is that the labels on a decreasing cycle C depend on an elementary path P from the reference edge to C . However, we know neither the path P nor the cycle C in advance, and choosing a specific cycle C may rule out certain paths P and vice versa.

We only describe how to search for decreasing cycles; increasing cycles can be found by searching for decreasing cycles in the mirrored representation. A decreasing cycle C is *outermost* if it is not contained in the interior of any other decreasing cycle. Clearly, if Γ contains a decreasing cycle, then it also has an outermost one. We first show that in this case this cycle is uniquely determined.



■ **Figure 3** The search from vw finds the non-decreasing cycle C . Edges are labeled $\ell_C(e)/\tilde{\ell}(e)$.



■ **Figure 4** Path Q and its prefix P that leaves C once and ends at a vertex p of C .

► **Lemma 4.** *If Γ contains a decreasing cycle, there is a unique outermost decreasing cycle.*

Proof. Assume that Γ has two outermost decreasing cycles C_1 and C_2 , i.e., C_1 does not lie in the interior of C_2 and vice versa. Let C be the cycle bounding the outer face of the subgraph $H = C_1 + C_2$ that is formed by the two decreasing cycles. By construction, C_1 and C_2 lie in the interior of C , and we claim that C is a decreasing cycle contradicting that C_1 and C_2 are outermost. To that end, we show that $\ell_C(e) = \ell_{C_1}(e)$ for any edge e that belongs to both C and C_1 , and $\ell_C(e) = \ell_{C_2}(e)$ for any edge e that belongs to both C and C_2 . Hence, all edges of C have a non-negative label since C_1 and C_2 are decreasing. By Proposition 3 there is at least one label of C that is positive, and hence C is a decreasing cycle.

It remains to show that $\ell_C(e) = \ell_{C_1}(e)$ for any edge e that belongs to both C and C_1 ; the case that e belongs to both C and C_2 can be handled analogously. Let Γ_H be the ortho-radial representation Γ restricted to H . We flip the cylinder to exchange the outer face with the central face and vice versa. More precisely, the reverse edge \bar{e} of e lies on the central face of the flipped representation $\bar{\Gamma}_H$ of Γ_H . Further, it proves that $\bar{\ell}_{\bar{C}}(\bar{e}) = \ell_C(e)$ and $\bar{\ell}_{\bar{C}_1}(\bar{e}) = \ell_{C_1}(e)$, where $\bar{\ell}$ is the labeling in $\bar{\Gamma}_H$. Hence, by Proposition 2 we obtain $\bar{\ell}_{\bar{C}}(\bar{e}) = \bar{\ell}_{\bar{C}_1}(\bar{e})$. Flipping back the cylinder, we obtain $\ell_C(e) = \ell_{C_1}(e)$. ◀

The core of our algorithm is an adapted left-first DFS. Given a directed edge e it determines the outermost decreasing cycle C in Γ such that C contains e in the given direction and e has the smallest label among all edges on C , if such a cycle exists. By running this test for each directed edge of G as the start edge, we find a decreasing cycle if one exists.

Our algorithm is based on a DFS that visits each vertex at most once. A left-first search maintains for each visited vertex v a reference edge $\text{ref}(v)$, the edge of the search tree via which v was visited. Whenever it has a choice which vertex to visit next, it picks the first outgoing edge in clockwise direction after the reference edge that leads to an unvisited vertex. In addition to that, we employ a filter that ignores certain outgoing edges during the search. To that end, we define for all outgoing edges e incident to a visited vertex v a *search label* $\tilde{\ell}(e)$ by setting $\tilde{\ell}(e) = \tilde{\ell}(\text{ref}(v)) + \text{rot}(\text{ref}(v) + e)$ for each outgoing edge e of v . In our search we ignore edges with negative search labels. For a given directed edge vw in G we initialize the search by setting $\text{ref}(w) = vw$, $\tilde{\ell}(vw) = 0$ and then start searching from w .

Let T denote the directed search tree with root w constructed by the DFS in this fashion. If T contains v , then this determines a *candidate cycle* C containing the edge vw . If C is a decreasing cycle, which we can easily check by determining an elementary path from the reference edge to C , we report it. Otherwise, we show that there is no outermost decreasing cycle C such that vw lies on C and has the smallest label among all edges on C .

It is necessary to check that C is essential and decreasing. For example the cycle in Figure 3 is found by the search and though it is essential, it is nondecreasing. This is caused by the fact that the label of vw is actually -4 on this cycle but the search assumes it to be 0.



(a) The edge vw lies on the outer face of H . (b) The edge vw does not lie on the outer face of H .

■ **Figure 5** The two possible embeddings of the subgraph formed by the decreasing cycle C and the path P , which was found by the search.

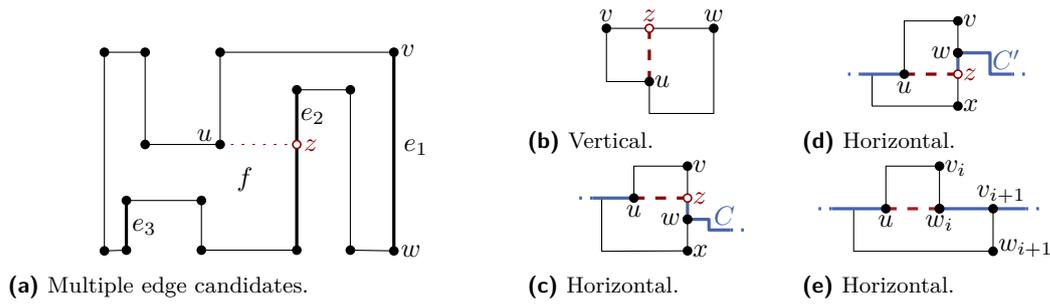
► **Lemma 5.** *Assume Γ contains a decreasing cycle. Let C be the outermost decreasing cycle of Γ and let vw be an edge on C with the minimum label, i.e., $\ell_C(vw) \leq \ell_C(e)$ for all edges e of C . Then the left-first DFS from vw finds C .*

Proof. Assume that the search does not find C . Let T be the tree formed by the edges visited by the search. Since the search does not find C by assumption, a part of $C[w, v]$ does not belong to T . Let xy be the first edge on $C[w, v]$ that is not visited, i.e., $C[w, x]$ is a part of T but $xy \notin T$. There are two possible reasons for this. Either $\tilde{\ell}(xy) < 0$ or y has already been visited before via another path Q from w with $Q \neq C[w, y]$. The case $\tilde{\ell}(xy) < 0$ can be excluded as follows. By the construction of the labels $\tilde{\ell}$, for any path P from w to a vertex z in T and any edge e' incident to z we have $\tilde{\ell}(e') = \text{rot}(vw + P + e')$. In particular, $\tilde{\ell}(xy) = \text{rot}(C[vw, xy]) = \ell_C(xy) - \ell_C(vw) \geq 0$ since the rotation can be rewritten as a label difference (see [3, Obs. 7]) and vw has the smallest label on C .

Hence, T contains a path Q from w to x that was found by the search before and Q does not completely lie on C . There is a prefix of Q (possibly of length 0) lying on C followed by a subpath not on C until the first vertex p of Q that again belongs to C ; see Figure 4. We set $P = Q[w, p]$ and denote the vertex where P leaves C by b . By construction the edge vw lies on $C[p, b]$. The subgraph $H = P + C$ that is formed by the decreasing cycle C and the path P consists of the three internally vertex-disjoint paths $P[b, p]$, $C[b, p]$ and $\overline{C}[b, p]$ between b and p . Since edges that are further left are preferred during the search, the clockwise order of these paths around b and p is fixed. In H there are three faces, bounded by C , $\overline{C}[b, p] + \overline{P}[p, b]$ and $P[b, p] + \overline{C}[p, b]$, respectively. Since C is an essential cycle and a face in H , it is the central face and one of the two other faces is the outer face. These two possibilities are shown in Figure 5. We denote the cycle bounding the outer face but in which the edges are directed such that the outer face lies locally to the left by C' . That is, the boundary of the outer face is \overline{C}' . We distinguish cases based on which of the two possible cycles constitutes \overline{C}' .

If $\overline{C}' = \overline{C}[b, p] + \overline{P}[p, b]$ forms the outer face of H , vw lies on C' as illustrated in Figure 5a and we show that C' is a decreasing cycle, which contradicts the assumption that C is the outermost decreasing cycle. Since P is simple and lies in the exterior of C , the path P is contained in C' , which means $C'[w, p] = P$. The other part of C' is formed by $C[p, w]$. Since C forms the central face of H , the labels of the edges on $C[p, w]$ are the same for C and C' by Proposition 2. In particular, $\ell_C(vw) = \ell_{C'}(vw)$ and all the labels of edges on $C[p, w]$ are non-negative because C is decreasing. The label of any edge e on both C' and P is $\ell_{C'}(e) = \ell_{C'}(vw) + \text{rot}(vw + P[w, e]) = \ell_C(vw) + \tilde{\ell}(e) \geq 0$. Thus, the labeling of C' is non-negative. Further, not all labels of C' are 0 since otherwise C would not be a decreasing cycle by Proposition 3. Hence, C' is decreasing and contains C in its interior, a contradiction.

If $\overline{C}' = \overline{C}[p, b] + P[b, p]$, the edge vw does not lie on C' ; see Figure 5b. We show that C' is a decreasing cycle containing C in its interior, again contradicting the choice of C . As above, Proposition 2 implies that the common edges of C and C' have the same labels



■ **Figure 6** Examples of augmentations. (a) The candidate edges of u are e_1 , e_2 and e_3 . (b) Insertion of vertical edge uz . (c) Γ_{vw}^u contains a decreasing cycle. (d) Γ_{vw}^u is valid. (e) Insertion of horizontal edge uw_i because there is a horizontal path from w_i to u .

on both cycles. It remains to show that all edges xy on $\bar{P}[p, b]$ have non-negative labels. To establish this we use paths to the edge that follows b on C . This edge bc has the same label on both cycles and thus provides a handle on $\ell_{C'}(xy)$. We make use of the following equations, which follow immediately from the definition of the (search) labels.

$$\begin{aligned} \ell_{C'}(bc) &= \ell_{C'}(xy) + \text{rot}(\bar{P}[xy, db]) + \text{rot}(dbc) = \ell_{C'}(xy) - \text{rot}(P[bd, yx]) - \text{rot}(cbd) \\ \ell_C(bc) &= \ell_C(vw) + \text{rot}(C[vw, ab]) + \text{rot}(abc) \\ \tilde{\ell}(yx) &= \text{rot}(C[vw, ab]) + \text{rot}(abd) + \text{rot}(P[bd, yx]) \end{aligned}$$

Since $\ell_C(bc) = \ell_{C'}(bc)$ and $\text{rot}(abd) = -\text{rot}(dba)$, we thus get

$$\begin{aligned} \ell_{C'}(xy) &= \ell_C(vw) + \text{rot}(C[vw, ab]) + \text{rot}(abc) + \text{rot}(P[bd, yx]) + \text{rot}(cbd) \\ &= \ell_C(vw) + \tilde{\ell}(yx) + \text{rot}(dba) + \text{rot}(abc) + \text{rot}(cbd). \end{aligned}$$

Since $\ell_C(vw) \geq 0$ and $\tilde{\ell}(yx) \geq 0$ (as yx was not filtered out), it follows that $\ell_{C'}(xy) \geq \text{rot}(abc) + \text{rot}(dba) + \text{rot}(cbd) = 2$ as this is the sum of clockwise rotations around a degree-3 vertex. Hence, C' is decreasing and contains C in its interior, a contradiction. Since both embeddings of H lead to a contradiction, we obtain a contradiction to our initial assumption that the search fails to find C . ◀

The left-first DFS clearly runs in $O(n)$ time. In order to guarantee that the search finds a decreasing cycle if one exists, we run it for each of the $O(n)$ directed edges of G . Since some edge must have the lowest label on the outermost decreasing cycle, Lemma 5 guarantees that we eventually find a decreasing cycle if one exists. Increasing cycles can be found by finding decreasing cycles in the mirror representation $\hat{\Gamma}$.

► **Theorem 6.** *Let G be a planar 4-graph on n vertices and let Γ be an ortho-radial representation of G . It can be determined in $O(n^2)$ time whether Γ is valid.*

4 Rectangulation

The core of the algorithm for drawing a valid ortho-radial representation Γ of a graph G by Barth et al. [2] is a *rectangulation procedure* that successively augments G with new vertices and edges to a graph G^* along with a valid ortho-radial representation Γ^* where every face of G^* is a *rectangle*. A regular face is a rectangle if it has exactly four turns, which are all right turns. The outer and central faces are rectangles if they have no turns. The ortho-radial representation Γ^* is then drawn by computing flows in two flow networks [3, Thm. 18].

To facilitate the analysis, we briefly sketch the augmentation procedure. Here it is crucial that we assume our instances to be normalized; in particular they do not have degree-1 vertices. The augmentation algorithm works by augmenting non-rectangular faces one by one, thereby successively removing concave angles at the vertices until all faces are rectangles. Consider a face f with a left turn (i.e., a concave angle) at u such that the following two turns when walking along f (in clockwise direction) are right turns; see Figure 6. We call u a *port* of f . We define a set of *candidate edges* that contains precisely those edges vw of f , for which $\text{rot}(f[u, vw]) = 2$; see Figure 6a. We treat this set as a sequence, where the edges appear in the same order as in f , beginning with the first candidate after u . The *augmentation* Γ_{vw}^u with respect to a candidate edge vw is obtained by splitting the edge vw into the edges vz and zw , where z is a new vertex, and adding the edge uz in the interior of f such that the angle formed by zu and the edge following u on f is 90° . The direction of the new edge uz in Γ_{vw}^u is the same for all candidate edges. If this direction is vertical, we call u a *vertical port* and otherwise a *horizontal port*. We note that any vertex with a concave angle in a face becomes a port during the augmentation process. In particular, the incoming edge of the vertex determines whether the port is horizontal or vertical. The condition for candidates guarantees that Γ_{vw}^u is an ortho-radial representation. It may, however, not be valid. The crucial steps in [2] are establishing the following facts.

- Fact 1)** Let u be a vertical port. Augmenting with the first candidate never produces a monotone cycle [3, Lemma 21].
- Fact 2)** Let u be a horizontal port. Augmenting with the first candidate never produces an increasing cycle [3, Lemma 22] and augmenting with the last candidate never produces a decreasing cycle [3, Lemma 24].
- Fact 3)** Let u be a horizontal port. If two consecutive candidates $e_i = v_i w_i$ and $e_{i+1} = v_{i+1} w_{i+1}$ produce a decreasing and an increasing cycle, respectively, then w_i, v_{i+1} and u lie on a path that starts at w_i or v_{i+1} , and whose edges all point right; see Figure 6e. A suitable augmentation can be found in $O(n)$ time. [3, Lemmas 25, 26].

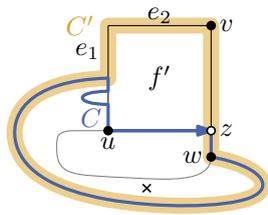
It thus suffices to test for each candidate whether Γ_{vw}^u is valid until either such a *valid augmentation* is found or we find two consecutive candidate edges where the first produces a decreasing cycle and the second produces an increasing cycle. Then, Fact 3 yields the desired valid augmentation. Since each valid augmentation reduces the number of concave angles, we obtain a rectangulation after $O(n)$ valid augmentations. Moreover, there are $O(n)$ candidates for each augmentation, each of which can be tested for validity (and increasing/decreasing cycles can be detected) in $O(n^2)$ time by Theorem 6. Thus, the augmentation algorithm can be implemented to run in $O(n^4)$ time.

In the remainder of this section we outline an improvement to $O(n^2)$ time, which is achieved in two steps. First, we show that due to the nature of augmentations the validity test can be done in $O(n)$ time. Second, for each augmentation we execute a post-processing that reduces the number of validity tests to $O(n)$ in total. The detailed constructions and omitted proofs of both steps are deferred to [24].

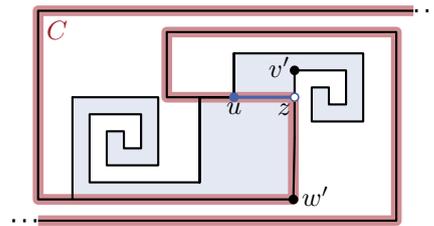
4.1 1st Improvement – Faster Validity Test

Recall from above that, once we picked a port u in a face f , there is an ordered set of candidate edges e_1, \dots, e_k . We know from [2] that one of the augmentations $\Gamma_{e_i}^u$ leads to a valid augmentation. We improve on naively testing the validity of $\Gamma_{e_i}^u$ in quadratic time.

If u is a vertical port (Figure 6b), Fact 1 guarantees that $\Gamma_{e_1}^u$ is valid, so no validity test is required. If u is a horizontal port (Figure 6c–e), we assume w.l.o.g. that the inserted edge points to the right; otherwise we consider the flipped representation $\bar{\Gamma}$.



■ **Figure 7** A decreasing cycle C that uses uz and an essential cycle C' derived from C .



■ **Figure 8** Here, the insertion of the edge uz to the last candidate $v'w'$ introduces an increasing cycle C with $l_C(uz) = -4$.

We show that in this case we can strongly restrict the direction of a monotone cycle as well as its label for the new edge uz . We start with the detection of decreasing cycles and show that for the augmentation with the first candidate edge $e_1 = vw$ a decreasing cycle C in Γ_{vw}^u may only use the new edge uz in the direction from u to z and in this case its label is 0.

► **Lemma 7.** *Let vw be the first candidate on f after u . If Γ_{vw}^u contains a decreasing cycle C , then C contains uz in this direction and $l_C(uz) = 0$.*

Proof. We first consider the case that C uses uz (and not zu) and assume that $l_C(uz) \neq 0$; see Figure 7. Since uz points right, $l_C(uz)$ is divisible by 4. Together with $l_C(uz) \geq 0$ because C is decreasing, we obtain $l_C(uz) \geq 4$. By Lemma 14 of [3] there is an essential cycle C' without uz in the subgraph H that is formed by the new rectangular face f' and C . The labels of any common edge e of C and C' are equal and $l_{C'}(e) = l_C(e) \geq 0$. All other edges of C' lie on f' . Since f' is rectangular, the labels of these edges differ by at most 1 from $l_C(uz)$. By assumption it is $l_C(uz) \geq 4$ and therefore $l_{C'}(e) \geq 3$ for all edges $e \in C' \cap f'$. Hence, C' is a decreasing cycle in G contradicting the validity of Γ . If $zu \in C$, it is $l_C(zu) \geq 2$ and a similar argument yields a decreasing cycle in Γ . ◀

While the same statement does not generally hold for all candidates, it does hold if the first candidate creates a decreasing cycle.

► **Lemma 8.** *Let vw be the first candidate and $v'w'$ be another candidate. Denote the edge inserted in $\Gamma_{v'w'}^u$ by uz' . If Γ_{vw}^u contains a decreasing cycle, any decreasing cycle C' in $\Gamma_{v'w'}^u$ uses uz' in this direction and $l_{C'}(uz') = 0$.*

Altogether, we can efficiently test which of the candidates produce decreasing cycles as follows. By Lemma 7, if the first candidate is not valid, then $\Gamma_{e_1}^u$ has a decreasing cycle that contains the new edge uz with label 0, which is hence the minimum label for all edges on the cycle. This can be tested in $O(n)$ time by Lemma 5. Fact 2 guarantees that we either find a valid augmentation or a decreasing cycle. In the former case we are done, in the second case Lemma 8 allows us to similarly restrict the labels of uz to 0 for the remaining candidate edges, thus allowing us to detect decreasing cycles in $\Gamma_{e_i}^u$ in $O(n)$ time for $i = 2, \dots, k$.

It is tempting to use the mirror symmetry to exchange increasing and decreasing cycles to deal with increasing cycles in an analogous fashion. However, this fails as mirroring invalidates the property that u is followed by two right turns in clockwise direction. For example, in Figure 8 inserting the edge to the last candidate introduces an increasing cycle C with $l_C(uz) = -4$. We therefore give a direct algorithm for detecting increasing cycles in this case.

Let $e_i = v_iw_i$ and $e_{i+1} = v_{i+1}w_{i+1}$ be two consecutive candidates for u such that $\Gamma_{e_i}^u$ contains a decreasing cycle but $\Gamma_{e_{i+1}}^u$ does not. If $\Gamma_{e_{i+1}}^u$ contains an increasing cycle, then by Fact 3 the vertices w_i, v_{i+1} and u lie on a path that starts at w_i or v_{i+1} , and whose edges

all point right. The presence of such a horizontal path P can clearly be checked in linear time, thus allowing us to also detect increasing cycles provided that the previous candidate produced a decreasing cycle. If P exists, we insert the edge uw_i or uv_{i+1} depending on whether P starts at w_i or v_{i+1} , respectively; see Figure 6e for the first case. By Proposition 3 this does not produce monotone cycles. Otherwise, if P does not exist, the augmentation $\Gamma_{e_{i+1}}^u$ is valid. In both cases we have resolved the horizontal port u successfully.

Summarizing, the overall algorithm for augmenting from a horizontal port u now works as follows. By exploiting Lemmas 7 and 8, we test the candidates in the order as they appear on f until we find the first candidate e for which Γ_e^u does not contain a decreasing cycle. Using Fact 3 we either find that Γ_e^u is valid, or we find a horizontal path as described above. In both cases this allows us to determine an edge whose insertion does not introduce a monotone cycle. Since in each test for a decreasing cycle the edge uz can be restricted to have label 0, each of the tests takes linear time. This improves the running time of the rectangulation algorithm to $O(n^3)$.

Instead of linearly searching for a suitable candidate for u we can employ a binary search on the candidates, which reduces the overall running time to $O(n^2 \log n)$.

4.2 2nd Improvement – 2-Phase Augmentation Step

We add a second phase to our augmentation step that post-processes the resulting augmentation after each step to reduce the total number of validity tests to $O(n)$.

More precisely, the first phase of the augmentation step inserts a new edge uz in a given ortho-radial representation Γ for a port u as before; we denote the resulting valid ortho-radial representation by Γ' . Afterwards, if u is a horizontal port, we apply the second phase on u . Let e_1, \dots, e_k be the candidates of u , where e_k is the candidate for the first validity test that does not fail in the first phase. We call e_1, e_{k-1} and e_k *boundary candidates* and the others *intermediate candidates*. The second phase augments Γ' such that afterwards each intermediate candidate belongs to a rectangle in the resulting ortho-radial representation Γ'' . Further, Γ'' has fewer vertices with concave angles becoming horizontal ports and at most two more vertices with concave angles becoming vertical ports during the remaining augmentation process. As the second phase is skipped for vertical ports, $O(n)$ augmentation steps are executed overall. Moreover, each edge can be an intermediate candidate for at most one vertex, which yields that there are $O(n)$ intermediate candidates over all augmentation steps. Finally, for each port there are at most three boundary candidates, which yields $O(n)$ boundary candidates over all augmentation steps. Assigning the validity tests to their candidates, the algorithm executes $O(n)$ validity tests overall. Altogether, this yields $O(n^2)$ running time in total.

► **Theorem 9.** *Given a valid ortho-radial representation Γ of a graph G , a corresponding rectangulation can be computed in $O(n^2)$ time.*

In particular, using Corollary 19 from [3], given a graph G with valid ortho-radial representation Γ , a corresponding ortho-radial drawing Δ can be computed in $O(n^2)$ time.

5 Conclusion

In this paper, we have described an algorithm that checks the validity of an ortho-radial representation in $O(n^2)$ time. In the positive case, we can also produce a corresponding drawing in the same running time, whereas in the negative case we find a monotone cycle. This answers an open question of Barth et al. [2] and allows for a purely combinatorial

treatment of the bend minimization problem for ortho-radial drawings. It is an interesting open question whether the running time can be improved to near-linear. However, our main open question is how to find valid ortho-radial representations with few bends.

References

- 1 Md. Jawaherul Alam, Stephen G. Kobourov, and Debajyoti Mondal. Orthogonal layout with optimal face complexity. *Computational Geometry*, 63:40–52, 2017.
- 2 Lukas Barth, Benjamin Niedermann, Ignaz Rutter, and Matthias Wolf. Towards a Topology-Shape-Metrics Framework for Ortho-Radial Drawings. In Boris Aronov and Matthew J. Katz, editors, *Computational Geometry (SoCG'17)*, volume 77 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.
- 3 Lukas Barth, Benjamin Niedermann, Ignaz Rutter, and Matthias Wolf. Towards a Topology-Shape-Metrics Framework for Ortho-Radial Drawings. *CoRR*, arXiv:1703.06040, 2017. arXiv:1703.06040.
- 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 P. Bertolazzi, G. Di Battista, and W. Didimo. Computing orthogonal drawings with the minimum number of bends. *IEEE Transactions on Computers*, 49(8):826–840, 2000.
- 6 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.
- 7 Therese Biedl. New lower bounds for orthogonal graph drawings. In Franz J. Brandenburg, editor, *Graph Drawing (GD'96)*, Lecture Notes of Computer Science, pages 28–39. Springer Berlin Heidelberg, 1996.
- 8 Therese Biedl and Goos Kant. A better heuristic for orthogonal graph drawings. *Computational Geometry*, 9(3):159–180, 1998.
- 9 Therese C. Biedl, Brendan P. Madden, and Ioannis G. Tollis. The three-phase method: A unified approach to orthogonal graph drawing. In Giuseppe DiBattista, editor, *Graph Drawing (GD'97)*, Lecture Notes in Computer Science, pages 391–402. Springer Berlin Heidelberg, 1997.
- 10 Thomas Bläsius, Ignaz Rutter, and Dorothea Wagner. Optimal orthogonal graph drawing with convex bend costs. *ACM Transactions on Algorithms*, 12(3):33, 2016.
- 11 Thomas Bläsius, Sebastian Lehmann, and Ignaz Rutter. Orthogonal graph drawing with inflexible edges. *Computational Geometry*, 55:26–40, 2016.
- 12 Yi-Jun Chang and Hsu-Chun Yen. On Bend-Minimized Orthogonal Drawings of Planar 3-Graphs. In Boris Aronov and Matthew J. Katz, editors, *Computational Geometry (SoCG'17)*, volume 77 of *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.
- 13 Sabine Cornelsen and Andreas Karrenbauer. Accelerated Bend Minimization. In Marc van Kreveld and Bettina Speckmann, editors, *Graph Drawing (GD'12)*, Lecture Notes of Computer Science, pages 111–122. Springer Berlin Heidelberg, 2012.
- 14 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.
- 15 Markus Eiglsperger, Michael Kaufmann, and Martin Siebenhaller. A Topology-shape-metrics Approach for the Automatic Layout of UML Class Diagrams. In *Software Visualization (SoftVis'03)*, pages 189–ff. ACM, 2003.
- 16 Stefan Felsner, Michael Kaufmann, and Pavel Valtr. Bend-optimal orthogonal graph drawing in the general position model. *Computational Geometry*, 47(3, Part B):460–468, 2014. Special Issue on the 28th European Workshop on Computational Geometry (EuroCG 2012).
- 17 Martin Fink, Herman Haverkort, Martin Nöllenburg, Maxwell Roberts, Julian Schuhmann, and Alexander Wolff. Drawing Metro Maps Using Bézier Curves. In W Didimo and M Patrignani, editors, *Graph Drawing (GD'13)*, Lecture Notes in Computer Science, pages 463–474. Springer International Publishing, 2013.

- 18 Ulrich Fößmeier and Michael Kaufmann. Drawing high degree graphs with low bend numbers. In Franz J. Brandenburg, editor, *Graph Drawing (GD'96)*, Lecture Notes in Computer Science, pages 254–266. Springer Berlin Heidelberg, 1996.
- 19 Carsten Gutwenger, Michael Jünger, Karsten Klein, Joachim Kupke, Sebastian Leipert, and Petra Mutzel. A New Approach for Visualizing UML Class Diagrams. In *Symposium on Software Visualization (SoftVis'03)*, pages 179–188, New York, NY, USA, 2003. ACM.
- 20 Madieh Hasheminezhad, S. Mehdi Hashemi, Brendan D. McKay, and Maryam Tahmasbi. Rectangular-Radial Drawings of Cubic Plane Graphs. *Computational Geometry: Theory and Applications*, 43:767–780, 2010.
- 21 Madieh Hasheminezhad, S. Mehdi Hashemi, and Maryam Tahmasbi. Ortho-radial drawings of graphs. *Australasian Journal of Combinatorics*, 44:171–182, 2009.
- 22 Seok-Hee Hong, Damian Merrick, and Hugo A. D. do Nascimento. Automatic Visualisation of Metro Maps. *Journal of Visual Languages and Computing*, 17(3):203–224, 2006.
- 23 S. Kieffer, T. Dwyer, K. Marriott, and M. Wybrow. HOLA: Human-like Orthogonal Network Layout. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):349–358, 2016.
- 24 Benjamin Niedermann, Ignaz Rutter, and Matthias Wolf. Efficient Algorithms for Ortho-Radial Graph Drawing. *CoRR*, arXiv:1903.05048, 2019. [arXiv:1903.05048](https://arxiv.org/abs/1903.05048).
- 25 Martin Nöllenburg and Alexander Wolff. Drawing and Labeling High-Quality Metro Maps by Mixed-Integer Programming. *Transactions on Visualization and Computer Graphics*, 17(5):626–641, 2011.
- 26 Achilleas Papakostas and Ioannis G. Tollis. Algorithms for area-efficient orthogonal drawings. *Computational Geometry*, 9(1):83–110, 1998.
- 27 Ulf Rüegg, Steve Kieffer, Tim Dwyer, Kim Marriott, and Michael Wybrow. Stress-Minimizing Orthogonal Layout of Data Flow Diagrams with Ports. In Christian Duncan and Antonios Symvonis, editors, *Graph Drawing (GD'14)*, Lecture Notes in Computer Science, pages 319–330. Springer Berlin Heidelberg, 2014.
- 28 R. Tamassia. On Embedding a Graph in the Grid with the Minimum Number of Bends. *Journal on Computing*, 16(3):421–444, 1987.
- 29 Roberto Tamassia, Giuseppe Di Battista, and Carlo Batini. Automatic graph drawing and readability of diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):61–79, 1988.
- 30 Roberto Tamassia, Ioannis G. Tollis, and Jeffrey Scott Vitter. Lower bounds for planar orthogonal drawings of graphs. *Information Processing Letters*, 39(1):35–40, 1991.
- 31 L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Transactions on Computers*, 30(02):135–140, 1981.
- 32 Yu-Shuen Wang and Ming-Te Chi. Focus+Context Metro Maps. *Transactions on Visualization and Computer Graphics*, 17(12):2528–2535, 2011.
- 33 Michael Wybrow, Kim Marriott, and Peter J. Stuckey. Orthogonal Connector Routing. In David Eppstein and Emden R. Gansner, editors, *Graph Drawing (GD'10)*, Lecture Notes in Computer Science, pages 219–231. Springer Berlin Heidelberg, 2010.