Extending Orthogonal Planar Graph Drawings Is Fixed-Parameter Tractable

Sujoy Bhore
Indian Institute of Technology Bombay, India

Robert Ganian
Technische Universität Wien, Austria

Liana Khazaliya
Technische Universität Wien, Austria

Fabrizio Montecchiani
University of Perugia, Italy

Martin Nöllenburg
Technische Universität Wien, Austria

Abstract

The task of finding an extension to a given partial drawing of a graph while adhering to constraints on the representation has been extensively studied in the literature, with well-known results providing efficient algorithms for fundamental representations such as planar and beyond-planar topological drawings. In this paper, we consider the extension problem for bend-minimal orthogonal drawings of planar graphs, which is among the most fundamental geometric graph drawing representations. While the problem was known to be NP-hard, it is natural to consider the case where only a small part of the graph is still to be drawn. Here, we establish the fixed-parameter tractability of the problem when parameterized by the size of the missing subgraph. Our algorithm is based on multiple novel ingredients which intertwine geometric and combinatorial arguments. These include the identification of a new graph representation of bend-equivalent regions for vertex placement in the plane, establishing a bound on the treewidth of this auxiliary graph, and a global point-grid that allows us to discretize the possible placement of bends and vertices into locally bounded subgrids for each of the above regions.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Graph algorithms

Keywords and phrases orthogonal drawings, bend minimization, extension problems, parameterized complexity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.18

Related Version *Full Version*: http://arxiv.org/abs/2302.10046

Funding *Robert Ganian*: Vienna Science and Technology Fund (WWTF) [10.47379/ICT22029]; Austrian Science Fund (FWF) [Y1329].
Liana Khazaliya: Austrian Science Fund (FWF) [Y1329]; European Union's Horizon 2020 COFUND programme [LogiCS@TUWien, grant agreement No. 101034440].
Fabrizio Montecchiani: Department of Engineering, University of Perugia, grants RICBA21LG: Algoritmi, modelli e sistemi per la rappresentazione visuale di reti and RICBA22CB: Modelli, algoritmi e sistemi per la visualizzazione e l'analisi di grafi e reti.
Martin Nöllenburg: Vienna Science and Technology Fund (WWTF) [10.47379/ICT22029].

39th International Symposium on Computational Geometry (SoCG 2023).
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 18; pp. 18:1–18:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Extending partial drawings of graphs while preserving certain desirable properties such as planarity is an algorithmic problem that received considerable attention in the last decade in graph theory, graph drawing, and computational geometry. Drawing extension problems are motivated, for instance, by visualizing networks, in which certain subgraphs represent important motifs that require a specific drawing, or by visualizing dynamic networks, in which new edges and vertices must be integrated in an existing, stable drawing. Generally speaking, we are given a graph G and a (typically connected) subgraph H of G with a drawing $\Gamma(H)$, which is called a *partial* drawing of G. The drawing $\Gamma(H)$ typically satisfies certain topological or geometric properties, e.g., planarity, upward planarity, or 1-planarity, and the goal of the corresponding extension problem is to extend $\Gamma(H)$ to a drawing $\Gamma(G)$ of the whole graph G (if possible) by inserting the missing vertices and edges into $\Gamma(H)$ while maintaining the required drawing properties.

A fundamental result in this line of research is the work of Angelini et al. [1], who showed that for planar graphs with a given partial planar drawing, the extension problem can be solved in linear time, thus matching the time complexity of unconstrained planarity testing. In fact, there is also a corresponding combinatorial characterization of planar graphs with extensible partial planar drawings via forbidden substructures [25]. In contrast to the above results, which consider topological graph embeddings, the planar drawing extension problem is NP-hard in its geometric variant, where one has to decide if a partial planar straight-line drawing $\Gamma(H)$ can be extended to a planar straight-line drawing of G [30].

In this paper, we study the geometric drawing extension problem arising in the context of one of the most fundamental graph drawing styles: orthogonal drawings [12, 16, 19, 29]. In a planar orthogonal drawing, edges are represented as polylines comprised of (one or more) horizontal and vertical segments with as few overall bends as possible, where edges are not allowed to intersect except at common endpoints. Orthogonal drawings find applications in various domains from VLSI and printed circuit board (PCB) design, to schematic network visualizations, e.g., UML diagrams in software engineering, argument maps, or flow charts.

Given the above, a key optimization goal in orthogonal drawings is bend minimization. This task is known to be NP-hard [22] when optimizing over all possible combinatorial embeddings of a given graph, but can be solved in polynomial time for a fixed combinatorial embedding using the network flow model of Tamassia [31]. Interestingly, the complexity of the bend minimization problem without a fixed embedding depends on the vertex degrees, which in the classical case of vertices being represented as points is naturally bounded by 4. If, however, the maximum vertex degree is 3, then there is a polynomial-time algorithm for bend minimization [4], and this result has recently been improved to linear time [15]; more generally, the problem is fixed-parameter tractable (FPT) in the number of degree-4 vertices [14]. In addition, it has been recently shown that the bend minimization problem is in XP (slice-wise polynomial) parameterized by the treewidth of the input graph [13].

Despite the general popularity of planar orthogonal graph drawings, the corresponding extension problem has only been considered recently [2]. While the authors of that paper showed that the existence of a planar orthogonal extension can be decided in linear time, the orthogonal bend-minimal drawing extension problem in general is easily seen to be NP-complete as it generalizes the case in which the pre-drawn part of the graph is empty [22]. Our paper addresses the parameterized complexity of the bend-minimal extension problem for planar orthogonal graph drawings under the most natural parameterization of the problem, which is the size of the subgraph that is still missing from the drawing. This parameter can be assumed to be small in many applications, e.g., when extending drawings of dynamic graphs with few added edges and vertices, and has been used broadly in the study of previous topological drawing extension problems (see, e.g., [17, 18]).

Contributions. In this paper, we establish the fixed-parameter tractability of the Bend-Minimal Orthogonal Extension (BMOE) problem when parameterized by the size κ of the missing subgraph (see the formal problem statement in Section 2). A general difficulty we had to overcome on our way to obtain our fixed-parameter algorithm is the fact that while there have been numerous recent advances in the parameterized study of drawing extension problems [18, 21, 23], the specific drawing styles considered in those papers were primarily topological in nature, while for bend minimization the geometry of the instance is crucial. In order to overcome this difficulty, we develop a new set of tools summarized below.

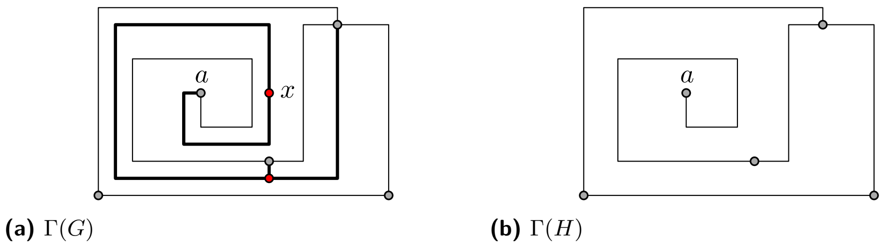
In Section 3, we make the first and simplest step towards fixed-parameter tractability of BMOE by applying an initial branching step to simplify the problem. This step allows us to reduce our target problem to Bend-Minimal Orthogonal Extension on a Face (F-BMOE), where the missing edges and vertices are drawn only in a marked face f and we have some additional information about how the edges are geometrically connected.

Next, in Section 4, we focus on solving an instance of F-BMOE. We show that certain parts of the marked face f are irrelevant and can be pruned away, and also use an involved argument to reduce the case of f being the outer face to the case of f being an inner face.

Once that is done, we enter the centerpiece of our approach in Section 5, where the aim is to obtain a suitable discretization of our instance. To this end, we split the face f into so-called *sectors*, which group together points that have the same "bend distances" to all of the connecting points on the boundary of f. Furthermore, we construct a *sector-grid* – a point-set such that each sector contains a bounded number of points from this set, and every bend-minimal extension can be modified to only use points from this set for all vertices and bends. While this latter result would make it easy to handle each individual sector by brute force, the issue is that the number of sectors can be very large, hindering tractability.

To deal with this obstacle, we capture the connections between sectors via a *sector graph* whose vertices are precisely the sectors and edges represent geometric adjacencies between sectors. Crucially, in Section 6 we show that the sector graph has treewidth bounded by a function of κ. This is non-trivial and relies on the previous application of the pruning step in Section 4. Having obtained this bound on the treewidth, the last step simply combines the already constructed sector grid with dynamic programming to solve F-BMOE (and hence also BMOE). It is perhaps worth pointing out the interesting contrast between the use of treewidth here as an implicit structural property of the sector graph – a crucial tool in our fixed-parameter algorithm – with the previously considered use of treewidth directly on the input graph – which is not known to lead to fixed-parameter tractability [13].

Related work. Several variants of drawing extension problems have been studied over the years. For instance, Chambers et al. [10] studied the problem of drawing a planar graph using straight-line edges with a prescribed convex polygon as the outer face, and proposed a method that produces drawings with polynomial area. Mchedlidze et al. [28] provide a characterization (which can be tested in linear time) to determine whether given a planar straight-line convex drawing of a biconnected subgraph G' of a planar graph G with a fixed planar embedding, this drawing can be extended to a planar straight-line drawing of G. Recently, Eiben et al. studied the problem of extending 1-planar drawings. While the problem was known to be NP-complete, they showed [18] that the problem is FPT when parameterized by the edge deletion distance. Later, in [17], they showed that the 1-planar



(a) $\Gamma(G)$

(b) $\Gamma(H)$

Figure 1 An orthogonal drawing of (a) a graph G and (b) a subgraph H of G.

extension is polynomial-time solvable when the number of vertices and edges to be added to the partial drawing is bounded. Hamm and Hliněný also studied the parameterized complexity of the extension problem in the setting of crossing minimization [23].

Other types of extension problems have also been investigated, e.g., Da Lozzo et al. [27] studied the upward planarity extension problem, and showed that this is NP-complete even for very restricted settings. Brückner and Rutter [9] showed that the partial level planarity problem is NP-complete again in severely restricted settings. For non-planar graph drawings, it is even NP-hard to determine whether a single edge can be inserted into a simple partial drawing of the remaining graph, i.e., a drawing in which any two edges intersect in at most one point [3]. Extension problems have been investigated also for other types of graph representations, in particular for intersection representations such as circular arc graphs [20] or circle graphs [8]. In the context of bend-minimal planar orthogonal drawing extension, Angelini et al. showed that the problem remains NP-hard even when a planar embedding of the whole graph is provided in the input [2].

2 Preliminaries and Basic Tools

We assume familiarity with basic concepts in parameterized complexity theory, notably fixed-parameter tractability and treewidth [11], and with standard graph drawing terminology. Recall that a planar drawing $\Gamma(G)$ is *orthogonal* if each edge is a polyline consisting of horizontal and vertical segments. A *bend* in a polygonal chain representing an edge in $\Gamma(G)$ is a point shared by two consecutive segments of the chain. For instance, Figure 1a shows an orthogonal drawing of a graph G in which edge ax has three bends.

Problem Statement. Let G be a planar graph and H be a connected subgraph of G. We call the complement $X = V(G) \setminus V(H)$ the *missing vertex set* of G, and $E_X = E(G) \setminus E(H)$ the *missing edge set*. Let $\Gamma(H)$ be a planar orthogonal drawing of H. A planar orthogonal drawing $\Gamma(G)$ *extends* $\Gamma(H)$ if its restriction to the vertices and edges of H coincides with $\Gamma(H)$. Moreover, $\Gamma(G)$ is a β-extension of $\Gamma(H)$ if it extends $\Gamma(H)$ and the total number of bends along the edges of E_X is at most β, for some $\beta \in \mathbb{N}$. For example, Figure 1a shows a 7-extension $\Gamma(G)$ of the drawing $\Gamma(H)$ in Figure 1b, with the missing vertices drawn in red.

Our problem of interest is defined as follows.

Bend-Minimal Orthogonal Extension (BMOE)
Input: $(G, H, \Gamma(H))$, integer β
Problem: Is there a β-extension $\Gamma(G)$ of $\Gamma(H)$?

We remark that BMOE is known to be NP-hard even when restricted to the case where $\beta = 0$ and $V(H) = \emptyset$ [22]. Also, unless specified otherwise, in the rest of the paper we only consider orthogonal drawings which are planar. Our parameter of interest is the number of vertices and edges missing from H, i.e., $\kappa = |V(G) \setminus V(H)| + |E(G) \setminus E(H)|$.

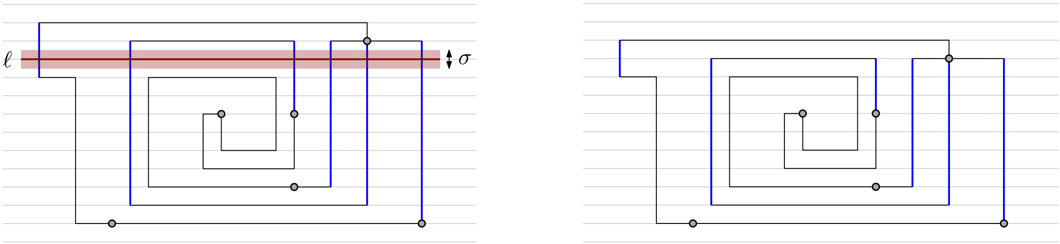


Figure 2 Two shape-equivalent orthogonal drawings such that the one on the right is obtained from the one on the left by applying a (σ, ℓ)-strip removal operation.

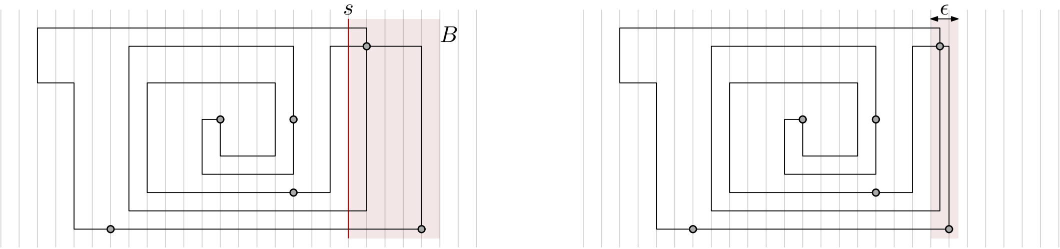


Figure 3 Two shape-equivalent orthogonal drawings such that the one on the right is obtained from the one on the left by applying Lemma 2 with the v-selection B.

Basic Tools. We introduce a set of redrawing operations that will be used as basic tools in several proofs. It is worth noting that similar operations as the ones introduced here, which are based on shortening or prolonging sets of parallel edges in orthogonal drawings, are well known (see, e.g., [6]). However, in our specific setting we have parts of the drawing that are given and cannot be modified, and handling this requires additional care in our arguments.

A *feature point* of an orthogonal drawing is a point representing either a vertex or a bend of an edge. An *edge-segment* of an orthogonal drawing is a segment that belongs to a polyline representing an edge. Two orthogonal drawings $\Gamma(G)$ and $\Gamma'(G)$ of a planar graph G are *shape-equivalent* if one can be obtained from the other by only shortening or lengthening some edge-segments. Figure 2 shows an example of two shape-equivalent drawings; in particular, the one on the right can be obtained from the one on the left by suitably shortening the blue (thicker) edge-segments. (We note that in the literature on orthogonal drawings, this is equivalent to saying that $\Gamma(G)$ and $\Gamma'(G)$ have the same *shape* but two different *metrics*.)

Let $\Gamma(G)$ be an orthogonal drawing of a graph G. Let ℓ be a horizontal (vertical) line that contains no feature points of $\Gamma(G)$ but intersects a set S of vertical (horizontal) edge-segments. Let l be the shortest distance between the endpoints of the segments in S and ℓ. For any $0 < \sigma < l$, a (σ, ℓ)*-strip removal operation* consists of decreasing the y-coordinates (x-coordinates) of all feature points above (to the right of) ℓ by σ. Analogously, for any $\sigma > 0$, a (σ, ℓ)*-strip addition operation* consists of increasing the y-coordinates (x-coordinates) of all feature points above (to the right of) ℓ by σ. See Figure 2 for an illustration of a (σ, ℓ)-strip removal operation. The following property readily follows.

▶ **Property 1.** *Let* $\Gamma(G)$ *and* $\Gamma'(G)$ *be two orthogonal drawings such that* $\Gamma'(G)$ *is obtained from* $\Gamma(G)$ *by applying a* (σ, ℓ)*-strip removal or addition operation. Then* $\Gamma(G)$ *and* $\Gamma'(G)$ *are shape-equivalent.*

Let B be a rectangle that intersects $\Gamma(G)$ such that only one side s of B is crossed by edges of G. We call B a *v-selection* if s is vertical or a *h-selection* otherwise. Also, the subdrawing of $\Gamma(G)$ inside B is called the *B-selected drawing*; see Figure 3 for an illustration of a v-selection and of the next lemma (whose proof easily follows from Property 1).

▶ **Lemma 2.** *Let $\Gamma(G)$ be an orthogonal drawing and let B be a v-selection (h-selection) of $\Gamma(G)$. For any $\epsilon > 0$, there is a drawing $\Gamma'(G)$ that is shape-equivalent to $\Gamma(G)$ and such that the B-selected drawing has width (height) at most ϵ and height (width) equal as in $\Gamma(G)$.*

3 Initial Branching

In this section, we make the first step towards the fixed-parameter tractability of BMOE by applying an initial branching step to simplify the problem – notably, this will allow us to focus on only extending the drawing inside a single face of H, and to assume that H is an induced subgraph of G.

We begin by introducing some additional notation that will be useful throughout the paper. Let $\langle(G, H, \Gamma(H)), \beta\rangle$ be an instance of BMOE. A vertex $w \in V(H)$ is called an *anchor* if it is incident to an edge in the missing edge set E_X. For a missing edge $vw \in E_X$ incident to a vertex $v \in V(H)$, we will use "ports" to specify a direction that vw could potentially use to reach v in an extension of $\Gamma(H)$; we denote these directions as d which is an element from $\{\downarrow$ (north), $\uparrow$ (south), $\leftarrow$ (east), $\rightarrow$ (west)$\}$. Formally, a *port candidate* for $vw \in E_X$ and $v \in V(H)$ is a pair (v, d). A *port-function* is an ordered set of port candidates which contains precisely one port candidate for each $vw \in E_X, v \in V(H)$, ordered lexicographically by v and then by w.

We can now formalize the target problem that we will obtain from BMOE via our exhaustive branching, which will be the focus of our considerations in Sections 4-6.

Bend-Minimal Orthogonal Extension on a Face (F-BMOE)
Input: A planar graph G_f; an induced subgraph H_f of G_f with $k = |X_f|$, where $X_f = V(G_f) \setminus V(H_f)$; a drawing $\Gamma(H_f)$ of H_f consisting of a single inner face f; a port-function $\mathcal{P}$.
Task: Compute the minimum β for which a β-extension of $\Gamma(H_f)$ exists and such that (1) missing edges and vertices are only drawn in the face f and (2) each edge $vx \in E_X$ where $v \in V(H)$ connects to x via its port candidate defined by $\mathcal{P}$, or determine that no such extension exists.

For the Turing reduction formalized in the next lemma, it will be useful to recall the definition of BMOE and κ from Section 2.

▶ **Lemma 3.** *There is an algorithm that solves an instance $\mathcal{I}$ of BMOE in time $3^{\mathcal{O}(\kappa)} \cdot T(|I|, k)$, where $T(a, b)$ is the time required to solve an instance of F-BMOE with instance size a and parameter value b.*

Proof Sketch. We exhaustively branch over all possible faces in which a missing vertex can be drawn, as well as over possible ports that will be used by each edge incident to an anchor in H. Also, additional care is needed with each missing edge with both endpoints in H in order to make H an induced subgraph of G. For this, we branch to determine whether the edge will be drawn as a straight-line segment (in which case we simply add it to H), or whether it will have at least one bend (in which case we subdivide it, mark the newly created vertex as missing, and remember that the total number of bends will decrease by 1). ◀

We note that the marked face f can be either the single inner face of $\Gamma(H_f)$ or the outer face. On a different note, while BMOE was stated as a decision problem for complexity-theoretic purposes, the output for F-BMOE is either an integer or "No". Two instances of F-BMOE are said to be *equivalent* if their outputs are the same. Note that checking whether an instance of F-BMOE admits some β-extension can be done in polynomial time by using the algorithm in [2]. The pre-drawn graph given as input to the algorithm in [2] will be $\Gamma(H_f)$ with a slight modification: if a vertex v makes an angle larger than $\frac{\pi}{2}$ in the non-marked face g of $\Gamma(H_f)$, then we add dummy vertices and connect them to v until all angles around v in g are $\frac{\pi}{2}$. This guarantees that a solution only draws missing vertices inside the marked face f (and not in g). Hence, we will assume to be dealing with instances where such an extension exists, and the task is to identify the minimum value of β. We will call a β-extension minimizing the value of β a *solution*.

4 Preprocessing

We can now focus on solving an instance of F-BMOE with only a single marked face f being of interest. The aim of this section is to make the first two steps that will allow us to solve F-BMOE. This includes pruning out certain parts of the face which are provably irrelevant, and reducing the case of f being the outer face to the case of f being an inner face.

4.1 Pruning

Let $\Gamma(G)$ be an orthogonal drawing of a graph G and let f be a face of $\Gamma(G)$. A *reflex corner* p of f is a feature point that makes an angle larger than π inside f. Also, if p is an anchor, then it is called an *essential* reflex corner. A projection ℓ of a reflex corner p is a horizontal or vertical line-segment in the interior of f that starts at p and ends at its first intersection with the boundary of f. Figure 4 (left) shows two projections ℓ_1 and ℓ_2 of a reflex corner p.

Observe that each projection ℓ of a reflex corner p divides the face f into two connected regions, which are themselves orthogonal polygons. If p is not essential and one of the two regions contains no reflex corners of its own (notice that inside this region, p needs no longer be a reflex corner) and no anchors, we call the region *redundant*. Our aim will be to show that such regions can be safely removed from the instance. More formally, recall that ℓ intersects the boundary of f in p on one side and in an element e that is either a vertex u or a point q on an edge of H on the other side of f. The *pruning operation* at ℓ for a redundant region ι works as follows. (1) If both p and e are vertices (which are therefore vertically or horizontally aligned) we add the edge pu into H, whose representation in $\Gamma(H)$ is ℓ. (2) If p is a vertex and e is an edge, we modify H by replacing q with a dummy vertex v_q that subdivides e and by adding the edge pv_q (whose representation in $\Gamma(H)$ is ℓ). (3) If p is part of an edge e' and e is also an edge, we modify H by replacing p and q with two dummy vertices v_p and v_q that subdivide e' and e and by adding the edge v_pv_q (whose representation in $\Gamma(H)$ is ℓ). We finally remove the boundary of ι from H and $\Gamma(H)$, except for the edge-segment ℓ and its end-vertices. The proof of the next lemma easily follows by suitably using v-/h-selections, see also Figure 5 for an illustration.

▶ **Lemma 4.** *Let $\mathcal{I} = \langle G_f, H_f, \Gamma(H_f), f, \mathcal{P}\rangle$ be an instance of F-TBOE. Let ℓ be a projection of some non-essential reflex corner in f, which gives rise to a redundant region ι. Then pruning ι at ℓ results in an instance $\mathcal{I}^\iota$ that is equivalent to $\mathcal{I}$.*

We can show that exhaustively applying Lemma 4 results in an instance with the following property: each projection of each non-essential reflex corner in f splits f into two faces, each of which has at least one port on its boundary. We call such instances *clean*; see Figure 4.

▶ **Lemma 5.** *There is a polynomial-time algorithm that takes as input an arbitrary instance of F-TBOE and outputs an equivalent instance which is clean.*

Given Lemma 5, we will hereinafter assume that our instances of F-TBOE are clean.

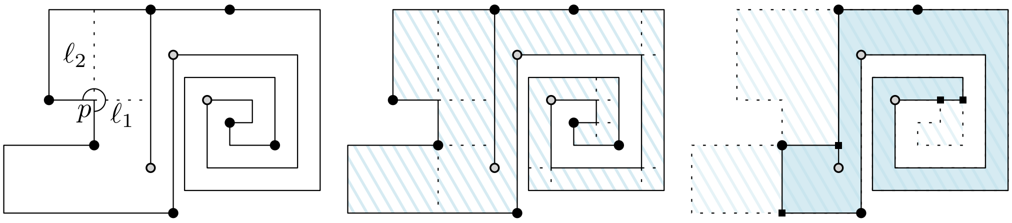


Figure 4 Left: A reflex corner p and its projections ℓ_1 and ℓ_2. Middle: A face (striped) with all its non-essential reflex corners and projections (anchor vertices have a gray filling while non-anchors are solid). Right: The corresponding clean instance (dummy vertices are drawn as small squares).

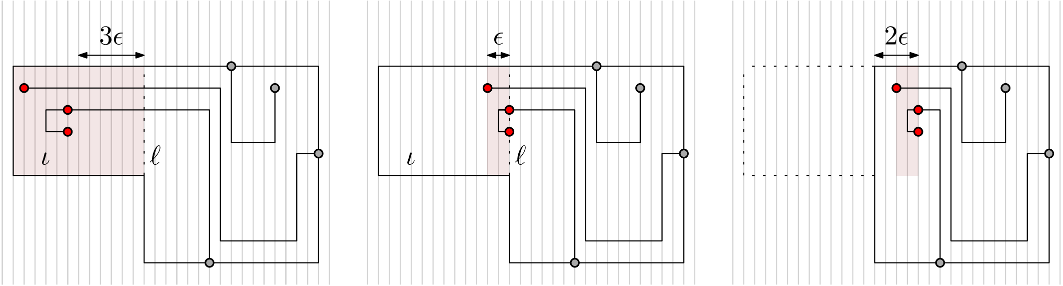


Figure 5 Illustration for the proof of Lemma 4.

4.2 Outer Face

Given an instance of F-BMOE where the marked face f is the outer face of $\Gamma(H_f)$, let us begin by constructing a rectangle that bounds $\Gamma(H_f)$ and will serve as a "frame" for any solution.

▶ **Observation 6.** *Let $\mathcal{I}$ be an instance of F-BMOE and let R be a rectangle that contains $\Gamma(H_f)$ in its interior. Then $\mathcal{I}$ admits a solution that lies in the interior of R.*

Based on Observation 6, we shall assume that any instance $\mathcal{I}$ is modified such that the outer face of $\Gamma(H_f)$ is a rectangle R containing no anchors (e.g., with four dummy vertices at its corners connected in a cycle). Notice that, while this ensures that f is no longer the outer face, f now contains a hole (that is, H_f is not connected anymore). The aim for the rest of this section is to remove this hole by connecting it to the boundary of R.

To do so, let us consider an arbitrary horizontal or vertical line-segment ζ that connects the boundary of R with an edge-segment in the drawing $\Gamma(H_f)$ and intersects no other edge-segment of $\Gamma(H_f)$. Observe that, w.l.o.g., we can assume that each edge-segment in a solution $\Gamma(G_f)$ only intersects ζ in single points (and not in a line-segment); otherwise, one may shift ζ by a sufficiently small ϵ to avoid such intersections. Roughly speaking, our aim will be to show that the instance $\mathcal{I}$ can be "cut open" along ζ to construct an equivalent instance where the boundary of the polygon includes R, and to branch in order to determine how the edges in a hypothetical solution cross through ζ. However, to do so we need to ensure that there is a solution, in which the number of such crossings through ζ is bounded.

Let us consider the drawing of a missing edge $e \in E_X$ in $\Gamma(G_f)$. The intersection points of e with ζ partition the drawing of e into polylines $e_1^\zeta, e_2^\zeta, \ldots, e_q^\zeta$, where each pair of consecutive polylines e_i^ζ and e_{i+1}^ζ touch ζ at a point, which we denote by z_i ($i = 1, \ldots, q-1$). We distinguish two cases depending on the structure of these polylines. A polyline e_j^ζ, $1 < j < q$, is called a ζ-*handle* if the unique region of the plane enclosed by e_j^ζ and ζ does not contain $\Gamma(H_f)$; otherwise the polyline is called a ζ-*spiral*. See Figure 6 for an illustration.

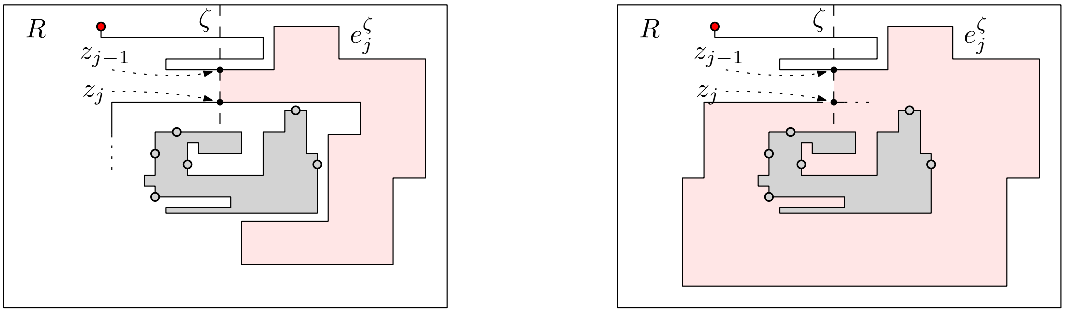


Figure 6 Illustration of ζ-handles (left) and ζ-spirals (right).

▶ **Lemma 7.** *Assume $\mathcal{I}$ and ζ are fixed as above. Then $\mathcal{I}$ admits a solution such that no missing edge contains a ζ-handle.*

Proof Sketch. By planarity, the polyline e^* representing any ζ-handle is not crossed by any edge (except possibly at common endpoints). Consider the subdrawing Γ_ζ of $\Gamma(G_f)$ formed by all vertices and edge-segments in the interior of the unique region of the plane enclosed by e^* and ζ. At high-level, we scale-down Γ_ζ and then define suitably h-/v-selections such that the transformed version of Γ_ζ can be moved to the other side of ζ without introducing crossings. At this point we can redraw e^* such that it does not cross ζ anymore and its number of bends is not increased. ◀

Next we deal with ζ-spirals: while they cannot be completely avoided, we show that one can bound the number of ζ-spirals for each edge by a function of the parameter k.

▶ **Lemma 8.** *Assume $\mathcal{I}$ and ζ are fixed as above. Then $\mathcal{I}$ admits a solution with no ζ-handles and at most $4k(k+1)$ ζ-spirals.*

Proof Sketch. The first part of the statement follows by Lemma 7. The second part can be proved by observing that pairs of consecutive ζ-spirals of the same edge can be shortcut and merged together into a single ζ-spiral if they do not enclose any vertex. On the other hand a vertex blocking this operation must be a missing vertex, and hence we have at most k consecutive blocked pairs for each of the at most $4k$ missing edges. ◀

With Lemma 8, we obtain that there exists a solution where the total number of edge-segments crossing through ζ is at most $4k(k+1)$. We can use this to branch on which edges cross through ζ and use this to make a "bridge" connecting R to the hole in f, thus resulting in an equivalent instance where f is modified to become an inner face with no holes.

▶ **Lemma 9.** *There is an algorithm that takes as input an instance $\mathcal{I}$ of F-BMOE where f is the outer face and solves it in time $2^{\mathcal{O}(k^2 \log k)} \cdot Q(|\mathcal{I}|, k)$, where $Q(a, b)$ is the time to solve an instance of F-BMOE with instance size a and parameter value b such that f is the inner face.*

Proof Sketch. We can assume that f is an inner face bounded by a rectangle R and containing a segment ζ defined as above. By Lemma 8, it is not restrictive to consider solutions such that each missing edge drawn in f contains no ζ-handles and at most $4k(k+1)$ ζ-spirals. That is, we shall consider solutions in which ζ is crossed at most $4k(k+1)$ times. The first task here is to branch over which missing edges will cross ζ (possibly multiple times) and in which order. The second task is to show that the precise position of these crossings along ζ is not important, because a hypothetical solution can always be redrawn so to use the given crossing points without increasing the number of bends. Once this is done, each such crossing point can be replaced with a dummy vertex that subdivides ζ and belongs to the new boundary of f, which now has no holes anymore. To this aim, we can use suitably defined h-/v-selections and (σ, ℓ)-strip additions. ◀

5 Discretizing the Instances

Our next aim is to define the sector graph and show that it suffices to consider only a bounded number of possible points in each sector for extending $\Gamma(H_f)$. Essentially, this allows us to combinatorially extract those properties of $\Gamma(H_f)$ that are relevant for solving F-BMOE.

5.1 Sectors and the Sector Graph

For a point $p \in f$, the *bend distance* $\mathrm{bd}(p,(a,d))$ to a port candidate (a,d) is the minimum integer q such that there exists an orthogonal polyline with q bends connecting p and a in the interior of f which arrives to a from direction d.

▶ **Definition 10.** *Let* $\mathcal{P} = ((a_1,d_1),\ldots,(a_q,d_q))$ *be an ordered set of port candidates. For each point* $p \in f$*, we define a bend-vector as the tuple* $\mathrm{vect}(p) = (\mathrm{bd}(p,(a_1,d_1)),\ldots,\mathrm{bd}(p,(a_q,d_q)))$.

▶ **Definition 11.** *Given an ordered set of port candidates* $\mathcal{P}$*, a* sector F *is a maximal connected set of points with the same bend-vector w.r.t.* $\mathcal{P}$.

When $\mathcal{P}$ is not specified explicitly, we will assume it to be the set of port candidates provided by the considered instance of F-BMOE. The face f is now partitioned into a set $\mathcal{F}$ of sectors. It is worth noting that sectors are connected regions in the face f, they do not overlap, and they cover the whole interior of f. We further notice that a sector can be degenerate, it may be a single point or a line-segment, and that pairs of (non-adjacent) sectors may have the same bend-vectors. At this point, we can define a graph representation capturing the adjacencies between the sectors in our instance; see Figure 7 for an illustration.

▶ **Definition 12.** *Sectors* A *and* B *are* adjacent *if there exists a point* p *in* A *and a direction* $d \in \{\uparrow,\downarrow,\leftarrow,\rightarrow\}$ *such that the first point outside of* A *hit by the ray starting from* p *in direction* d *is in* B.

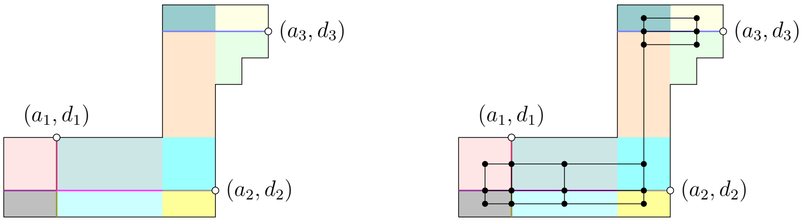


Figure 7 Left: partitioning a face f into a set $\mathcal{F}$ of sectors, with three anchors marked using white circles. Right: the graph representation of $\mathcal{F}$.

Observe that the relationship of being adjacent is symmetric; furthermore, for a specific direction d we say that sector A is *d-adjacent* to B if A is adjacent to B for this choice of d. The *sector graph* $\mathcal{G}$ is the graph whose vertex set is the set of sectors $\mathcal{F}$, and adjacencies of vertices are defined via the adjacency of sectors.

It will be useful to establish some basic properties of the sector graph. For instance, it is not difficult to observe that the sector graph is a connected planar graph. Furthermore, we can show that the boundary between two sectors is, in a sense, simple. Concerning its size, we observe that each sector contains at least one intersection point between two projections and that any such intersection point can be shared by at most nine sectors (four non-degenerate sectors plus five degenerate sectors). Hence:

▶ **Observation 13.** *The number of vertices in $\mathcal{G}$ is upper-bounded by $9x^2$, where x is the number of feature points in $\Gamma(H_F)$.*

5.2 The Sector-Grid

A property of sectors that will become important later is that, inside each sector, we only need a bounded number of positions for the placement of feature points in a hypothetical solution. In particular, our aim will be to construct a "universal" point-set with the property that there exists a solution which places feature points only on these points, and where the intersection of the point-set with each sector is upper-bounded by a function of the parameter. Before we construct such a universal point-set, we will first need to subdivide sectors into "subsectors" which have grid-like connections to each other. Crucially, we will show that the number of subsectors in each sector is upper-bounded by a function of k.

Let us fix a sector S and a direction $d \in \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$, say w.l.o.g. $d = \rightarrow$. Let a reflex corner be *critical* if it is incident to at least two distinct sectors, and (S, d)-*critical* if it is critical and also can be reached by a ray from some point in S traveling in direction d. To construct the subsectors of S, let us project all (S, d)-critical reflex corners (for all four choices of d) into S to obtain a grid, and make each induced grid cell in S a *subsector* of S. Observe that for each subsector in each sector S, it holds that its entire boundary in each direction is either the boundary of f, or touches the boundary of a single other "adjacent" subsector (which may or may not belong to S).

Crucially, we show that the number of such subsectors obtained from each sector is not too large. This will be important when using sectors for dynamic programming in Section 6, since it will allow us to bound the size of the universal point-set in each sector.

▶ **Lemma 14.** *For each S, d, there are at most $4k$ (S, d)-*critical *reflex corners.*

By applying Lemma 14 on all sides of each sector S, we obtain that S is partitioned into at most $(8k)^2$ subsectors. Observe that we may refine the sector graph constructed earlier by partitioning sectors into subsectors, with adjacencies between subsectors defined in the same way as between sectors. Note that by definition, each pair of adjacent subsectors share the complete side of the boundary that connects them. Hence, we can define a *subsector-column* as a set of subsectors which form a path in the subsector graph and span the same vertical strip in $\Gamma(H_f)$, and similarly a *subsector-row* is a set of subsectors which forms a path in the subsector graph and span the same horizontal strip in $\Gamma(H_f)$.

With the above in mind, we proceed to build the universal point-set. As our first step, we construct an auxiliary set of points we call a *skeleton*. Let us now choose an arbitrary horizontal line-segment for each subsector-row that intersects it, and similarly an arbitrary vertical line-segment for each subsector-column that intersects it. To construct the skeleton, for each subsector v, we define the point p_v to be the point at the intersection of the two line-segments intersecting the subsector.

Let $\texttt{subgridsize}(k) = 112k^3 + 202k^2 + 85k$. We place a set of $\texttt{subgridsize}(k) \times \texttt{subgridsize}(k)$ points in a grid-like arrangement into each subsector v, where the points are centered at p_v and the grid underlying these points occupies a square area of $\epsilon \times \epsilon$ for a sufficiently small ϵ. In particular, we choose ϵ to be sufficiently small so that a horizontal or vertical projection of any pair of grid points intersects with the same line-segment of $\Gamma(H_f)$. We call this point-set $\mathcal{S}_v$ the *subsector-grid* of a subsector v; in the degenerate cases where v is a line-segment or single point, the subsector-grid is a set of points on that segment or just a single point, respectively.

▶ **Lemma 15.** *There exists a solution such that each feature point not in* $\Gamma(H_f)$ *lies on a subsector-grid point of some subsector.*

Proof Sketch. The proof undergoes several steps. We first argue that if a polyline representing (part of) an edge drawn inside a sector-column (or analogously inside a sector-row) $\mathcal{A}$ contains a large number of bends, then we can redraw it and obtain an equivalent solution. This requires similar arguments as in Lemmas 7 and 8 (although in a different setting), together with new arguments dealing with edges that have a "staircase" shape. The second step is then to prove that similar redrawing arguments can be adopted to show that there are not too many disjoint polylines that represent the same edge inside $\mathcal{A}$. The last step is to show that the feature points of a hypotethical solution that lies in a subsector can always be mapped to the specific point-set defined by the subsector-grid. ◀

From Lemmas 14, 15 and by setting $\texttt{gridsize}(k) = \texttt{subgridsize}(k)^2 \cdot (8k)^2$, we obtain:

▶ **Corollary 16.** *Given an instance* $\mathcal{I}$ *of F-BMOE we can construct a point-set (called a* sector grid*) in time* $\mathcal{O}(|\mathcal{I}|)$ *with the following properties:* (1) $\mathcal{I}$ *admits a solution whose feature points all lie on the sector grid, and* (2) *each sector contains at most* $\texttt{gridsize}(k)$ *points of the sector grid.*

6 Exploiting the Treewidth of Sector Graphs

In this section, we complete the proof of our fixed-parameter tractability result by first showing that the sector graphs in fact have treewidth bounded by a function of the parameter k, and then by using this fact to design a dynamic programming algorithm solving F-BMOE.

6.1 Sector Graphs Are Tree-Like

We begin by introducing some notation that will be useful in this subsection. Let $\mathcal{P} = ((a_1, d_1), \ldots, (a_q, d_q))$ be the ordered set of port candidates for the considered face f. Also, $q \le 4k$, because the degree of the vertices being added is at most 4. For each $1 \le i \le q$, let $\mathcal{P}_i = ((a_1, d_1), \ldots, (a_i, d_i))$ be a prefix of length i of $\mathcal{P}$. For each $1 \le i \le q$, we denote by $\mathcal{F}_i$ and $\mathcal{G}_i$ the set of sectors and the sector graph, respectively, obtained by considering the bend distances to $\mathcal{P}_i$. Using this terminology, we obtain that the graph $\mathcal{G}_q$ is precisely the sector graph of our initial instance, which we will also simply denote as $\mathcal{G}$. Furthermore, for a sector $F \in V(\mathcal{G}_t)$ we denote by $\mathcal{U}_F^{t+1}$ the set of sectors in $\mathcal{G}_{t+1}$ that F is partitioned into when one additionally considers bend distances to (a_{t+1}, d_{t+1}).

▶ **Lemma 17.** *The sector graph* $\mathcal{G}_1$ *is a tree.*

Lemma 17 will be used as a base of an inductive argument establishing a bound on the treewidth of $\mathcal{G}$. See Figure 8 for an example of the sectors for two port candidates. We start by considering how each sector $F \in \mathcal{F}_t$ maps to a subset $\mathcal{U}_F^{t+1}$ of sectors in $\mathcal{F}_{t+1}$. Towards this aim, let us now consider an arbitrary sector $F \in \mathcal{F}_t$ for some $1 \le t \le q$. We say that a line-segment δ on the boundary of F is an F-*baseline* if (1) each point in F can be reached by a ray starting at and orthogonal to δ, and (2) δ touches F on one side and points in $f \setminus F$ on the other side. When F is clear from context, we simply use baseline for brevity.

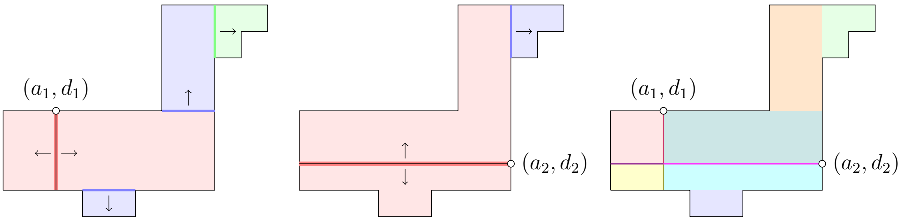


Figure 8 Sectors with respect to (a) the first port; (b) the second port; (c) $\mathcal{P}_2$. For a sector of each color, the segment on the border highlighted with the same color is its baseline; for (c) different sectors have different colors, and notice that at the intersection of the rays from (a_1, d_1) and (a_2, d_2) there is also a single point sector.

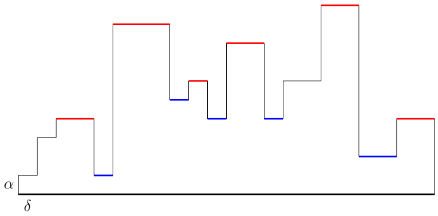


Figure 9 The segments colored red (blue) are local maxima (minima).

▶ **Lemma 18.** *Each sector in $\mathcal{F}_t$, $1 \leq t \leq q$, admits at least one baseline.*

The existence of a baseline is already quite helpful to obtain the desired bound on the treewidth, but not yet sufficient on its own. In particular, this implies that each sector has the shape of a histogram. Next, we show that the bend distances to any "additional port" cannot differ too much within a sector.

▶ **Lemma 19.** *For every sector $F \in \mathcal{F}_t$, $t \in [1, q-1]$, and every pair $F_1, F_2 \in \mathcal{U}_F^{t+1}$, $|\operatorname{bd}(p, (a_{t+1}, d_{t+1})) - \operatorname{bd}(q, (a_{t+1}, d_{t+1}))| \leq 3$ for every pair of points $p \in F_1$, $q \in F_2$.*

With Lemmas 18 and 19, we are ready to proceed to the most difficult part of establishing our bound on the treewidth of the sector graph. Let us fix some F-baseline δ for a sector F in the sector graph $\mathcal{G}_t$, $1 \leq t \leq q$. Consider the polyline α obtained when traversing F in clockwise fashion from one endpoint of δ to the other, where α does not intersect δ. We call a line-segment in α a *local maximum* (*minimum*) if α makes a right (left) turn both before and after the line-segment (see Figure 9). Let $\xi_{\max}(F)$ ($\xi_{\min}(F)$) denote the number of local maxima (local minima) in F; note that since each sector is a histogram, $\xi_{\max}(F) = \xi_{\min}(F) + 1$.

▶ **Lemma 20.** *For every sector $F \in \mathcal{F}_t$, $1 \leq t \leq q-1$, we have $|\mathcal{U}_F^{t+1}| \leq 4 + \xi_{\max}(F)$ and $\max_{F' \in \mathcal{U}_F^{t+1}} \xi_{\max}(F') \leq \xi_{\max}(F)$.*

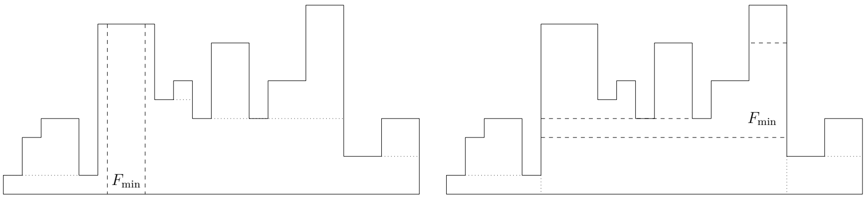


Figure 10 Cases of relative location of the $F_{\min}$ sector in F relative to the F-baseline, Lemma 20.

To obtain the main result of this section (Theorem 22), we will combine Lemma 20 with the following lemma that bounds the number of local maxima in each sector.

▶ **Lemma 21.** *For each sector F in $V(\mathcal{G})$, $\xi_{\max}(F) \leq 4k$.*

▶ **Theorem 22.** *Let $\mathcal{G}$ be a sector graph of a face f of the drawing $\Gamma(G)$. Then $\mathrm{tw}(\mathcal{G}) \leq (4+4k)^{4k}$.*

Proof Sketch. We prove the claim by induction on the number of port candidates for f, where the base of an induction exactly follows from the result of Lemma 17. For the inductive step, we assume that $\mathrm{tw}(\mathcal{G}_t) = \mathcal{O}(k^t)$ and our aim will be to show that $\mathrm{tw}(\mathcal{G}_{t+1})$ is $\mathcal{O}(k^{t+1})$. To do so, we replace each occurrence of a sector v in a bag with all of the sectors in $\mathcal{U}_F^{t+1}$. ◀

6.2 The Final Step

At this point, we have shown that an instance $\mathcal{I} = \langle G_f, H_f, \Gamma(H_f), f, \mathcal{P}\rangle$ with $k = |V(G_f) \setminus V(H_f)|$ of F-BMOE admits a sector graph $\mathcal{G}$ of treewidth at most $(4+4k)^{4k}$ (Theorem 22), and that a bend-minimal extension of $\Gamma(H_f)$ to an orthogonal planar drawing of G_f can be assumed to only contain feature points on the sector-grid points as per Corollary 16, of which there are at most `gridsize`(k) many per sector. This allows us to proceed to the final ingredient for our algorithm:

▶ **Lemma 23.** *F-BMOE can be solved in time $2^{k^{\mathcal{O}(1)}} \cdot |V(G_f)|$.*

Proof Sketch. Thanks to Theorem 22, we can use known results to compute a nice tree decomposition (T, χ) of $\mathcal{G}$ of small width. Next we design a dynamic program that runs along T and at each point stores all possible options of how a hypothetical bend-minimal extension can intersect the sector-grid points of the sectors in the current bag. ◀

By combining Lemma 23 with Lemma 3 and Observation 13, we conclude:

▶ **Corollary 24.** *BMOE can be solved in time $2^{\kappa^{\mathcal{O}(1)}} \cdot n$, where n is the number of feature points of $\Gamma(H)$.*

7 Concluding Remarks

We have established the fixed-parameter tractability of the extension problem for bend-minimal orthogonal drawings, marking a notable addition to our understanding of drawing extension problems. What distinguishes this result from some of its predecessors on, e.g., extending 1-planar [17], simple k-planar [21] or crossing-minimal [23] drawings, is that these examples were topological while orthogonal planar drawings are geometric in nature. We believe this is one of the reasons why it seems impossible to use previously developed techniques in our setting, a fact which inspired the development of a novel machinery that we believe will find applications beyond the specific context of the problem studied here.

As an example of this, a minor adjustment of our technique is already sufficient to also obtain a fixed-parameter algorithm for the problem of extending an orthogonal planar drawing while preserving a bound δ on the number of bends per edge [5, 7] parameterized by $\kappa + \delta$. But the technique could also possibly be applied to more general drawing styles, such as extending drawings restricted to boundedly many allowed edge slopes [24, 26].

References

1 Patrizio Angelini, Giuseppe Di Battista, Fabrizio Frati, Vít Jelínek, Jan Kratochvíl, Maurizio Patrignani, and Ignaz Rutter. Testing planarity of partially embedded graphs. *ACM Trans. Algorithms*, 11(4):32:1–32:42, 2015. doi:10.1145/2629341.

2 Patrizio Angelini, Ignaz Rutter, and T. P. Sandhya. Extending partial orthogonal drawings. *J. Graph Algorithms Appl.*, 25(1):581–602, 2021. doi:10.7155/jgaa.00573.

3 Alan Arroyo, Fabian Klute, Irene Parada, Birgit Vogtenhuber, Raimund Seidel, and Tilo Wiedera. Inserting one edge into a simple drawing is hard. *Discrete & Computational Geometry*, pages 1–26, 2022. doi:10.1007/978-3-030-60440-0_26.

4 Giuseppe Di Battista, Giuseppe Liotta, and Francesco Vargiu. Spirality and optimal orthogonal drawings. *SIAM J. Comput.*, 27(6):1764–1811, 1998. doi:10.1137/S0097539794262847.

5 Therese Biedl and Goos Kant. A better heuristic for orthogonal graph drawings. *Comput. Geom. Theory Appl.*, 9(3):159–180, 1998. doi:10.1016/S0925-7721(97)00026-6.

6 Therese Biedl, Anna Lubiw, Mark Petrick, and Michael J. Spriggs. Morphing orthogonal planar graph drawings. *ACM Trans. Algorithms*, 9(4):29, 2013. doi:10.1145/2500118.

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. doi:10.1145/2838736.

8 Guido Brückner, Ignaz Rutter, and Peter Stumpf. Extending partial representations of circle graphs in near-linear time. In *47th International Symposium on Mathematical Foundations of Computer Science, MFCS 2022, August 22-26, 2022, Vienna, Austria*, volume 241 of *LIPIcs*, pages 25:1–25:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.MFCS.2022.25.

9 Guido Brückner and Ignaz Rutter. Partial and constrained level planarity. In *Discrete Algorithms (SODA'17)*, pages 2000–2011. SIAM, 2017. doi:10.1137/1.9781611974782.130.

10 Erin W. Chambers, David Eppstein, Michael T. Goodrich, and Maarten Löffler. Drawing graphs in the plane with a prescribed outer face and polynomial area. *J. Graph Algorithms Appl.*, 16(2):243–259, 2012. doi:10.7155/jgaa.00257.

11 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

12 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.

13 Emilio Di Giacomo, Giuseppe Liotta, and Fabrizio Montecchiani. Orthogonal planarity testing of bounded treewidth graphs. *J. Comput. Syst. Sci.*, 125:129–148, 2022. doi:10.1016/j.jcss.2021.11.004.

14 Walter Didimo and Giuseppe Liotta. Computing orthogonal drawings in a variable embedding setting. In Kyung-Yong Chwa and Oscar H. Ibarra, editors, *Algorithms and Computation (ISAAC'98)*, volume 1533 of *LNCS*, pages 79–88. Springer, 1998. doi:10.1007/3-540-49381-6_10.

15 Walter Didimo, Giuseppe Liotta, Giacomo Ortali, and Maurizio Patrignani. Optimal orthogonal drawings of planar 3-graphs in linear time. In *Discrete Algorithms (SODA'20)*, pages 806–825. SIAM, 2020. doi:10.1137/1.9781611975994.49.

16 Christian A. Duncan and Michael T. Goodrich. Planar orthogonal and polyline drawing algorithms. In Roberto Tamassia, editor, *Handbook of Graph Drawing and Visualization*, chapter 7, pages 223–246. CRC Press, 2013.

17 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending nearly complete 1-planar drawings in polynomial time. In Javier Esparza and Daniel Král', editors, *Mathematical Foundations of Computer Science (MFCS'20)*, volume 170 of *LIPIcs*, pages 31:1–31:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.MFCS.2020.31.

18 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending partial 1-planar drawings. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors,

Automata, Languages, and Programming (ICALP'20), volume 168 of *LIPIcs*, pages 43:1–43:19. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.43.

19 Markus Eiglsperger, Sándor P. Fekete, and Gunnar W. Klau. Orthogonal graph drawing. In Michael Kaufmann and Dorothea Wagner, editors, *Drawing Graphs: Methods and Models*, volume 2025 of *LNCS*, chapter 6, pages 121–171. Springer-Verlag, 2001. doi:10.1007/3-540-44969-8_6.

20 Jirí Fiala, Ignaz Rutter, Peter Stumpf, and Peter Zeman. Extending partial representations of circular-arc graphs. In *Graph-Theoretic Concepts in Computer Science - 48th International Workshop, WG 2022, Tübingen, Germany, June 22-24, 2022, Revised Selected Papers*, volume 13453 of *Lecture Notes in Computer Science*, pages 230–243. Springer, 2022. doi:10.1007/978-3-031-15914-5_17.

21 Robert Ganian, Thekla Hamm, Fabian Klute, Irene Parada, and Birgit Vogtenhuber. Crossing-optimal extension of simple drawings. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021*, volume 198 of *LIPIcs*, pages 72:1–72:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.72.

22 Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001. doi:10.1137/S0097539794277123.

23 Thekla Hamm and Petr Hlinený. Parameterised partially-predrawn crossing number. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry, SoCG 2022*, volume 224 of *LIPIcs*, pages 46:1–46:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.SoCG.2022.46.

24 Udo Hoffmann. On the complexity of the planar slope number problem. *J. Graph Algorithms Appl.*, 21(2):183–193, 2017. doi:10.7155/jgaa.00411.

25 Vít Jelínek, Jan Kratochvíl, and Ignaz Rutter. A Kuratowski-type theorem for planarity of partially embedded graphs. *Comput. Geom. Theory Appl.*, 46(4):466–492, 2013. doi:10.1016/j.comgeo.2012.07.005.

26 Balázs Keszegh, János Pach, and Dömötör Pálvölgyi. Drawing planar graphs of bounded degree with few slopes. *SIAM J. Discrete Math.*, 27(2):1171–1183, 2013. doi:10.1137/100815001.

27 Giordano Da Lozzo, Giuseppe Di Battista, and Fabrizio Frati. Extending upward planar graph drawings. *Comput. Geom. Theory Appl.*, 91:101668, 2020. doi:10.1016/j.comgeo.2020.101668.

28 Tamara Mchedlidze, Martin Nöllenburg, and Ignaz Rutter. Extending convex partial drawings of graphs. *Algorithmica*, 76(1):47–67, 2016. doi:10.1007/s00453-015-0018-6.

29 Takao Nishizeki and Md. Saidur Rahman. *Planar Graph Drawing*, volume 12 of *Lecture Notes Series on Computing*. World Scientific, 2004. doi:10.1142/5648.

30 Maurizio Patrignani. On extending a partial straight-line drawing. *Int. J. Found. Comput. Sci.*, 17(5):1061–1070, 2006. doi:10.1142/S0129054106004261.

31 Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987. doi:10.1137/0216030.