Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 25, no. 1, pp. 581–602 (2021)
DOI: 10.7155/jgaa.00573

Extending Partial Orthogonal Drawings

Patrizio Angelini[1] *Ignaz Rutter*[2] *Sandhya T P*[2]

[1]John Cabot University, Rome, Italy
[2]University of Passau, Passau, Germany

Submitted: April 2021	Reviewed: June 2021	Revised: August 2021
Accepted: September 2021	Final: October 2021	Published: October 2021
Article type: Regular paper		Communicated by: G. Liotta

Abstract. We study the planar orthogonal drawing style within the framework of partial representation extension. Let (G, H, Γ_H) be a partial orthogonal drawing, i.e., G is a graph, $H \subseteq G$ is a subgraph, Γ_H is a planar orthogonal drawing of H, and $|\Gamma_H|$ is the number of vertices and bends in Γ_H.

We show that the existence of an orthogonal drawing Γ_G of G that extends Γ_H can be tested in linear time. If such a drawing exists, then there is also one that uses $O(|\Gamma_H|)$ bends per edge. On the other hand, we show that it is NP-complete to find an extension that minimizes the number of bends or has a fixed number of bends per edge.

1 Introduction

One of the most popular drawing styles are *orthogonal drawings*, where vertices are represented by points and edges are represented by chains of horizontal and vertical segments connecting their endpoints. Such a drawing is *planar* if no two edges share an interior point. An interior point of an edge where a horizontal and a vertical segment meet is called a *bend*. The main aesthetic criterion for planar orthogonal drawings is the number of bends on the edges.

A large body of literature is devoted to optimizing the number of bends in planar orthogonal drawings. The complexity of the problem strongly depends on the particular input. If the combinatorial embedding can be chosen freely, then it is NP-complete to decide whether there exists a drawing without bends [18]. If the input graph comes with a fixed combinatorial embedding, then a bend-optimal drawing that preserves the given embedding can be computed efficiently by a classical result of Tamassia [27]. A recent trend has been to investigate under which conditions

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – RU 1903/3-1.
A preliminary version of this paper appeared in [3].

E-mail addresses: pangelini@johncabot.edu (Patrizio Angelini) rutter@fim.uni-passau.de (Ignaz Rutter) thekkumpad@fim.uni-passau.de (Sandhya T P)

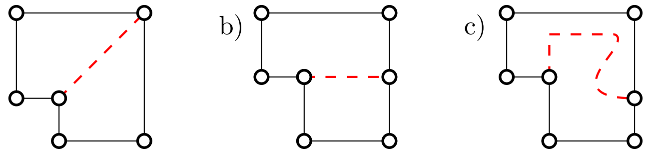

Figure 1: An instance of the partial representation extension problem (G, H, Γ_H) is given. The graph H is solid black and the edges of $E(G) \setminus E(H)$ are dashed red. (a) (G, H, Γ_H) admits a planar extension, but not an orthogonal extension. (b) (G, H, Γ_H) admits an orthogonal extension with no bends (c) An orthogonal representation of G (the curved part of the dashed edge has no bends) that extends the description of the solid black drawing of H. There exists no drawing of G with this representation that extends the given drawing of H.

the variable-embedding case becomes tractable. For maxdeg-3 graphs a bend-optimal drawing can be computed efficiently [10], which has recently been improved to linear time [12]. The problem is also FPT with respect to the number of degree-4 vertices [11], and if one discounts the first bend on each edge, an optimal solution can be computed even for individual convex cost functions on the edges [4, 5]. We refer to the survey [13] for further references. In light of this popularity and the existence of a strongly developed theory, it is surprising that planar orthogonal drawings have not been investigated within the framework of partial representation extension. Especially so, since it has been considered in the related context of simultaneous representations [1].

In the partial representation extension problem, the input graph G comes together with a subgraph $H \subseteq G$ and a representation (drawing) Γ_H of H. One then seeks a drawing Γ_G of G that *extends* Γ_H, i.e., whose restriction to H coincides with Γ_H. The partial representation extension problem has recently been considered for a large variety of different types of representations. For planar straight-line drawings, it is NP-complete [26], whereas for topological drawings there exists a linear-time algorithm [2], as well as a characterization via forbidden substructures [19]. Moreover, it is known that, if a topological drawing extension exists, then it can be drawn with polygonal curves such that each edge has a number of bends that is linear in the complexity of Γ_H [6]. Here the complexity of Γ_H is the number of vertices and bends in Γ_H. Most recently the problem has been investigated in the context of 1-planarity [14]. Besides classical drawing styles, it has also been studied for contact representations [7] and for geometric intersection representations, e.g., for (proper/unit) interval graphs [20, 22], chordal graphs [21], circle graphs [8], and trapezoid graphs [24].

In this paper, we provide an in-depth study of partial representation extension problems for the planar orthogonal drawing style. Since the aesthetics are of particular importance for the quality of such a drawing, we put a major emphasis on extension questions in relation to the number of bends. It is worth noting that even the seminal work of Tamassia [27] already mentions the idea of preserving the shape of a given subgraph by maintaining its orthogonal representation, i.e., a description of the angles around each vertex and the directions of the bends along each edge, via modifications in his flow network. However, this approach only preserves the shape of the subgraph as described by an orthogonal representation, and not necessarily its drawing. Fig. 1 shows that there are partial planar orthogonal drawings that can be extended in a planar way, but not orthogonally (Fig. 1a). We also provide an example that can be extended orthogonally while maintaining planarity (Fig. 1b). Moreover, even if an orthogonal representation O_G of G preserves a given orthogonal representation O_H of a drawing Γ_H of H, as in the formulation of Tamassia, a planar drawing Γ_G of G realizing O_G that extends Γ_H does not necessarily exist (Fig. 1c).

Contribution and Outline. After presenting preliminaries in Section 2, we give a linear-time algorithm for deciding the existence of an orthogonal drawing extension in Section 3. Then, we consider the realizability problem, where we are given an orthogonal extension in the form of a suitable planar embedding, and we seek an orthogonal drawing extension that optimizes the number of bends. Along the lines of a result by Chan et al. [6], we show that there always exists an orthogonal drawing extension such that each edge has a number of bends that is linear in the complexity of Γ_H in Section 4. We complement these findings in Section 5 by showing that it is NP-hard to minimize the number of bends and NP-complete to test whether there exists an orthogonal drawing extension with a fixed number of bends per edge.

2 Preliminaries

We call the circular clockwise ordering of the edges around a vertex v in an embedding the *rotation* at v. Let $G = (V, E)$ be a simple undirected graph and let $H \subseteq G$ be a subgraph. We refer to the vertices and edges of H as H*-vertices* and H*-edges*, respectively. Similarly, we refer to the vertices of $V(G) \setminus V(H)$ and to the edges of $E(G) \setminus E(H)$ as G*-vertices* and G*-edges*, respectively. We denote $|\Gamma_H|$ by the number of vertices and bends in Γ_H. For a connected graph, the *facial walk* of a face is the closed walk that consists of all the vertices and edges incident to the face.

Let (G, H, Γ_H) be a triple composed of a graph G, a subgraph $H \subseteq G$, and a planar orthogonal drawing Γ_H of H. We denote by RepExt(ortho) (RepExt stands for representation extension) the problem of testing whether G admits a planar orthogonal drawing Γ_G that extends Γ_H. In Γ_H, we say that an H-edge is *attached to* one of the four *ports* (*top, bottom, left, right*) of its end vertices. If there is no H-edge attached to a port of a vertex, then this port is *free*; note that the free ports are those at which the G-edges can be attached in Γ_G. For two H-edges e and e' that are consecutive in the rotation at a vertex v in Γ_H, we denote by $\mathcal{P}_H(e, e') = k$ the fact that there exist exactly k free ports of v when moving from e to e' in clockwise order around their common endvertex. For example, in Fig. 2(c), the left and right ports are free, while the others are not, and we have $\mathcal{P}_H(e_1, e_2) = 1$ and $\mathcal{P}_H(e_2, e_1) = 1$. We call $\mathcal{P}_H(e, e') = k$ a *port constraint*, and we denote by $\mathcal{P}_H$ the set of all port constraints in Γ_H. Note that, for a vertex v with rotation $e_1, \dots, e_h$ in Γ_H, with $h \le 4$, we have $\sum_{i=1}^{h} \mathcal{P}_H(e_i, e_{i+1}) = 4 - \deg(v)$, where we define $e_{h+1} := e_1$.

We now show that to solve an instance (G, H, Γ_H) of the RepExt(ortho) problem, it suffices to only consider the port constraints determined by Γ_H together with the embedding $\mathcal{E}_H$ of H in Γ_H. More specifically, we prove the following characterization, which could also be deduced from [1].

Theorem 1 *Let (G, H, Γ_H) be an instance of* RepExt(ortho)*. Let $\mathcal{E}_H$ be the embedding of H in Γ_H, and let $\mathcal{P}_H$ be the port constraints induced by Γ_H. Then, (G, H, Γ_H) admits an orthogonal drawing extension if and only if G admits a planar embedding $\mathcal{E}_G$ that extends $\mathcal{E}_H$ and such that, for every port constraint $\mathcal{P}_H(e, e') = k$, there exist at most k G-edges between e and e' in the rotation at v in $\mathcal{E}_G$, where v is the common vertex of the H-edges e and e'.*

Proof: One direction is trivial; namely, if there exists an orthogonal drawing Γ_G of G that extends Γ_H, then the embedding of G in Γ_G satisfies the two properties by construction. Suppose now that there exists an embedding $\mathcal{E}_G$ of G that satisfies the two properties. Since $\mathcal{E}_G$ is planar and extends $\mathcal{E}_H$, we can route each G-edge uv as an arbitrary curve, while respecting the rotation at u and v in $\mathcal{E}_G$, without crossing any other edge. Also, the fact that $\mathcal{E}_G$ satisfies the port constraints in $\mathcal{P}_H$ implies that, for each G-edge uv, we can assign free ports of u and v to uv, in such a way

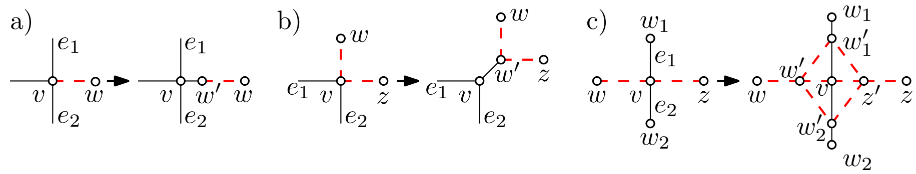

Figure 2: Gadgets for H-vertices.

that no port is assigned to more than one edge. Thus, by approximating the curve representing each G-edge uv with an orthogonal polyline, it is possible to construct an orthogonal drawing of G extending Γ_H. Note that in Theorem 5 we will prove that this can even be done by using orthogonal polylines with a limited number of bends. $\square$

In view of Theorem 1, we define a new problem, called RepExt(top+port), which is linear-time equivalent to RepExt(ortho). Here, top stands for *topological* drawing, i.e., one in which edges are represented as Jordan arcs. An instance of this problem is a 4-tuple $(G, H, \mathcal{E}_H, \mathcal{P}_H)$ and the goal is to test whether G admits an embedding $\mathcal{E}_G$ that satisfies the conditions of Theorem 1. In order to unify the terminology, we also refer to the *Partially Embedded Planarity* problem studied in [2] as RepExt(top). Recall that an instance of this problem is a triple $(G, H, \mathcal{E}_H)$, and the goal is to test whether G admits an embedding $\mathcal{E}_G$ that extends $\mathcal{E}_H$. As proved in [2], RepExt(top) can be solved in linear time.

3 Testing Algorithm

In this section, we show that RepExt(ortho) can be solved in linear time. By Theorem 1, it suffices to prove that RepExt(top+port) can be solved in linear time. The algorithm is based on a linear-time reduction to RepExt(top), which is known to be linear-time solvable [2]. Namely, starting from an instance $(G, H, \mathcal{E}_H, \mathcal{P}_H)$ of RepExt(top+port), we construct an instance $(G', H', \mathcal{E}_{H'})$ of RepExt(top) that admits a solution if and only if $(G, H, \mathcal{E}_H, \mathcal{P}_H)$ does.

We start by initializing $G' = G$, $H' = H$, and $\mathcal{E}_{H'} = \mathcal{E}_H$. Then, for each vertex v such that $1 < \deg_H(v) < \deg_G(v)$, we modify the instance as described in the following; see Fig. 2.

Case 1: Suppose first that $\deg_H(v) = 3$ and $\deg_G(v) = 4$, and let $e = vw$ be the unique G-edge incident to v; refer to Fig. 2(a). Since $\deg_H(v) = 3$, there exist exactly two H-edges e_1 and e_2 such that e_1 immediately precedes e_2 in the rotation at v in $\mathcal{E}_H$ and $\mathcal{P}_H(e_1, e_2) = 1$. Note that, to respect the port constraint, we have to guarantee that e is placed between e_1 and e_2 in the rotation at v in $\mathcal{E}_G$. For this, we subdivide e with a new H'-vertex w', that is, we remove e from G', and we add the vertex w' and the edges vw' and $w'w$ to G'. Also, we add w' and vw' to H', and embed vw' in between e_1 and e_2 in the rotation at v.

Case 2: Suppose now that $\deg_H(v) = 2$ and $\deg_G(v) \geq 3$. Let e_1 and e_2 be the two H-edges incident to v, and let $e = vw$ and $e^* = vz$ be the at most two G-edges incident to v. We distinguish two cases, based on whether $\mathcal{P}_H(e_1, e_2) = 2$ and $\mathcal{P}_H(e_2, e_1) = 0$ (or vice versa), or $\mathcal{P}_H(e_1, e_2) = \mathcal{P}_H(e_2, e_1) = 1$.

Case 2.a: If $\mathcal{P}_H(e_1, e_2) = 2$, then we need to guarantee that both e and e^* (if it exists) are placed between e_1 and e_2 in the rotation at v in $\mathcal{E}_G$; refer to Fig. 2(b). For this, we remove e and e^* from G', and we add a new vertex w' and the edges vw', $w'w$, and $w'z$ to G'. Also, we add w'

and vw' to H', and insert vw' between e_1 and e_2 in the rotation at v in $\mathcal{E}_{H'}$. Note that, if e^* does not exist, this is the same procedure as in the previous case.

Case 2.b: If $\mathcal{P}_H(e_1, e_2) = \mathcal{P}_H(e_2, e_1) = 1$, then we have to place e and e^* (if it exists) with respect to the path composed of the edges e_1 and e_2; refer to Fig. 2(c). Note that, if e^* does not exist, then e can be on any of the two sides of this path, and thus in this case we do not perform any modification. If e^* exists, then we just have to guarantee that e and e^* lie on different sides of the path. For this, we subdivide e, e^*, e_1, and e_2 with a new vertex each, that is, we remove these edges from G' (e_1 and e_2 also from H'), and we add four new vertices w', z', w'_1, and w'_2 to G'. Also, we add to G' the edges vw', vz', vw'_1, and vw'_2, and the edges $w'w$, $z'z$, w'_1w_1, and w'_2w_2, where w_1 and w_2 are the endpoints of e_1 and e_2, respectively, different from v. Further, we add the edges $w'w'_1$, w'_1z', $z'w'_2$, and w'_2w' to G'. Finally, we add the vertices w'_1 and w'_2, and the edges vw'_1, w'_1w_1, vw'_2, and w'_2w_2 also to H'; in $\mathcal{E}_{H'}$, we place w'_1w_1 and w'_2w_2 in the rotations at w_1 and at w_2, respectively, in the same position as e_1 and e_2, respectively, in $\mathcal{E}_H$. The rotations at v, w'_1, and w'_2 in $\mathcal{E}_{H'}$ do not need to be set, since each of these vertices has at most two incident H'-edges. The above construction leads to the following lemma.

Lemma 1 *The instance $(G', H', \mathcal{E}_{H'})$ has an embedding extension if and only if $(G, H, \mathcal{E}_H, \mathcal{P}_H)$ has an embedding extension satisfying the port constraints.*

Proof: Suppose that $(G', H', \mathcal{E}_{H'})$ admits an embedding extension, and let $\mathcal{E}_{G'}$ be the corresponding embedding of G'. We construct an embedding $\mathcal{E}_G$ of G that determines an embedding extension of $(G, H, \mathcal{E}_H, \mathcal{P}_H)$ satisfying the port constraints, as follows. Let v be any vertex of G. By construction, v is also a vertex of G'.

Suppose first that all the neighbors of v in G' also belong to G, that is, none of the described modifications has been applied to v. By construction, this is the case only if v satisfies the conditions of neither Case 1 nor Case 2, or if it satisfies the condition of Case 2.b, but edge e^* does not exist. In the former case, we have that in $(G, H, \mathcal{E}_H, \mathcal{P}_H)$ either there exists no G-edge incident to v, or there exists at most one H-edge incident to v. If there exists no G-edge incident to v, the rotation at v in $\mathcal{E}_{H'}$ is the same as the one in $\mathcal{E}_H$, and there exists no port constraint at v. On the other hand, if there exists at most one H-edge incident to v, there exists no port constraint at v in $\mathcal{P}_H$, by definition, and every rotation at v in $\mathcal{E}_G$ trivially extends the rotation at v in $\mathcal{E}_H$. Thus, in this case, we set the rotation at v in $\mathcal{E}_G$ to be the same as the one in $\mathcal{E}_{G'}$. In the latter case, when Case 2.b applies but edge e^* does not exist, we already argued that any rotation at v in $\mathcal{E}_G$ satisfies the port constraints and extends the rotation at v in $\mathcal{E}_H$, so we can again set the rotation at v in $\mathcal{E}_G$ to be the same as the one in $\mathcal{E}_{G'}$.

Suppose then that there exists exactly one neighbor of v in G' that does not belong to G. Then, by construction, this neighbor of v is the vertex w' that we introduced in one of the first two cases we described above. Namely, either it holds that $\deg_H(v) = 3$ and $\deg_G(v) = 4$, or it holds that $\deg_H(v) = 2$, $\deg_G(v) \geq 3$, $\mathcal{P}_H(e_1, e_2) = 2$, and $\mathcal{P}_H(e_2, e_1) = 0$, where e_1 and e_2 are the H-edges incident to v. In both cases, we obtain the rotation at v in $\mathcal{E}_G$ by contracting the edge vw', and by merging the rotations at v and at w' in $\mathcal{E}_{G'}$. This guarantees that the rotation at v in $\mathcal{E}_G$ extends the rotation at v in $\mathcal{E}_H$ and that the port constraint $\mathcal{P}_H(e_1, e_2) = 1$ (resp. $\mathcal{P}_H(e_1, e_2) = 2$) at v is satisfied, since the edge vw (resp. the edges vw and vz) appears between e_1 and e_2 in $\mathcal{E}_G$.

Suppose finally that there exists more than one neighbor of v in G' that does not belong to G. Then, by construction, $\deg_{G'}(v) = 4$, and the four neighbors of v in G' are the ones that we introduced in Case 2.b when e^* exists. Namely, v is incident to two H-edges e_1 and e_2, and $\mathcal{P}_H(e_1, e_2) = \mathcal{P}_H(e_2, e_1) = 1$. Observe that, since the subgraph of G' induced by the vertices v, w', z', w'_1, and w'_2 is triconnected, the vertices w', w'_1, z', w'_2 appear in the rotation at v in $\mathcal{E}_{G'}$

either in this order or in its reverse. In the former case (the latter being analogous), we set the rotation at v in $\mathcal{E}_G$ so that w, w_1, z, w_2 appear in this order. This trivially extends the rotation at v in $\mathcal{E}_H$, since $\deg_H(v) = 2$, and guarantees that the port constraints at v are satisfied, since w and z use non-consecutive ports of v.

We further observe that, due to our transformation, the cycles in H bijectively correspond to the cycles in H', and that a vertex lies inside a cycle in $\mathcal{E}_H$ if and only if it lies inside the corresponding cycle in $\mathcal{E}_{H'}$. Together with the above discussion, this implies that $\mathcal{E}_G$ extends $\mathcal{E}_H$, since $\mathcal{E}_{G'}$ extends $\mathcal{E}_{H'}$. Finally, since G' contains G as a minor, the fact that $\mathcal{E}_{G'}$ is a planar embedding implies that $\mathcal{E}_G$ is a planar embedding, which concludes the proof of this direction.

The proof for the other direction is analogous. In fact, given a planar embedding $\mathcal{E}_G$ of G that is a solution for the instance $(G, H, \mathcal{E}_H, \mathcal{P}_H)$, we can construct a planar embedding $\mathcal{E}_{G'}$ of G' that determines an embedding extension of $(G', H', \mathcal{E}_{H'})$, as follows.

Let v be any vertex of G'. If v is also a vertex of G and all the neighbors of v in G' also belong to G, then we can set the rotation at v in $\mathcal{E}_{G'}$ to be the same as the one in $\mathcal{E}_G$, as discussed above. To cover all the other cases (either v or at least one of its neighbors is not a vertex of G), it is enough to consider the three cases in the construction we described above.

In the first two cases, the fact that $\mathcal{E}_G$ satisfies the port constraint $\mathcal{P}_H(e_1, e_2) = 1$ (resp. $\mathcal{P}_H(e_1, e_2) = 2$) implies that vw (resp. both vw and vz) appears between e_1 and e_2 in the rotation at v in $\mathcal{E}_G$. Thus, inserting vw' in the rotation at v in $\mathcal{E}_{G'}$ in the same position as vw (resp. both vw and vz) in the rotation at v in $\mathcal{E}_G$ yields a rotation at v in $\mathcal{E}_{G'}$ that extends the one at v in $\mathcal{E}_{H'}$. The same trivially holds for the rotation at w', since $\deg_{H'}(w') = 1$.

In the last case, when v is incident to two H-edges e_1 and e_2, and $\mathcal{P}_H(e_1, e_2) = \mathcal{P}_H(e_2, e_1) = 1$, the fact that $\mathcal{E}_G$ satisfies the port constraints implies that the vertices w, w_1, z, w_2 appear in the rotation at v in $\mathcal{E}_G$ either in this order or in its reverse. In both cases, it is possible to set the rotations at v, w', w'_1, z', w'_2 in $\mathcal{E}_{G'}$ so that the triconnected subgraph induced by these vertices is embedded according to its unique planar embedding, and all the vertices of G', except for v, lie outside of the cycle induced by w', w'_1, z', w'_2. Note that each of these five vertices is incident to at most two H'-edges, and thus every of its rotations in $\mathcal{E}_{G'}$ trivially extends the one in $\mathcal{E}_{H'}$. This concludes the proof of the lemma. □

Theorem 2 *The* RepExt(top+port) *problem can be solved in linear time.*

Proof: Given an instance $I = (G, H, \mathcal{E}_H, \mathcal{P}_H)$ of RepExt(top+port), we construct the instance $I' = (G', H', \mathcal{E}_{H'})$ of RepExt(top) that has linear size as described above. This takes $O(1)$ time per vertex, and hence total linear time. By Lemma 1, I has a solution if and only if I' has one. Since the existence of a solution of I' can be tested in linear time [2], the statement follows. □

As a consequence of Theorems 1 and 2, we conclude the following.

Theorem 3 *The* RepExt(ortho) *problem can be solved in linear time.*

4 Realizability with Bounded Number of Bends

In this section we prove that, if there exists an orthogonal drawing extension for an instance (G, H, Γ_H) of RepExt(ortho), then there also exists one in which the number of bends per edge is linear in the complexity of the drawing Γ_H. By subdividing H at the bends of Γ_H, we can assume that Γ_H is a bend-free drawing of H. To achieve the desired edge complexity, it then suffices to show that $O(|\Gamma_H|)$ bends per edge suffice. This result can be considered as the counterpart for the orthogonal setting of the one by Chan et al. [6] for the polyline setting. In their work, in fact, they show that a positive instance (G, H, Γ_H) of the RepExt(top) problem can always be realized with at most $O(|V(H)|)$ bends per edge when Γ_H is a planar straight-line drawing of H.

Our approach follows the algorithm given in [6], with a main technical difference which is due to the peculiar properties of orthogonal drawings. Their algorithm first constructs a planar supergraph G' of G that is Hamiltonian using a method of Pach et al. [25, Lemma 5]. The main step of the algorithm of Chan et al. [6] involves the *contraction* of some edges of G' [6, Lemma 3]. This operation identifies the two end-vertices of the contracted edge and merges their adjacency lists. However, both the construction of the supergraph G' and the contractions may produce vertices of degree greater than 4, which implies that the resulting graph does not admit an orthogonal drawing any longer. As such, these operations are not suitable for the realization of orthogonal drawings. In order to overcome this problem, we consider instead the *Kandinsky* model [16], which extends the orthogonal drawing model to also allow for vertices of large degree. Once the drawing has been computed, we remove the previously added parts and by adding a small amount of additional bends on the G-edges, we arrive at a orthogonal drawing of the initial graph G. More specifically, we prove the following theorem.

Theorem 4 *Let (G, H, Γ_H) be an instance of* RepExt(ortho)*. Suppose that G admits a planar orthogonal drawing Γ_G that extends Γ_H, and let $\mathcal{E}_G$ be the embedding of G in Γ_G. Then we can construct a planar Kandinsky drawing of G in $O(n^2)$-time, where n is the number of vertices of G, that realizes $\mathcal{E}_G$, extends Γ_H, and has at most $192|\Gamma_H|$ bends per edge.*

An overview of the algorithm to construct the desired Kandinsky orthogonal drawing Γ_G^* of G, whose main steps follow the method in [6], is given below.

Step 1: Consider a face F of Γ_H with facial walks $W_1, W_2, \ldots, W_k$. Construct an ε-approximation of F and let W_i' be the orthogonal polygon that approximates W_i, $1 \leq i \leq k$. Let F' be the face bounded by the approximated boundary components of F; refer to Lemma 2, and to Fig. 3b.

Step 2: Partition F' into rectangles [15] and construct a graph K by placing a vertex at the center of each rectangle and by joining the vertices of adjacent rectangles. Let T be a spanning tree of K. For each facial walk W_i, add a new vertex near W_i as a leaf of T (see Fig. 3c).

Step 3: Construct the multigraph G_F induced by the vertices lying inside or on the boundary of F and by contracting each facial walk of F to a single vertex. Then draw G_F along T. Now, reconstruct the edges of $G \setminus H$ and the edges between G_F and other components of G inside F. Refer to Lemma 6 and to Fig. 4.

Let Γ_G^* be the resulting Kandinsky orthogonal drawing and we then transform Γ_G^* into an orthogonal drawing Γ_G of G with $O(|\Gamma_H|)$ bends per edge that extends Γ_H. An illustration is given in Fig. 8.

To prove Theorem 4, we first need a couple of tools and we present those tools as lemmas before delving into the actual proof of the theorem.

Since the embedding of G is fixed, it is enough to consider a face F of Γ_H and prove Theorem 4 for that particular face. We first show how to construct an inner ε-approximating orthogonal polygon for each facial walk of F using a technique similar to the one from [6]. We adopt the definition of an inner ε-approximation of a facial walk from [6] which is defined as follows. Let W_1 be the outer facial walk of a face F and $W_2, \ldots, W_k$ be inner facial walks. An inner ε-approximation of W_i is a simple polygon P_i (a closed polygonal arc with no self-intersections) such that:

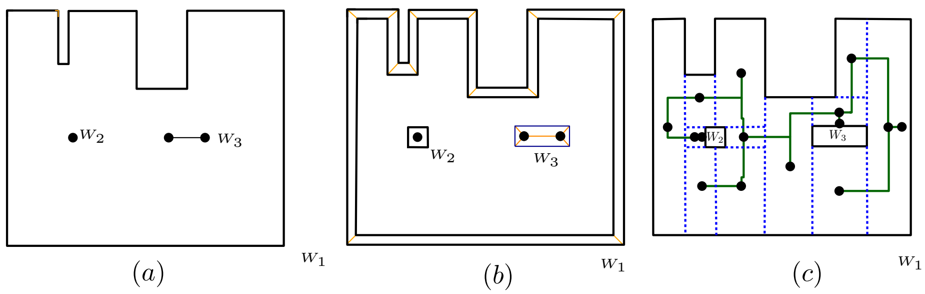

Figure 3: (a) A face with outer walk W_1 and, inner facial walks W_2 and W_3. (b) An approximation F' of F. (c) A face and a corresponding tree T.

1. P_i is ε-close to W_i, that is, every point of P_i is within distance ε of a point of W_i and vice-versa,
2. the inner facial walk W_i lies in the interior of P_i if $2 \le i \le k$, and
3. the outer facial walk W_1 lies in the exterior of P_1.

Lemma 2 *Let W be a facial walk in a face F of an orthogonal drawing of a graph G in the plane. An inner ε-approximating orthogonal polygon P_ε of W can be constructed in $O(|W|)$ time so that P_ε has at most $\max\{4, |W| + l\}$ vertices, where l is the number of degree-1 vertices in W.*

Proof: If W is an isolated vertex v, then approximate W with a square of side length $\sqrt{2}\varepsilon$ centered at v. Next, assume that W contains more than one vertex. We consider each vertex of degree 1 in W as a sequence of two degree-2 vertices that are connected by an infinitesimally short edge that forms a 270°-angle with the single edge incident to v inside F. Consider a corner e, v, e' of W, where e and e' are two consecutive edges and v is their shared vertex. Let α denote the angle formed by e and e' inside F. We choose v' as the point on the angular bisector of α in side F at distance ε from v. If $(v_i)_{i=1}^{k}$ is the sequence of vertices in W, then by joining $(v'_i)_{i=1}^{k}$, we get an orthogonal polygon that ε-approximates W. □

We now prove two auxiliary lemmas, which follow the structure of Lemmas 5 and 6 in [6]. Assume that G is a Hamiltonian graph with Hamiltonian cycle C. Lemma 3 provides a method to draw the edges of C, assuming that the vertex locations are fixed. Lemma 4 explains how to draw the remaining edges of G.

Lemma 3 *Let C be a cycle with fixed vertex locations, and suppose we are given a planar orthogonal drawing of a tree T with no bends, in which the vertices of C are leaves of T at their fixed locations. Then for every $\varepsilon > 0$ there is a planar Kandinsky drawing of C with at most $3|E(T)|$ bends per edge and ε-close to T.*

Proof: Let $p_1, \ldots, p_n$ be the vertices of the cycle C in order. To construct a planar poly-line drawing of C, Lemma 5 of [6] explains a method as follows. First of all, n ε-approximations $\theta_i (1 \le i \le n)$ of the given drawing of T are constructed, using Lemma 2. For $1 \le j \le n-1$, we denote by Q_j the unique path in T from P_j to P_{j+1} and we denote T_i by the collection of all paths Q_j for $j \le i$. If a vertex v of T_i is incident to an edge that is not in T_i, then the edge e is

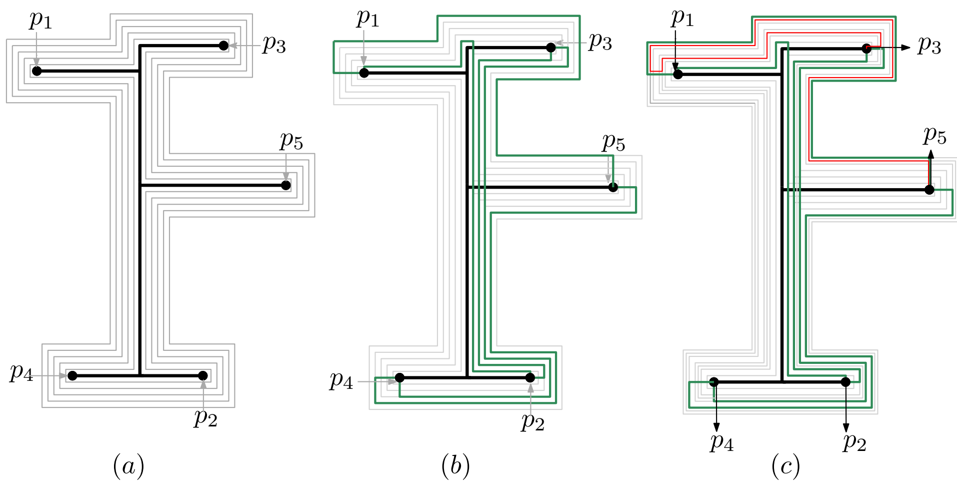

Figure 4: (a) An orthogonal drawing of a tree T together with approximations θ_i along T (b) A planar orthogonal drawing of the Hamiltonian cycle C with respect to T (c) The edge p_3p_5 is drawn using approximations of T.

replaced from θ_i to get θ'_i by approximating v with adjacent vertices v_1 and v_2 which lies on the edge bisector of e with adjacent edges (An example can be seen in Fig. 4 (b). The vertex p_3 is not a part of Q_1 and hence the edge adjacent to it is replaced with an edge bisector in Q'_1).

In order to draw the edges of C, we follow the same method explained above by constructing an $(i\varepsilon/n+1)$-approximation θ_i of the given orthogonal drawing of T using Lemma 2, for $1 \leq i \leq n$, and by routing the edges of C through the corresponding θ'_i's. An example given in Fig. 4 (a)-(c) illustrates how an edge is drawn along T using the approximations θ_i.

Here, note that each edge of C is replaced with a part of an approximation of θ_i and θ_i has at most $3|E(T)|$ edges. Hence each edge of C is replaced with an orthogonal arc that has at most $3|E(T)|$ bends. □

Lemma 4 *Let G be a Hamiltonian multigraph with a given planar embedding and fixed vertex locations. Suppose we are given an orthogonal drawing of a tree T with no bends, whose leaves include all the vertices of G at their fixed locations. Then for every $\varepsilon > 0$ there is a planar Kandinsky drawing of G so that*

1. *the drawing is ε-close to T,*
2. *the drawing realizes the given embedding,*
3. *the vertices of G are at their fixed locations,*
4. *every edge has at most $6|E(T)|$ bends, and*
5. *every edge comes close to any leaf of T at most twice, and only does so by terminating at or bending near the leaf.*

Proof: We closely follow Lemma 6 of [6] to construct a planar poly-line drawing of G, which works as follows. Using Lemma 5 of [6], a planar poly-line drawing of C with respect to the given drawing of T is constructed. Next, m approximations $\Delta_{i,k}$ of θ'_i are constructed using Lemma 2, for each $1 \leq i \leq n$ and $1 \leq k \leq m$, where $m = |E(G)|$. By using the fact that $\Delta_{i,k}$ crosses C twice; once when it traverses from p_i to θ'_{i+1} and from θ'_n to p_1; the $\Delta_{i,k}$ is cut into two pieces. Then the endpoints are joined with p_i and p_1 to get two curves $\Delta'_{i,k}$ and $\Delta''_{i,k}$. Note that one of the curves lies inside C, while the other lies outside. To route an edge p_ip_j, the path concatenating the straight-line polygons $\Delta'_{i,k}$ and $\Delta'_{j,k}$ is used. To construct a planar Kandinsky drawing of G, we continue in a similar manner. First, we route the edges of the Hamiltonian cycle C using Lemma 3 and then route the remaining edges by creating additional approximations of the curves θ'_i. Here, corresponding to an edge at most $6E(T)$ bends are introduced, since an edge p_ip_j is a concatenation of two approximations $\Delta'_{i,k}$ and $\Delta'_{j,k}$. An example shown in Fig. 5(a)-(c) depicts the procedure of drawing an edge through the approximations $\Delta_{i,k}$.

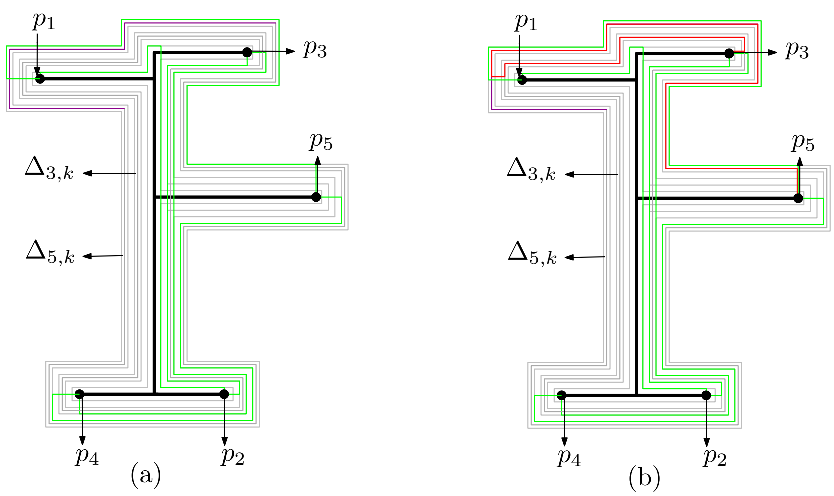

Figure 5: The edge p_3p_5 is drawn. (a) The polygons $\Delta_{3,k}$ and $\Delta_{5,k}$ are drawn in gray color. (b) The edge p_3p_5 is drawn in red color using parts of $\Delta_{3,k}$ and $\Delta_{5,k}$.

□

Now, in order to make the given graph Hamiltonian, we use the following result by Pach and Wenger [25].

Lemma 5 (Pach, Wenger [25]) *For a planar graph G, a Hamiltonian planar graph G' with $|E(G')| \leq 5|E(G)| - 10$ can be constructed from G by subdividing and adding edges in linear time. The construction is such that each edge of G is subdivided by at most two new vertices.*

Next, we assume that a planar embedding of the graph G together with a set of vertices $U \subseteq V(G)$ is given, where every element of U has a fixed location. The next lemma shows a method to route the edges of G by converting it into a Hamiltonian graph and then contracting the edges if at least one of its endpoint is not U. Finally, we undo the edge contractions to obtain a drawing of the original graph G.

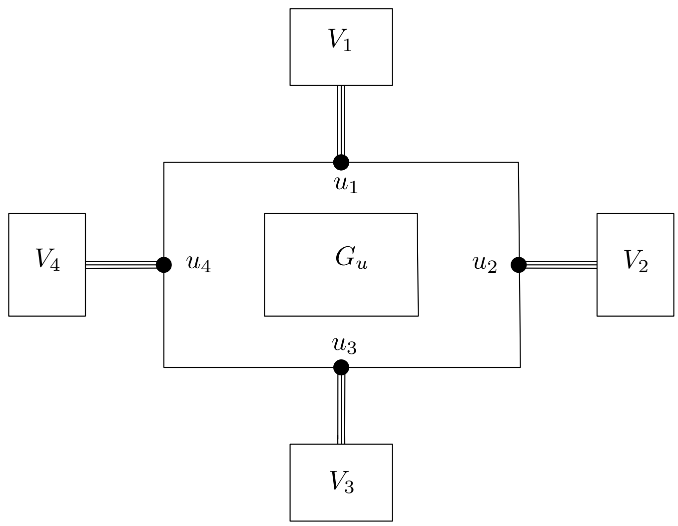

Figure 6: The graph G'_u.

Lemma 6 *Let G be a multigraph with a given planar embedding and fixed locations for a subset U of its vertices. Suppose we are given a planar orthogonal drawing of a tree T with no bends, whose leaves include all the vertices in U at their fixed locations. Then for every $\varepsilon > 0$ there is a planar Kandinsky drawing of G so that*

1. *the drawing is ε-close to T,*
2. *the drawing realizes the given embedding,*
3. *the vertices in U are at their fixed locations, and*
4. *each edge has at most $18|V(T)|$ bends and comes close to each vertex u in U at most 6 times, where coming close to u means intersecting an ε-neighborhood of u. Furthermore, any edge that comes close to u will either terminate at u or enter the ε-neighborhood of u, bend at a point in this ε-neighborhood, and then leave it.*

Proof: From a given graph G, construct a Hamiltonian graph G' with a Hamiltonian cycle C by subdividing each edge of G at most twice, and by adding some edges using Lemma 5. We traverse through C and whenever we encounter an edge e that has at least one endpoint not in U, then we contract e. Continue this process to get a multigraph G'' with a Hamiltonian cycle C' such that $V(G'') = U$.

Now, using Lemma 4, find a planar Kandinsky drawing Γ'' for G'' with respect to T. Fix a vertex $u \in V(G'')$ and let V_u be the vertices of G that have been contracted into u. Next, we have to draw the subgraph $G_u = G'[V_u]$ and route the edges that connect vertices from V_u to $V(G') \setminus V_u$. To draw G_u, construct a small disk around u in Γ''. Since Γ'' is a planar Kandinsky drawing, we can partition the neighborhood $N_{G''}(u)$ of u in G'' into four sets V_i, for $i = 1, \ldots, 4$, depending on the side of u to which its edge attaches in clockwise order. Now, let $G'_u = (V', E')$

with $V' = V_u \cup N_{G''}(u) \cup \{u_1, u_2, u_3, u_4\}$ and $E' = E(G_u) \cup \{u_1u_2, u_2u_3, u_3u_4, u_4u_1\} \cup \{u_ix : x \in V_i \text{ for } 1 \leq i \leq 4\} \cup \{yu_i : \text{ there exists an edge } yx \text{ in } G' \text{ with } x \in V_i \text{ and } y \in V_u\}$ (see Fig. 6).

Note that G'_u is a planar multigraph and hence it has a Kandinsky drawing Γ_u with at most two bends on each edge that can be computed in linear time [16]. Using Γ_u we can route the edges that connect V_u and V_i by ignoring the vertex u_i. Thus we get a Kandinsky drawing of G' with at most $6|E(T)| + 2$ bends per edge (using Lemma 4 and the two extra bends that are added while reconstructing G_u). Since each edge of G is subdivided at most twice to get G', each edge of G has $3(6|E(T)| + 2) = 18|E(T)| + 6 < 18|V(T)|$ bends. In addition, since each edge of G' comes close to a leaf of T at most twice, an edge of G comes close to a vertex of U at most six times. □

Now, we have all the required tools to prove Theorem 4.

4.1 Proof of Theorem 4:

Let F be a face of Γ_H. Let $W_i : 1 \leq i \leq a$ be facial walks inside F with isolated vertices and let $W_i, a+1 \leq i \leq a+b$, be facial walks inside F that involve more than one vertex. Construct an inner ε-approximation F' of F using Lemma 2 and let W'_i denote the orthogonal polygon that approximates W_i. Since $|W_i| \geq 2$ and by Lemma 2, $|W'_i| \leq \max\{4, |W_i| + l_i\} \leq |W_i| + 2 + l_i$, where l_i is the number of degree-1 vertices in W_i and $a+1 \leq i \leq a+b$. So we have $|F'| \leq \sum_{i=1}^{a} |W_i| + \sum_{i=a+1}^{a+b} (|W_i| + 2 + l_i) = a + 2b + \sum_{i=a+1}^{a+b} |W_i| + \sum_{i=a+1}^{a+b} l_i \leq a + 2b + \sum_{i=a+1}^{a+b} |W_i| + |\Gamma_H|$ (This is Step 1 as explained in the overview of the algorithm). Now, partition F' into rectangles using at most $n/2 + h - 1$ rectangles in time $O(n^{3/2} \log n)$ [15], where n is the number of vertices and h is the number of holes. So in our case, the number of rectangles will be
$\frac{1}{2}|F'| + a - 1 \leq \frac{1}{2}(a + 2b + \sum_{i=a+1}^{a+b} |W_i| + |\Gamma_H|) + a - 1 = \frac{1}{2} \sum_{i=a+1}^{a+b} |W_i| + \frac{1}{2}(3a + 2b) - 1 + \frac{1}{2}|\Gamma_H|$. Place a vertex at the center of each rectangle. Construct a graph K by joining the vertices of adjacent rectangles (we call two rectangles adjacent if they share one side) if the line segment joining the respective centers lies inside F'. Note that K is a connected graph. Let T be a spanning tree of K. Then T has $\frac{1}{2} \sum_{i=a+1}^{a+b} |W_i| + \frac{1}{2}(3a + 2b) - 1 + \frac{1}{2}|\Gamma_H|$ vertices.

Also, we can find a no-bend planar orthogonal drawing of T that adds at most two new vertices to each edge of T as follows. We consider two adjacent rectangles R_1 and R_2 with their respective centers c_1 and c_2. We start from c_1 and by spending zero bend, we reach line containing the shared segment between R_1 and R_2 and we add a vertex u there at the meeting point. We also draw a perpendicular line from c_2 to S and add a new vertex v at the intersection. From u, we can reach v and from v to c_2 using no bends.

Now, for each facial walk $W_i : 1 \leq i \leq a$ (that are isolated vertices), add the corresponding isolated vertex as a leaf to T. For each facial walk $W_i : a+1 \leq i \leq a+b$, add a new vertex near

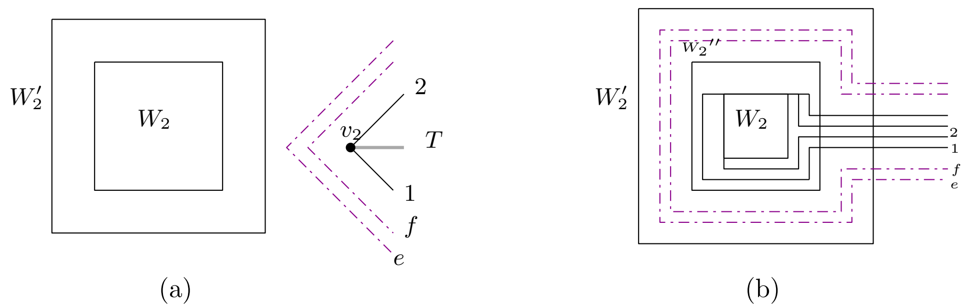

Figure 7: The vertex v_2 is incident to T. We route the edges e and f through the newly created buffer zone between W_2' and W_2'' and the edges incident to W_2 are routed through W_2'' and W_2.

to W_i as a leaf of T. This adds $a+b$ vertices to T and now, the number of vertices of T is

$$\begin{aligned}|V(T)| &= a+b+\frac{1}{2}\sum_{i=a+1}^{a+b}|W_i|+\frac{1}{2}(3a+2b)-1+\frac{1}{2}|\Gamma_H| \\ &\quad +2\left(\frac{1}{2}\sum_{i=a+1}^{a+b}|W_i|+\frac{1}{2}(3a+2b)+\frac{1}{2}|\Gamma_H|-1\right) \\ &= \frac{3}{2}\sum_{i=a+1}^{a+b}|W_i|+\frac{1}{2}(11a+8b)+\frac{3}{2}|\Gamma_H|-3\end{aligned}$$

(This concludes Step 2 in the overview of the algorithm).

Construct the multigraph G_F induced by the vertices lying inside or on the boundary of F, by contracting each facial walk of F to a single vertex. Draw G_F along T using Lemma 6. Note that the vertices corresponding to facial walks (inside F) are drawn at fixed locations. Here, each edge of G_F has at most $18|V(T)|$ bends.

Now, we reconstruct the edges between G_F and the non-isolated boundary components of F, following the same method as in Theorem 1, [6]. That is, by creating a buffer zone in between F' and F, the above mentioned edges are routed through the zone (see Figure 7). This adds at most $|W_i|+5$ bends for each edge.

Next, we have to add the edges of $G \setminus H$ that belong to F according to the given embedding $\mathcal{E}_G$ of G. By Lemma 4, an edge can come close at most six times to a vertex in U and thus an edge needs at most $6(|W_i|+5) = 6|W_i|+30$ bends to go around W_i. So altogether there are at most $6\sum_{i=a+1}^{a+b}|W_i|+30b$ bends along the whole edge to go around all the W_i's. Since we started with $18|V(T)|$ bends (Lemma 4) for each edge, this number increased to at most $6\sum_{i=a+1}^{a+b}|W_i|+30b+18|V(T)|$. Thus the total number of bends per edge can be calculated as follows.

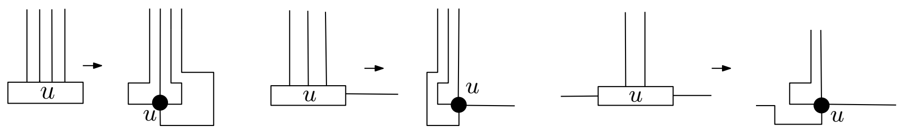

Figure 8: Re-routing the edges incident to a vertex u in the Kandinsky drawing Γ_G^K to obtain the orthogonal drawing Γ_G.

$$
\begin{aligned}
6\sum_{i=a+1}^{a+b}|W_i| + 30b + 18|V(T)| &\leq 6\sum_{i=a+1}^{a+b}|W_i| + 30b \\
&\quad + 18\left(\frac{3}{2}\sum_{i=a+1}^{a+b}|W_i| + \frac{1}{2}(11a+8b) + \frac{3}{2}|\Gamma_H| - 3\right) \\
&\leq 33\sum_{i=a+1}^{a+b}|W_i| + 99a + 102b + 27|\Gamma_H| - 54 \\
&\leq 33 \times 2|\Gamma_H| + 99 \times |\Gamma_H| + 27|\Gamma_H| \text{ since} \\
&\qquad \sum_{i=a+1}^{a+b}|W_i| \leq 2|\Gamma_H| \text{ and } a + 2b \leq |\Gamma_H| [6] \\
&\leq 192|\Gamma_H|
\end{aligned}
$$

Finally, by introducing a few more bends per edge, we can transform the Kandinsky drawing produced in Theorem 4 into an orthogonal drawing. This leads to the following theorem.

Theorem 5 *Let (G, H, Γ_H) be an instance of* RepExt(ortho)*. Suppose that G admits an orthogonal drawing Γ_G that extends Γ_H, and let $\mathcal{E}_G$ be the embedding of G in Γ_G. Then we can construct a planar orthogonal drawing of G in $O(n^2)$-time, where n is the number of vertices of G, that realizes $\mathcal{E}_G$, extends $\mathcal{H}$, and has at most $200|\Gamma_H|$ bends per edge.*

Proof: We first create a planar Kandinsky drawing Γ_G^K of the given graph G using Theorem 4. Let u be a vertex of G. Since G has an orthogonal drawing, we have that $\deg(u) \leq 4$. Note that, in Γ_G^K, some of the edges incident to u may be attached to the same port. Our goal is to change the port to which some of the edges are attached, in such a way that every edge is attached to a different port, while respecting the rotation at u in $\mathcal{E}_G$. Note that we only reroute G-edges, as H-edges have a fixed drawing and therefore no two H-edges can attach to the same port of a vertex. Since Γ_G is an orthogonal drawing extension, $\mathcal{E}_G$ satisfies the port constraints, and such a rerouting can be achieved as illustrated in Fig. 8. Note that this adds at most four bends at each endpoint of an edge, and thus adds at most eight bends per edge.

Applying this operation to all the vertices of G yields a planar orthogonal drawing Γ_G of G that realizes $\mathcal{E}_G$, extends $\mathcal{H}$, and has at most $200|\Gamma_H|$ bends per edge (at most twice four additional bends on each edge). $\square$

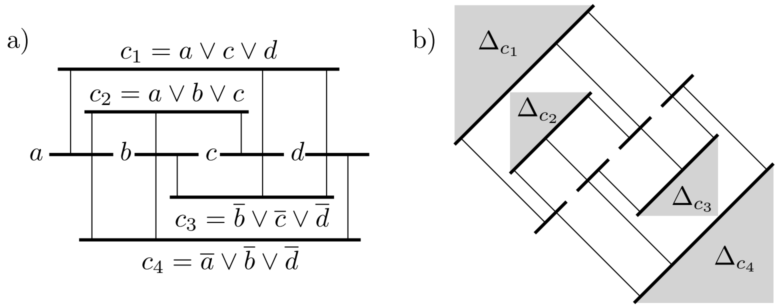

Figure 9: A representation of an instance of monotone planar 3-SAT with four variables a, b, c, d and four clauses c_1, c_2, c_3, c_4 (a). Image of the vertically-stretched version of (a) under the mapping Φ (b).

5 Bend-Optimal Extension

In this section we study the problem of computing an orthogonal drawing extension of an instance $I = (G, H, \Gamma_H)$ of RepExt(ortho) with the minimum number of bends. Observe that, if H is empty, this is equivalent to computing a bend-minimal drawing of G, which is NP-complete if the embedding of G is not fixed [17]. We thus assume that G comes with a fixed planar embedding $\mathcal{E}_G$ that satisfies the port constraints of Γ_H, and we study the complexity of computing a bend-optimal drawing Γ_G of G with embedding $\mathcal{E}_G$ that extends Γ_H.

Here, we specifically focus on the restricted case where $V(H) = V(G)$ and $E(H) = \emptyset$, which we call Orthogonal Point Set Embedding with Fixed Mapping. We show that, even in this case, it is NP-hard to minimize the number of bends on the edges. On the positive side, we show that in this case the existence of a drawing that uses one bend per edge can be tested in polynomial time.

Theorem 6 Orthogonal Point Set Embedding with Fixed Mapping *is NP-complete.*

Proof: The problem is contained in NP, as it can be solved non-deterministically in polynomial time as follows. First, we non-deterministically guess the positions and the directions of the bends and use this information to subdivide the edges. Second, we non-deterministically guess for a drawing the horizontal and vertical order of the vertices (including the information which of them are aligned). This combinatorially fixes the orthogonal drawing, and we can then test in polynomial time whether there exists a drawing with the given vertex order.

To show NP-hardness, we give a reduction from the NP-complete problem *monotone planar 3-SAT* [9]. In this variant of 3-SAT, the variable–clause graph is planar and has a layout where the variables are represented by horizontal segments on the x-axis, the clauses by horizontal segments above and below the x-axis, and each variable is connected to each clause containing it by a vertical segment. Moreover, the clauses above the x-axis contain only positive literals and the clauses below contain only negative literals; see Fig. 9a.

A *box* is an axis-aligned rectangle whose bottom-left and top-right corners contain two H-vertices, connected by a G-edge. We consider non-degenerate boxes, and thus this G-edge requires at least one bend; when this edge is drawn with one bend, there is a choice whether it contains the top-left or the bottom-right corner of the box. In these cases we say that the box is *drawn top* and *drawn bottom*, respectively. We now describe our *variable*, *pipe*, and *clause gadgets*.

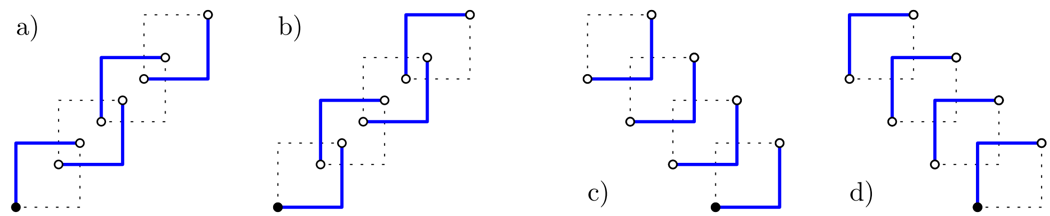

Figure 10: Variable gadget with $h = 4$ boxes (a,b). In (a) the odd boxes are drawn top and the even boxes are drawn bottom, (b) shows the opposite. Pipe gadget (c,d). In (c) all boxes are drawn bottom, in (d) they are all drawn top. In all cases the base point is marked.

A *variable gadget* consists of $h > 0$ boxes $R_1, \ldots, R_h$ that are 3×3-squares, where the bottom-left corner of R_i lies at $b+(2(i-1), 2(i-1))$, for an arbitrary base point $o = (o_x, o_y)$; see Fig. 10a-b. The crucial property is that in a one-bend drawing of the gadget, R_i is drawn bottom if and only if R_{i+1} is drawn top for $i = 1, \ldots, h-1$. Thus, in such a drawing, either all the *odd boxes* (those with odd indices) are drawn top and all the *even boxes* (those with even indices) are drawn bottom, or vice versa. This will be used to encode the truth value of a variable.

A *(positive) pipe gadget* works similarly; see Fig. 10c-d. For a base point o, it consists of $h > 0$ boxes $R_1, \ldots, R_h$ that are 3×3-squares such that the bottom-left corner of R_i lies at $o+(-2(i-1), 2(i-1))$; see Fig. 10c-d. The decisive property is that in a one-bend drawing of the gadget, if R_1 is drawn bottom, then also all other boxes of the gadget can be drawn bottom (see Fig. 10c), whereas if R_1 is drawn top, then so must be all other boxes of the gadget (see Fig. 10d). Negative pipe gadgets are symmetric with respect to the line $y = x$ and behave symmetrically.

The last gadget we describe is the *(positive) clause gadget*; negative clause gadgets are symmetric with respect to the line $y = x$ and behave symmetrically. The positive clause gadget has three *input boxes* R_1, R_2, R_3, whose corners lie on a single line with slope 1; we assume that R_1 lies left of R_2, which in turn lies left of R_3. To simplify the description, we assume that the left lower corners of these rectangles lie at $(x, x), (y, y)$, and (z, z), respectively. Refer to Fig. 11a.

We create three *literal* boxes L_1, L_2, L_3 that are 3×3-squares. The lower left corner of L_1 is $(x-3, y+2)$, the lower left corner of L_2 if $(y-2, y+2)$, and the lower left corner of L_3 is $(y, z+3)$. Note that the interiors of L_2 and R_2 intersect in a unit square, and therefore, if R_2 is drawn top, then L_2 must be drawn top, too, thereby encoding that the corresponding literal of the clause is set to false. To obtain the same behavior for the other input and literal rectangles, we add two *transmission boxes* T_1 and T_2. The lower left corner of T_1 is $(x-1, x+2)$ and its upper right corner is $(x+1, y+3)$. The bottom-left and top-right corner of T_2 are $(y+2, z+2)$ and $(z+1, z+4)$, respectively. This guarantees that, also for $i = 1, 3$, if R_i is drawn top, then T_i and L_i are drawn top. We finally have a *blocker* box B, with corners at $(x-1, z+4)$ and $(x+1, z+7)$; and a *clause box*, whose corners are in the centers of L_1 and L_3, respectively.

Note that the G-edge connecting the two corners of the clause box, which we call the *clause edge*, requires at least two bends, as any one-bend drawing cuts horizontally through either the blocker B or the literal square L_2; see Fig. 11a. The following claim shows that the possibility of drawing it with exactly two bends depends on the drawings of the literal boxes of the clause gadget, and thus on the truth values of the literals; see Fig. 11b-c.

Claim 1 *If the other edges are drawn with one bend, then the clause edge can be drawn with two bends if and only if not all literal boxes are drawn top.*

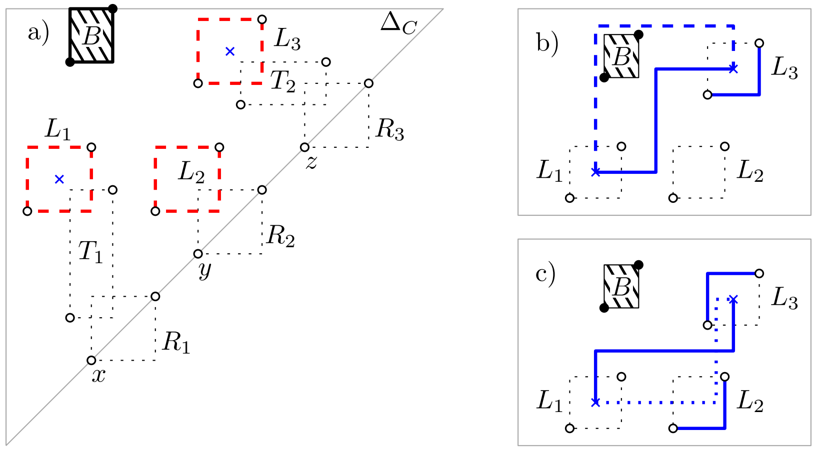

Figure 11: Clause gadget with input rectangles R_1, R_2, R_3. The bottom-left and top-right corner of the clause box are drawn as crosses (a). The image of the triangle Δ_C under the mapping $(x, y) \mapsto (x - y, x + y)$ is drawn gray. The possibilities of routing the clause edge with two bends, if L_3 is drawn bottom (b) and if L_3 is drawn top and L_2 is drawn bottom (c).

Proof: Suppose, for a contradiction, that the clause edge is drawn with two bends, but all three literal boxes are drawn top. Then, starting from the center of L_1, the clause edge must first intersect the bottom or the right side of L_1. If it intersects the bottom side, then it further consists of a horizontal segment and a vertical segment that then ends at the center of L_3. But then either the horizontal segment cuts horizontally through T_1, or the vertical segment cuts vertically through R_2. Both cases contradict the assumption that the drawing is without crossings. Hence we can assume that the clause edge intersects the right side of L_1. Since it cannot intersect the left side of L_2, there must be a bend on the segment between the centers of L_1 and L_2 that lies outside of these two boxes. To reach the other endpoint of the clause edge with just one more bend, the next segment has to be vertical and must end at the y coordinate of the endpoint of the clause edge that lies inside the literal box L_3. But then the last horizontal segment of the clause edge crosses the literal edge of the box L_3, which is drawn top by assumption.

On the other hand, we show that if at least one of L_1, L_2, L_3 is not drawn top, then we can draw the clause edge with two bends. Assume that L_3 is drawn bottom. Depending on whether the top-left or bottom-right corner of L_1 is used, we can draw the clause edge as indicated by the solid or the dashed curve in Fig. 11b. Note that this is independent of whether L_2 is drawn top or bottom. Now assume that L_3 uses its top-left corner. If L_1 is drawn bottom, we can draw the clause edge as indicated by the solid curve in Fig. 11c. Finally, if both L_1 and L_3 use their top-left corner, but L_2 does not, we can route the clause edge as indicated by the dotted curve in Fig. 11c. □

We are now ready to put the construction together. Consider the layout of the variable–clause graph, where each variable x is represented by a horizontal segment s_x on the x-axis, and each clause $C = (c_1, c_2, c_3)$ with only positive (only negative) literals by a horizontal segment s_C above (below) the x-axis. Further, the occurrence of a variable x in a clause C is represented by a vertical visibility segment $s_{x,C}$ that starts at an inner point of s_x and ends at an inner point of s_C; see Fig. 9a. We call these points *attachment points*. By suitably stretching the drawing horizontally, we may assume that all segments start and end at points with integer coordinates divisible by 8. We also stretch the whole construction vertically by a factor of n, which guarantees that for each clause segment s_C the right-angled triangle Δ_C, whose long side is s_C and that lies above s_C (below s_C if C consists of negative literals) does not intersect any other segments in its interior. Note that the initial drawing fits on a grid of polynomial size [23], and the transformations only increase the area polynomially. For the construction it is useful to consider this representation rotated by 45° in counterclockwise direction and scaled by a factor of $\sqrt{2}$ back to the grid. This is achieved by the affine mapping $\Phi\colon (x,y) \mapsto (x-y, x+y)$; see Fig. 9b.

For each variable segment s_x with left endpoint $(a,0)$ and right endpoint $(b,0)$ we create a variable gadget with $h = (b-a)/2$ boxes and base point (a,a). For each clause segment s_C above the x-axis with attachment points $(a_1,b),(a_2,b),(a_3,b)$, we create a positive clause gadget with input boxes at $(a_i - b, a_i + b)$. For each vertical segment $s_{x,C}$ above the x-axis with attachment points $(a,0)$ and (a,b), we create a positive pipe gadget of $h = (b/2) - 2$ boxes at base point $(a-2, a-2)$. Note that, together with the box of the variable gadget of x at (a,a) and the input box of C at $(a-b, a+b)$, the newly placed boxes form a pipe gadget that consists of $h+2$ boxes. Since distinct vertical segments on the same side of the x-axis have horizontal distance at least 8, the boxes of distinct pipes do not intersect, and the placement is such that only the first and last box of each pipe gadget intersect boxes that belong to the corresponding variable or clause gadget. Finally note that for each clause C, except for the input boxes, the clause gadget lies inside the image of the triangle Δ_C under the mapping Φ, since the attachment points are interior points of s_C, and the x-coordinates of its endpoints are divisible by 8. Hence, the only interaction of the clause gadget with the remainder of the construction is via the input variables. The proof of the following claim is based on showing that we can draw each box with exactly one bend and each clause edge with exactly two bends, if and only if the original instance of monotone planar 3-SAT is satisfiable.

Claim 2 *Let φ be an instance of monotone planar 3-SAT, with γ clauses. Also, let β be the number of boxes in the instance (G, H, Γ_H) of* RepExt(ortho) *constructed as described above. Then, the formula φ is satisfiable if and only if the instance (G, H, Γ_H) admits an extension with at most $k = \beta + \gamma$ bends.*

Proof: Assume we are given a satisfying assignment of φ. For each variable, we draw the odd boxes bottom and the even boxes top if the variable is assigned value true, and the other way around if it is false. For each clause C, let x be a variable that satisfies it. We discuss the case that C contains only positive literals, the case that it only contains negative literals is symmetric. We draw C in such a way that the input box of x is drawn bottom and all other input boxes are drawn top. We draw the boxes of the pipe gadget that connects x to C bottom, and the remaining pipe gadgets that connect to other variables to C top. Note that the latter cannot cause crossings, and the former do not cause a crossing, since it only intersects with an odd box of the variable gadget of x, which is drawn bottom since x is true. By Claim 1, the clause edge of C can be drawn with two bends. Altogether, we obtain a crossing-free orthogonal drawing Γ_G of the instance that has $\beta + \gamma$ bends (one bend per box, and one additional bend per clause).

Conversely, assume that there exists a drawing Γ_G with $\beta + \gamma$ bends. Recall that each box requires at least one bend, and each clause edge requires at least two bends. It follows that each clause edge is drawn with two bends, and that each edge of the remaining gadgets is drawn with one bend. We now assign a variable x the value true if and only if its odd boxes are drawn bottom. Let C be a clause with only positive literals; the case with only negative literals is symmetric. Since the clause edge of C is drawn with two bends, it follows that at least one of the input boxes

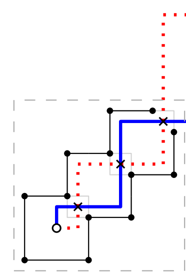

Figure 12: Gadget for forcing an edge to use $k = 4$ bends. All vertices and the thin solid black lines are H-vertices. Up to minor geometric adjustments, the thick blue and dotted red lines show the only two ways to draw the G-edge between the two H-vertices u and v with k bends. Scaling the lower left part to make it sufficiently small results in a construction that behaves like a box.

is drawn bottom. Then all boxes of the corresponding pipe are also drawn bottom, and therefore an odd box of the corresponding variable is also drawn bottom. Hence the variable is true and C is satisfied. □

Since the construction has polynomially many vertices and edges on a polynomial size grid, it can be executed in polynomial time. Moreover, by construction, $V(H) = V(G)$, $E(H) = \emptyset$, and $E(G)$ is a matching. The statement of the theorem follows. □

We observe that the graph in the reduction of Theorem 6 is a matching, and therefore also requiring a fixed embedding would not make the problem easier. We further observe that the non-clause G-edges require one bend, and the clause edge require two bends, each. Hence, by subdividing each non-clause G-edge with a G-vertex, and each clause edge with two G-vertices, we get the following corollary.

Corollary 1 *It is NP-complete to decide whether a partial orthogonal drawing (G, H, Γ_H) admits an extension without bends.*

Similarly, we can ask whether an instance (G, H, Γ_H) admits an extension with at most k bends per edge for a fixed number k. The construction depicted in Fig. 12 shows how to force an edge to use k bends for any fixed number k. By making the part that enforces the first $k - 1$ bends sufficiently small, we essentially obtain the behavior of the box gadget from the proof of Theorem 6.

Corollary 2 *For any fixed $k \geq 2$, it is NP-complete to decide whether an instance (G, H, Γ_H) of* RepExt(ortho) *admits an extension that uses at most k bends per edge, even if $V(G) = V(H)$.*

On the positive side, if all vertices are predrawn, the existence of an extension with at most k bends per edge can be tested efficiently for $k = 0$ and $k = 1$.

Theorem 7 *Let (G, H, Γ_H) be an instance of* RepExt(ortho) *with $V(G) = V(H)$ and let $k \in \{0, 1\}$. It can be tested in polynomial time whether (G, H, Γ_H) admits an extension with at most k bends per edge.*

Proof: For $k = 0$ we simply draw each G-edge as the straight-line segment between its endpoints, and check whether this is a crossing-free orthogonal drawing.

For $k = 1$ we proceed as follows. While there exists a G-edge $e = uv$ whose endpoints have the same x- or the same y-coordinates, we do the following. If e must be drawn as a straight-line (if u and v have the same x- or the same y-coordinates), the instance (G, H, Γ_H) is equivalent to the instance (G, H', Γ'_H), where H' is obtained from H by adding e, and Γ'_H is obtained from inserting e as a straight-line segment. By applying this reduction rule, we eventually arrive at an instance (G'', H'', Γ''_H) such that the endpoints of each G-edge have distinct x- and distinct y-coordinates. Now for each such edge, there are precisely two ways to draw them with one bend. It is then straightforward to encode the existence of choices that lead to a planar drawing into a 2-SAT formula. □

6 Conclusions

In this paper we studied the problem of extending a partial orthogonal drawing. We gave a linear-time algorithm to test the existence of such an extension, and we proved that if one exists, then there is also one whose edge complexity is linear in the size of the given drawing. On the other hand, we showed that, if we also restrict the total number of bends or the number of bends per edge, then deciding the existence of an extension is NP-complete.

Concerning future work we feel that the most important questions are the following:

1. The complexity of $200|\Gamma_H|$ bends per edge resulting from the transition to orthogonal drawings is significantly worse than the one of $72|V(H)|$ bends per edge in the case of arbitrary polygonal drawings [6]. Can this number be significantly reduced to, say, less than $100|\Gamma_H|$?

2. As mentioned in the introduction, Tamassia [27] already observed that an orthogonal representation of H can be efficiently extended to an orthogonal representation of G. However, drawing such an extension may require to modify the drawing Γ_H of the given subgraph. Is it possible to efficiently test whether a given orthogonal representation can be drawn such that it extends a given drawing Γ_H?

Acknowledgements. The authors wish to thank the anonymous referees whose suggestions improved the presentation of this paper.

References

[1] P. Angelini, S. Chaplick, S. Cornelsen, G. Da Lozzo, G. Di Battista, P. Eades, P. Kindermann, J. Kratochvíl, F. Lipp, and I. Rutter. Simultaneous orthogonal planarity. In Y. Hu and M. Nöllenburg, editors, *Proceedings of the 24th International Symposium on Graph Drawing and Network Visualization (GD'16)*, volume 9801 of *Lecture Notes in Computer Science*, pages 532–545. Springer, 2016. doi:10.1007/978-3-319-50106-2_41.

[2] P. Angelini, G. Di Battista, F. Frati, V. Jelínek, J. Kratochvíl, M. Patrignani, and I. Rutter. Testing planarity of partially embedded graphs. *ACM Trans. Algorithms*, 11(4):32:1–32:42, 2015. doi:10.1145/2629341.

[3] P. Angelini, I. Rutter, and S. T. P. Extending partial orthogonal drawings. In D. Auber and P. Valtr, editors, *Graph Drawing and Network Visualization - 28th International Symposium, GD 2020, Vancouver, BC, Canada, September 16-18, 2020, Revised Selected Papers*, volume

12590 of *Lecture Notes in Computer Science*, pages 265–278. Springer, 2020. doi:10.1007/978-3-030-68766-3_21.

[4] T. Bläsius, S. Lehmann, and I. Rutter. Orthogonal graph drawing with inflexible edges. *Comput. Geom.*, 55:26–40, 2016. doi:10.1016/j.comgeo.2016.03.001.

[5] T. Bläsius, I. Rutter, and D. Wagner. Optimal orthogonal graph drawing with convex bend costs. *ACM Trans. Algorithms*, 12(3):33:1–33:32, 2016. doi:10.1145/2838736.

[6] T. M. Chan, F. Frati, C. Gutwenger, A. Lubiw, P. Mutzel, and M. Schaefer. Drawing partially embedded and simultaneously planar graphs. *J. Graph Algorithms Appl.*, 19(2):681–706, 2015. doi:10.7155/jgaa.00375.

[7] S. Chaplick, P. Dorbec, J. Kratochvíl, M. Montassier, and J. Stacho. Contact representations of planar graphs: Extending a partial representation is hard. In D. Kratsch and I. Todinca, editors, *Graph-Theoretic Concepts in Computer Science*, pages 139–151, Cham, 2014. Springer International Publishing. doi:10.1007/978-3-319-12340-0_12.

[8] S. Chaplick, R. Fulek, and P. Klavík. Extending partial representations of circle graphs. *Journal of Graph Theory*, 91(4):365–394, 2019. doi:10.1002/jgt.22436.

[9] M. de Berg and A. Khosravi. Optimal binary space partitions for segments in the place. *International Journal of Computational Geometry & Applications*, 22(03):187–205, 2012. doi:10.1142/S0218195912500045.

[10] G. Di Battista, G. Liotta, and F. Vargiu. Spirality and optimal orthogonal drawings. *SIAM J. Comput.*, 27(6):1764–1811, 1998. doi:10.1137/S0097539794262847.

[11] W. Didimo and G. Liotta. Computing orthogonal drawings in a variable embedding setting. In K.-Y. Chwa and O. H. Ibarra, editors, *Algorithms and Computation, 9th International Symposium, ISAAC '98, Taejon, Korea, December 14-16, 1998, Proceedings*, volume 1533 of *Lecture Notes in Computer Science*, pages 79–88. Springer, 1998. doi:10.1007/3-540-49381-6_10.

[12] W. Didimo, G. Liotta, G. Ortali, and M. Patrignani. Optimal orthogonal drawings of planar 3-graphs in linear time. In S. Chawla, editor, *Proceedings of the 30th ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 806–825. SIAM, 2020. doi:10.1137/1.9781611975994.49.

[13] C. A. Duncan and M. T. Goodrich. Planar orthogonal and polyline drawing algorithms. In R. Tamassia, editor, *Handbook on Graph Drawing and Visualization*, pages 223–246. Chapman and Hall/CRC, 2013.

[14] E. Eiben, R. Ganian, T. Hamm, F. Klute, and M. Nöllenburg. Extending Partial 1-Planar Drawings. In A. Czumaj, A. Dawar, and E. Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 43:1–43:19, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2020.43.

[15] D. Eppstein. Graph-theoretic solutions to computational geometry problems. In C. Paul and M. Habib, editors, *Graph-Theoretic Concepts in Computer Science*, pages 1–16, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. doi:10.1007/978-3-642-11409-0_1.

[16] U. Fößmeier and M. Kaufmann. Drawing high degree graphs with low bend numbers. In F. J. Brandenburg, editor, *Graph Drawing*, pages 254–266, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg. doi:10.1007/BFb0021809.

[17] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing*, pages 286–297, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg. doi:10.1007/3-540-58950-3_384.

[18] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001. doi:10.1137/S0097539794277123.

[19] V. Jelínek, J. Kratochvíl, and I. Rutter. A Kuratowski-type theorem for planarity of partially embedded graphs. *Comput. Geom.*, 46(4):466–492, 2013. doi:10.1016/j.comgeo.2012.07.005.

[20] P. Klavík, J. Kratochvíl, Y. Otachi, I. Rutter, T. Saitoh, M. Saumell, and T. Vyskocil. Extending partial representations of proper and unit interval graphs. *Algorithmica*, 77(4):1071–1104, 2017. doi:10.1007/s00453-016-0133-z.

[21] P. Klavík, J. Kratochvíl, Y. Otachi, and T. Saitoh. Extending partial representations of subclasses of chordal graphs. *Theor. Comput. Sci.*, 576:85–101, 2015. doi:10.1016/j.tcs.2015.02.007.

[22] P. Klavík, J. Kratochvíl, Y. Otachi, T. Saitoh, and T. Vyskocil. Extending partial representations of interval graphs. *Algorithmica*, 78(3):945–967, 2017. doi:10.1007/s00453-016-0186-z.

[23] D. E. Knuth and A. Raghunathan. The problem of compatible representatives. *SIAM J. Discret. Math.*, 5(3):422–427, 1992. doi:10.1137/0405033.

[24] T. Krawczyk and B. Walczak. Extending partial representations of trapezoid graphs. In H. L. Bodlaender and G. J. Woeginger, editors, *Proceedings of the 43rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG'17)*, volume 10520 of *Lecture Notes in Computer Science*, pages 358–371. Springer, 2017. doi:10.1007/978-3-319-68705-6_27.

[25] J. Pach and R. Wenger. Embedding planar graphs at fixed vertex locations. *Graphs and Combinatorics*, 17:717–728, 2001. doi:10.1007/PL00007258.

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

[27] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal on Computing*, 16(3):421–444, 1987. doi:10.1137/0216030.