# Extending Partial 1-Planar Drawings

**Eduard Eiben**
Department of Computer Science, Royal Holloway, University of London, Egham, United Kingdom
eduard.eiben@rhul.ac.uk

**Robert Ganian**
Algorithms and Complexity Group, TU Wien, Austria
rganian@gmail.com

**Thekla Hamm**
Algorithms and Complexity Group, TU Wien, Austria
thamm@ac.tuwien.ac.at

**Fabian Klute**
Algorithms and Complexity Group, TU Wien, Austria
fklute@ac.tuwien.ac.at

**Martin Nöllenburg**
Algorithms and Complexity Group, TU Wien, Austria
noellenburg@ac.tuwien.ac.at

## Abstract

Algorithmic extension problems of partial graph representations such as planar graph drawings or geometric intersection representations are of growing interest in topological graph theory and graph drawing. In such an extension problem, we are given a tuple $(G, H, \mathcal{H})$ consisting of a graph $G$, a connected subgraph $H$ of $G$ and a drawing $\mathcal{H}$ of $H$, and the task is to extend $\mathcal{H}$ into a drawing of $G$ while maintaining some desired property of the drawing, such as planarity.

In this paper we study the problem of extending partial 1-planar drawings, which are drawings in the plane that allow each edge to have at most one crossing. In addition we consider the subclass of IC-planar drawings, which are 1-planar drawings with independent crossings. Recognizing 1-planar graphs as well as IC-planar graphs is NP-complete and the NP-completeness easily carries over to the extension problem. Therefore, our focus lies on establishing the tractability of such extension problems in a weaker sense than polynomial-time tractability. Here, we show that both problems are fixed-parameter tractable when parameterized by the number of edges missing from $H$, i.e., the edge deletion distance between $H$ and $G$. The second part of the paper then turns to a more powerful parameterization which is based on measuring the vertex+edge deletion distance between the partial and complete drawing, i.e., the minimum number of vertices and edges that need to be deleted to obtain $H$ from $G$.

**2012 ACM Subject Classification** Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Computational geometry

**Keywords and phrases** Extension problems, 1-planarity, parameterized algorithms

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2020.43

**Category** Track A: Algorithms, Complexity and Games

**Related Version** A full version of this article is available at `https://arxiv.org/abs/2004.12222`.

**Funding** *Robert Ganian*: Robert Ganian acknowledges support by the Austrian Science Fund (FWF, project P31336).
*Thekla Hamm*: Thekla Hamm acknowledges support by the Austrian Science Fund (FWF, projects P31336 and W1255-N23).


47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 43; pp. 43:1–43:19
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 1 Introduction

In the last decade, algorithmic extension problems of partial *planar* graph drawings have received a lot of attention in the fields of graph algorithms and graph theory as well as in graph drawing and computational geometry. In this problem setting, the input consists of a planar graph $G$, a connected subgraph $H$ of $G$, and a planar drawing $\mathcal{H}$ of $H$; the question is then whether $\mathcal{H}$ can be extended to a planar drawing of $G$. This extension problem is motivated from applications in network visualization, where important patterns (subgraphs) are required to have a special layout, or where new vertices and edges in a dynamic graph must be inserted into an existing (partial) connected drawing, which must remain stable to preserve its mental map [31]. A major result on the extension of partial planar drawings is the linear-time algorithm of Angelini et al. [2] which can answer the above question as well as provide the desired planar drawing of $G$ (if it exists). The result of Angelini et al. contrasts other extension problems in the context of computational geometry and graph drawing, which are typically NP-complete [7, 9–13, 16, 23–26, 30, 32], even in settings where recognition is polynomial-time solvable. On the other end of the planarity spectrum, Arroyo et al. [3, 4] studied drawing extension problems, where the number of crossings per edge is not restricted, yet the drawing must be *simple*, i.e., any pair of edges can intersect in at most one point. They showed that the simple drawing extension problem is NP-complete [3], even if just one edge is to be added [4].

In this paper, we study the algorithmic extension problem of partial drawings of 1-planar graphs, one of the most natural and most studied generalizations of planarity [17, 27, 33], and of partial drawings of IC-planar graphs, a natural restriction of 1-planarity [1, 5, 29, 35]. A 1-planar graph is a graph that admits a drawing in the plane with at most one crossing per edge; for IC-planarity, we additionally require that no two crossed edges are adjacent. Unlike planarity testing, recognizing 1-planar graphs is NP-complete [20, 28], even if the graph is a planar graph plus a single edge [8]. Recognition of IC-planar graphs also remains NP-complete [6].

**Contributions.** Given a graph $G$, a connected subgraph $H$, and a 1-planar drawing $\mathcal{H}$ of $H$, the 1-Planar Extension problem asks whether $\mathcal{H}$ can be extended by inserting the remaining vertices $V_{\text{add}} = V(G) \setminus V(H)$ and edges $E_{\text{add}} = E(G) \setminus E(H)$ of $G$ into $\mathcal{H}$ while maintaining the property of being 1-planar. The IC-Planar Drawing Extension problem is then defined analogously, but for IC-planarity.

The NP-completeness of these extension problems is a simple consequence of the NP-completeness of the recognition problem [6, 20, 28] (see also Section 3). With this in mind, the aim of this paper is to establish the tractability of the problems when $\mathcal{H}$ is almost a complete 1-planar drawing of $G$. To capture this setting, we turn to the notion of *fixed-parameter tractability* [15, 19] and consider two natural parameters which capture how complete $\mathcal{H}$ is: the *edge deletion distance* between $H$ and $G$ (denoted by $k$), and the *vertex+edge deletion distance* between $H$ and $G$ (denoted by $\kappa$). More precisely, $k$ is equal to $|E(G) \setminus E(H)|$ and $\kappa$ is equal to $|V(G) \setminus V(H)| + |E(G[V(H)]) \setminus E(H)|$. We refer to Section 3 for formal definitions and a discussion of the parameters.

After introducing necessary notation in Section 2 and introducing the problem formally in Section 3, we consider the edge deletion distance $k$ in **Section 4**. Our first result is:

▶ **Theorem 1.** *1-Planar Drawing Extension is* FPT *when parameterized by $k$.*

The proof of Theorem 1 involves the use of several ingredients:

1. Introducing and developing a notion of *patterns*, which are combinatorial objects that capture critical information about the potential interaction of newly added edges with $\mathcal{H}$;
2. a pruning procedure that reduces our instance to an equivalent sub-instance where $H$ has treewidth bounded in $k$;
3. an embedding graph, which carries information about the drawing $\mathcal{H}$; and finally
4. completing the proof by constructing a formula $\Phi$ in Monadic Second Order Logic to check whether a pattern can "fit" in the embedding graph, using Courcelle's Theorem [14].

Next, we turn towards the question of whether one can obtain an *efficient* fixed-parameter algorithm for the extension problem. In particular, due to the use of Courcelle's Theorem [14] to model-check $\Phi$, the algorithm obtained in the proof of Theorem 1 will have a prohibitive dependency on the parameter $k$. In this direction, we note that it is not immediately obvious how one can design an efficient and "formally clean" purely combinatorial algorithm for the pattern-fitting task (i.e., the task we relegate to model checking $\Phi$ in the embedding graph). At the very least, using a direct translation of the model-checking procedure would come at a significant cost in terms of presentation clarity.

That being said, one can observe that the main reason for the use of patterns is that it is not at all obvious where (i.e., in which cell of the drawing) one should place the vertices used to extend $\mathcal{H}$. Indeed, our second result for parameter $k$ assumes that $V_{\text{add}} = \emptyset$ and avoids using Courcelle's Theorem.

▶ **Theorem 2.** *1-Planar Drawing Extension parameterized by $k$ can be solved in time $\mathcal{O}(k^{2k} \cdot n^{\mathcal{O}(1)})$ if $V(G) = V(H)$.*

This algorithm uses entirely different techniques – notably, it prunes the search space for inserting each individual edge via a combination of geometric and combinatorial arguments, and then applies exhaustive branching. We note that the techniques used to prove Theorem 1 and 2 can be directly translated to also obtain analogous results for the IC-planarity setting.

In **Section 5**, we turn our attention to the vertex+edge deletion distance $\kappa$ as a parameter, which represents a more relaxed way of measuring how complete $\mathcal{H}$ is than $k$ – indeed, while $\kappa \leq k$, it is easy to construct instances where $\kappa = 1$ but $k$ can be arbitrarily large. For our third result, we start with IC-planar drawings.

▶ **Theorem 3.** *IC-Planar Drawing Extension is* FPT *parameterized by $\kappa$.*

The proof of Theorem 3 requires a significant extension of the toolkit developed for Theorem 1. The main additional complication lies in the fact that the number of edges that are missing from $H$ is no longer bounded by the parameter. To deal with this, we show that the added vertices can only connect to the boundary of a cell in a bounded number of "ways" (formalized via a notion we call *regions*), and we use this fact to develop a more general notion of patterns and embedding graphs than those used for Theorem 1.

Finally, in **Section 6**, we present a first step towards the tractability of 1-Planar Drawing Extension parameterized by $\kappa$. We note that the techniques developed for the other parameterizations and problem variants cannot be applied to solve this case – the main difference compared to the setting of Theorem 3 is that the "missing" vertices can be incident to many edges with crossings, which prevents the use of our bounded-size patterns to capture the behavior of new edges. As our final contribution, we investigate the special case of $\kappa = 2$, i.e., when adding two new vertices.

▶ **Theorem 4.** *1-Planar Drawing Extension is polynomial-time tractable if $\kappa \leq 2$.*

We note that even this, seemingly very restricted, subcase of 1-Planar Drawing Extension was non-trivial and required the combination of several algorithmic techniques (this contrasts to the case of $|V_{\text{add}}| = 1$, whose polynomial-time tractability is a simple corollary of one of our lemmas). In particular, the algorithm uses a new two-step "delimit-and-sweep" approach: first, we apply branching to find a curve with specific properties that bounds the instance by a left and right "delimiter". The second step is then a left-to-right sweep of the instance that iteratively pushes the left delimiter towards the right one while performing dynamic programming combined with branching and network-flow subroutines.

Albeit being a special case, we believe these delimited instances with two added vertices can play a role in a potential XP algorithm parameterized by $\kappa$ – the existence of which we leave open for future work.

**Further Related Work.** In addition to the given related work on extension problems, it is also worth noting that identifying a substructure of bounded treewidth and applying Courcelle's Theorem to decide an MSO formula on it has been preciously used for a graph drawing problem by Grohe [21], namely to identify graph drawings of bounded crossing number. Both the way in which one arrives at bounded treewidth and the nature of the employed MSO formula are substantially different from our approach, which is not surprising as the problem of generating drawings from scratch and the problem of extending partial drawings are in general fundamentally different. Specifically in the case of generating drawings, the MSO formula could essentially encode the existence of a drawing with bounded crossing number by inductively planarizing crossings of pairs of edges; here the planarity of the planarization can of course be captured via excluded $K_{3,3}$ and $K_5$ minors by MSO. This approach is not possible in our setting. There are examples of 1-planar graphs which have partial drawings which cannot be extended to a 1-plane drawing. Thus a planarization with respect to the added parts of a solution needs to be compatible with the partial drawing and cannot be encoded by an MSO formula straightforwardly.

## 2 Preliminaries

Let $G$ be a simple graph, $V(G)$ its vertices, and $E(G)$ its edges. We use standard graph terminology [18]. For $r \in \mathbb{N}$, we write $[r]$ as shorthand for the set $\{1, \ldots, r\}$. We also assume a basic understanding of parameterized complexity theory [15, 19], Monadic Second Order (MSO) Logic and Courcelle's Theorem [14].

A *drawing* $\mathcal{G}$ of $G$ in the plane $\mathbb{R}^2$ is a function that maps each vertex $v \in V(G)$ to a distinct point $\mathcal{G}(v) \in \mathbb{R}^2$ and each edge $e = uv \in E(G)$ to a simple open curve $\mathcal{G}(e) \subset \mathbb{R}^2$ with endpoints $\mathcal{G}(u)$ and $\mathcal{G}(v)$. In a slight abuse of notation we often identify a vertex $v$ and its drawing $\mathcal{G}(v)$ as well as an edge $e$ and its drawing $\mathcal{G}(e)$. Throughout the paper we will assume that: (i) no edge passes through a vertex other than its endpoints, (ii) any two edges intersect in at most one point, which is either a common endpoint or a proper *crossing* (i.e., edges cannot touch), and (iii) no three edges cross in a single point. For a drawing $\mathcal{G}$ of $G$ and $e \in E(G)$, we use $\mathcal{G} - e$ to denote the drawing of $G - e$ obtained by removing the drawing of $e$ from $\mathcal{G}$, and for $J \subseteq E(G)$ we define $\mathcal{G} - J$ analogously.

We assume that readers are familiar with the notion of *planarity* and *faces*. The *boundary* of a face is the set of edges and vertices whose drawing delimits the face. Further, $\mathcal{G}$ induces for each vertex $v \in V(G)$ a cyclic order of its neighbors by using the clockwise order of its incident edges. This set of cyclic orders is called a *rotation scheme*. Two planar drawings $\mathcal{G}_1$ and $\mathcal{G}_2$ of the same graph $G$ are *equivalent* if they have the same rotation scheme and the same outer face; equivalence classes of planar drawings are also called *embeddings*. A *plane* graph is a planar graph with a fixed embedding.

A drawing $\mathcal{G}$ is *1-planar* if each edge has at most one crossing and a graph $G$ is *1-planar* if it admits a 1-planar drawing. Similarly to planar drawings, 1-planar drawings subdivide the plane into connected regions, which we call *cells* in order to distinguish them from the faces of a planar drawing. The *planarization* $G^\times$ of a 1-planar drawing $\mathcal{G}$ of $G$ is a graph $G^\times$ with $V(G) \subseteq V(G^\times)$ that introduces for each crossing $c$ of $\mathcal{G}$ a *dummy vertex* $c \in V(G^\times)$ and that replaces each pair of crossing edges $uv, wx$ in $E(G)$ by the four *half-edges* $uc, vc, wc, xc$ in $E(G^\times)$, where $c$ is the crossing of $uv$ and $wx$. In addition all crossing-free edges of $E(G)$ belong to $E(G^\times)$. Obviously, $G^\times$ is planar and the drawing $\mathcal{G}^\times$ of $G^\times$ corresponds to $\mathcal{G}$ with the crossings replaced by the dummy vertices.

## 3 Extending 1-Planar Drawings

Given a graph $G$ and a subgraph $H$ of $G$ with a 1-planar drawing $\mathcal{H}$ of $H$, we say that a drawing $\mathcal{G}$ of $G$ is an *extension* of $\mathcal{H}$ if the planarization $H^\times$ of $\mathcal{H}$ and the planarization $\mathcal{G}^\times$ of $\mathcal{G}$ restricted to $\mathcal{H}^\times$ have the same embedding. We formalize our problem of interest as:

> 1-Planar Drawing Extension
> *Instance:* A graph $G$, a connected subgraph $H$ of $G$, and a 1-planar drawing $\mathcal{H}$ of $H$.
> *Task:* Find an 1-planar extension of $\mathcal{H}$ to $G$, or correctly identify that there is none.

The IC-Planar Drawing Extension problem is then defined analogously. Both problem definitions follow previously considered drawing extension problems, where the connectivity of $H$ is considered a well-motivated and standard assumption [22, 30, 31].

Given an instance $(G, H, \mathcal{H})$ of 1-Planar Drawing Extension, a *solution* is a 1-planar drawing $\mathcal{G}$ of $G$ that is an extension of $\mathcal{H}$. We refer to $V_{\text{add}} := V(G) \setminus V(H)$ as the *added vertices* and to $E_{\text{add}} := E(G) \setminus E(H)$ as the *added edges*. Let $V_{\text{inc}} = \{v \in V(H) \mid \exists vw \in E_{\text{add}}\}$, i.e., $V_{\text{inc}}$ is the set of vertices of $H$ that are incident to at least one added edge. We also distinguish added edges whose endpoints are already part of the drawing, and added edges with at least one endpoint yet to be added into the drawing – notably, we let $E^H_{\text{add}} := \{vw \in E_{\text{add}} \mid v, w \in V(H)\}$ and $E^{\neg H}_{\text{add}} := E_{\text{add}} \setminus E^H_{\text{add}}$. This distinction will become important later, since it opens up two options for how to quantify how "complete" the drawing of $\mathcal{H}$ is. It is worth noting that, without loss of generality, we may assume each vertex in $V_{\text{add}}$ to be incident to at least one edge in $E_{\text{add}}$ and hence $|V_{\text{add}} \cup V_{\text{inc}}| \le 2|E_{\text{add}}|$.

Given the NP-completeness of recognizing 1-planar [20, 28] and IC-planar [6] graphs we get as an immediate consequence that also the corresponding extension problems are NP-complete. In view of the NP-completeness of the problem, it is natural to ask about its complexity when $H$ is nearly "complete", i.e., we only need to extend the drawing $\mathcal{H}$ by a small part of $G$. In this sense, deletion distance represents the most immediate way of quantifying how far $H$ is from $G$, and the parameterized complexity paradigm [15, 19] offers complexity classes that provide a more refined view on "tractability" in this setting.

The most immediate way of capturing the completeness of $H$ in this way is to parameterize the problem via the *edge deletion distance* to $G$ – formalized by setting $k = |E_{\text{add}}|$. The aim of Section 4 is to establish the fixed-parameter tractability of 1-Planar Drawing Extension parameterized by $k$. A second parameter that we consider is the *vertex+edge deletion distance* to $G$, i.e., the minimum number of vertices and edges that need to be deleted from $G$ to obtain $H$. We call this parameter $\kappa$ and set $\kappa = |V_{\text{add}}| + |E^H_{\text{add}}|$. The parameterization by $\kappa$ is the topic of Section 5 and 6. Since we can always assume that each added vertex is incident to at least one added edge, $|V_{\text{add}}| + |E^H_{\text{add}}| \le |E_{\text{add}}|$ and so parameterizing by $\kappa$ leads to a more general (and difficult) parameterized problem.

## 4 Using Edge Deletion Distance for Drawing Extensions

The main goal of this section is to establish the fixed-parameter tractability of 1-Planar Drawing Extension parameterized by the edge deletion distance $k$.

We note that one major obstacle faced by a fixed-parameter algorithm is that it is not at all obvious how to decide where the vertices in $V_{\text{add}}$ should be drawn in an augmented drawing of $H$. As a follow-up, we will show that when $V_{\text{add}} = \emptyset$ (i.e., $V(H) = V(G)$), it is possible to obtain a more self-contained combinatorial algorithm with a significantly better runtime; this is presented in Subsection 4.2.

### 4.1 A Fixed-Parameter Algorithm for 1-Planar Drawing Extension

Our first step towards a proof of the desired tractability result is the definition of a *pattern*, which is a combinatorial object capturing essential information about a potential 1-planar extension of $\mathcal{H}$. The formal definition of pattern is given in Definition 1. Definition 2 then defines the notion of *derived patterns*, which create a link between solutions to an instance of 1-Planar Drawing Extension and patterns.

To given an intuition of the patterns, assume that a pattern consists of a tuple $(S, Q, C)$ and let $(G, H, \mathcal{H})$ be a 1-Planar Drawing Extension instance. Then, the general intuition is that $S$ represents the set of faces in $\mathcal{H}^\times$ which contain at least a part of the drawing of an edge in $E_{\text{add}}$ in a hypothetical 1-planar extension $\mathcal{G}$ of $\mathcal{H}$. Crucially, our aim is to keep the size of patterns bounded in $k$, and so we only "anchor" $S$ to $\mathcal{H}^\times$ by storing information about which faces will contain individual edges in $E_{\text{add}}$, vertices from $V_{\text{add}}$, and be adjacent to individual vertices in $V_{\text{inc}}$; this is captured by the mapping $Q$. The third piece of information we store is $C$, which represents the cyclic order of how edges in $E_{\text{add}}$ exit or enter the boundary of each face (including the case where an edge crosses through an edge into the same face, i.e., occurs twice when traversing the boundary of that face).

▶ **Definition 1.** *A* pattern *for an instance $(G, H, \mathcal{H})$ is a tuple $(S, Q, C)$ where*

1. *$S$ is a set of at most $2k$ elements;*
2. *$Q$ is a mapping from $V_{add} \cup E_{add} \cup V_{inc}$ which maps:* **(a)** *vertices in $V_{add}$ to elements of $S$,* **(b)** *edges in $E_{add}$ to ordered pairs of elements of $S$, and* **(c)** *vertices in $V_{inc}$ to subsets of $S$.*
3. *$C$ is a mapping from $S$ that maps each $s \in S$ to a cyclically ordered multiset of pairs $((e_1, q_1), (e_2, q_2), \ldots, (e_\ell, q_\ell))$, where each $e_i$ is in $E_{add}$ and each $q_i$ is in $V_{inc} \cup \{\text{crossing}\}$. Moreover, $C$ must satisfy the following conditions:* **(a)** *for each $s \in S$ and each tuple $(e, q) \in C(s)$ such that $q \in V_{inc}$, it must hold that $s \in Q(q)$ and $e$ is incident to $q$ in $G$;* **(b)** *for each $e \in E_{add}$ and $s \in S$, if $e$ occurs in at least one tuple in $C(s)$, then $s \in Q(e)$ and $C(s)$ contains at most two tuples of the form $(e, *)$, where $*$ is an arbitrary element;* **(c)** *for each $s \in S$, each tuple occurs at most once in $C(s)$ with the exception of tuples containing "*crossing*", which may occur twice.*

Let $\mathcal{P}$ be the set of all patterns for our considered instance $(G, H, \mathcal{H})$. Let $\#\text{pat}(k) = 2k \cdot (2^{2k})^{3k} \cdot ((2k)! \cdot 2^{3k})^{2k}$ and note that $|\mathcal{P}| \leq \#\text{pat}(k) \in 2^{\mathcal{O}(k^2 \log k)}$. In particular, the number of possible patterns can be bounded by first considering $2k$ options for $|S|$, multiplying this by the at most $(2^{2k})^{3k}$-many ways of choosing $Q$, and finally multiply this by the number of choices for $C$ which can be bounded as follows: for each $s \in S$, $C(s)$ is a set that forms a subset (of size at most $2k$) of the $3k$-cardinality set of tuples (note that $e = \{a, b\} \in E_{\text{add}}$ can only occur in the tuples $(e, a)$, $(e, b)$ and $(e, \text{crossing})$).

The intuition behind patterns will be formalized in the next definition, which creates a link between solutions to our instance and patterns.

▶ **Definition 2.** *Let $(G, H, \mathcal{H})$ be a 1-Planar Drawing Extension instance. For each solution $\mathcal{G}$ of $(G, H, \mathcal{H})$ we define a derived pattern $P = (S, Q, C)$ as follows:*

- *$S$ is the set of faces of $\mathcal{H}^\times$ which have a non-empty intersection with $\mathcal{G}(e)$ for some $e \in E_{add}$.*
- *For $v \in V_{add}$ we set $Q(v)$ to the face $f$ of $\mathcal{H}^\times$ for which $\mathcal{G}(v)$ lies inside $f$, for $e \in E_{add}$ we set $Q(e)$ to the set of at most two faces which have a non-empty intersection with $\mathcal{G}(e)$, and for $w \in V_{inc}$ we set $Q(w)$ to all faces in $S$ incident to $w$ in $\mathcal{G}$.*
- *For a face $s \in S$ we consider all edges $e = uv \in E_{add}$ with a non-empty intersection between $\mathcal{G}(e)$ and $s$. It follows that there is an edge $e' \in E(H)$ on the boundary of $s$ such that $\mathcal{G}(e)$ crosses $\mathcal{G}(e')$, or $u \in V_{inc}$ and $u$ is on the boundary of $s$, or both. We set $C(s)$ as the ordered set of these crossing points or vertices when traversing $s$ in clockwise fashion.*

Our next task is to define *valid patterns*; generally speaking, these are patterns which are not malformed and could serve as derived patterns for a hypothetical solution. One notable property that every valid pattern must satisfy is that all vertices and edges mapped by $Q$ to some $s \in S$ can be drawn in a 1-planar way while respecting $C(s)$.

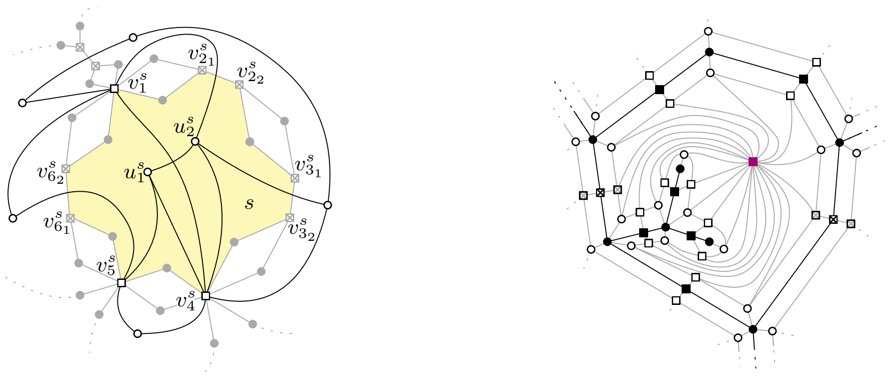
▶ **Definition 3.** *For an instance $(G, H, \mathcal{H})$, a pattern $P = (S, Q, C)$ is* valid *if there exists a* pattern graph *$G_P$ with a 1-planar drawing $\mathcal{G}_P$ satisfying the following properties:*

- *$V_{add} \cup V_{inc} \subseteq V(G_P)$ and $E_{add} \subseteq E(G_P)$.*
- *$\mathcal{G}_P - E_{add}$ is a planar drawing.*
- *$S$ is a subset of the inner faces of $\mathcal{G}_P - E_{add}$.*
- *Each $v \in V_{add}$ is contained in the face $Q(v)$ of $\mathcal{G}_P - V_{add} - E_{add}$.*
- *Each $e \in E_{add}$ is contained in the face(s) $Q(e)$ of $\mathcal{G}_P - E_{add}$.*
- *Each $v \in V_{inc}$ is incident to the faces $Q(v)$ of $\mathcal{G}_P - E_{add}$.*
- *When traversing the inner side of the boundary of each face $s$ of $\mathcal{G}_P - E_{add}$ in clockwise fashion, the order in which each edge $e \in E_{add}$ is seen in $\mathcal{G}_P$ together with the information whether $e$ crosses here or ends in its endpoint in $V_{inc}$, is precisely $C(s)$.*

Note that the instance $(G, H, \mathcal{H})$ in the definition of a valid pattern is only important to define $V_{\mathrm{add}}$, $V_{\mathrm{inc}}$, and $E_{\mathrm{add}}$. Moreover, observe that for each solution $\mathcal{G}$ of an instance $(G, H, \mathcal{H})$, the derived pattern is valid by definition. An illustration of a pattern graph is provided in Part (a) of Figure 1. We also remark that, when comparing a pattern graph to a hypothetical solution which draws an edge into the outer face of $\mathcal{H}$, we will map the outer face to an inner face of the pattern graph.

▶ **Lemma 4.** *Given pattern $P = (S, Q, C)$, in time $\mathcal{O}((k!)^k \cdot k^{2k+1})$ we can either construct a pattern graph $G_P$ together with the drawing $\mathcal{G}_P$ satisfying all the properties of Definition 3 or decide that $P$ is not valid.*

**Proof Sketch.** The idea is to build a planarized version $G'_P$ of the pattern graph with size bounded in $\mathcal{O}(k)$. To build the graph we introduce for every $s \in S$ a cycle containing the vertices in $C(s)$. We further subdivide each edge on a cycle by dummy vertices and identify the vertices corresponding to the same vertex in $V_{\mathrm{inc}}$ or the same crossing. Finally we add each vertex in $V_{\mathrm{add}}$ and connect it to the necessary vertices on the face's boundary. Crossings between these in-face edges can then be guessed since there are only $O(k)$ such crossings. Finding the drawing $\mathcal{G}'_P$ that adheres to Definition 3 can then be done by iterating all possible rotation schemes. To turn $\mathcal{G}'_P$ into a 1-plane drawing replace all crossing vertices by crossing




**(a)** A pattern graph as constructed in Lemma 4. The face representing $s \in S$ is yellow, gray disks are dummy vertices. Black circles are in $V_{\text{add}}$. Squares are either in $V_{\text{inc}}$ or represent crossings.

**(b)** An example of an embedding graph. The white vertices are shadow-vertices, the purple one the face vertex, and gray edges got added.

**Figure 1** Examples for the definition of a pattern and the embedding graph.

edges (see also Part (a) of Figure 1a). For the reverse direction, simply observe that from a given pattern graph we can delete and contract all "unecessary" vertices and edges. Then all that remains are the cycles of vertices representing a set $C(s)$ and vertices in $V_{\text{add}}$. ◀

Next, we will define an annotated ("labeled") graph representation of $\mathcal{H}$ and $\mathcal{H}^\times$'s faces. The *embedding graph* $H^*$ of $\mathcal{H}$ is obtained from $\mathcal{H}^\times$ by:

1. subdividing each uncrossed edge $e$ (resulting in vertex $v_e$);
2. creating a vertex for each face $f$ of $\mathcal{H}^\times$ (resulting in vertex $v_f$);
3. traversing the boundary of each face $f$[1] and whenever we see a vertex $v$ (including the vertices created in Step 1) we create a shadow copy of $v$ and place it right next to $v$ in the direction we saw $v$ from. Add a cycle connecting the shadow vertices we created in $f$ in the order they were created, and direct it in clockwise fashion[2];
4. connecting $v_f$ to all shadow-vertices created by traversing $f$, and all shadow copies of a vertex $v$ to the original $v$.

Observe that the embedding graph is a connected plane graph. We label the vertices of the embedding graph to distinguish original vertices, edge-vertices, face-vertices, crossing-vertices and shadow-vertices, and use at most $2k$ special labels to identify vertices in $V_{\text{inc}}$. An illustration of the embedding graph is provided in Part (b) of Figure 1. Next, we show that it suffices to restrict our attention to the parts of $H^*$ which are "close" to vertices in $V_{\text{inc}}$.

▶ **Lemma 5.** *Let $I = (G, H, \mathcal{H})$ be an instance of 1-*Planar Drawing Extension*. Let $Z$ be the set of all vertices in $H^*$ of distance at least $4k+7$ from each vertex in $V_{inc}$. Let $G'$, $H'$, and $\mathcal{H}'$ be obtained by deleting all vertices in $Z$ from $G$, $H$, and $\mathcal{H}$ respectively. Then:*

1. *If $I$ is a YES-instance, then each connected component of $G'$ contains at most one connected component of $H'$[3];*

[1] formally, we draw a curve in $f$ that closely follows the boundary until it forms a closed curve.
[2] Note that this may create multiple shadow copies of a vertex. The reason we use shadow copies of vertices instead of using the original vertices is that when traversing the inner boundary of a face, a vertex may be seen multiple times, and such shadow-vertices allow us to pinpoint from which part of the face we are visiting the given vertex.
[3] This can be seen not to hold in general if we allow $H$ to be disconnected.

2. *$I$ is a YES-instance if and only if for each connected component $A$ of $H'$ the restriction of $\mathcal{H}'$ to $H'[A]$ can be extended to a drawing of the connected component of $G'$ containing $A$. Moreover, given such a 1-planar extension for every connected component of $G'$, we can output a solution for $I$ in linear time.*

We split the proof of Lemma 5 into proofs for the two individual points.

**Proof of Point 1.** For the sake of contradiction let $J$ be a connected component of $G'$ that contains two distinct connected components $H'_1$ and $H'_2$ of $H'$. Since $J$ is a connected component, there must be a path $P$ from a vertex $v_1 \in H'_1$ to a vertex $v_2 \in H'_2$ in $J - (H'_1 \cup H'_2)$, and moreover $P$ must have length at most $k$. By definition, both $v_1$ and $v_2$ are in $V_{\text{inc}}$. To complete the proof, it suffices to show that in any solution $\mathcal{G}$, $v_1$ and $v_2$ have distance at most $4k+4$ in $H^*$.

Moreover, in any solution $\mathcal{G}$, two consecutive vertices of $P$ are either drawn in the same face of $\mathcal{H}^\times$ or in two adjacent faces of $\mathcal{H}^\times$. Observe that the distance in $H^*$ between two face-vertices for the faces that share an edge is 4, and that the distance from an original vertex $v$ to a face-vertex of a face incident to $v$ is 2. Therefore, if $(G, H, \mathcal{H})$ is a YES-instance, then the distance between $v_1$ and $v_2$ in $H^*$ must be at most $4k+4$. ◀

**Proof of Point 2.** The forward direction is obvious. For the backward direction, let $G_1, \dots, G_r$ be the connected components of $G'$ and for $i \in [r]$ let $H_i$ and $\mathcal{H}_i$ be the restriction of $H'$ and $\mathcal{H}'$, respectively, to $G_i$. Moreover, let $\mathcal{H}_i^\times$ be the planarization derived from $\mathcal{H}_i$ and note that $H_i$ is connected for all $i \in [r]$ by Point 1. Now let us fix an arbitrary $i \in [r]$ such that $H_i$ is not empty and let $\mathcal{G}_i$ be a 1-planar extension of $\mathcal{H}_i$ to $G_i$.

Observe that each face of $\mathcal{H}^\times$ is completely contained in precisely one face of $\mathcal{H}_i^\times$. Moreover, if a face $f$ of $\mathcal{H}_i^\times$ contains at least two faces $f_1$ and $f_2$ of $\mathcal{H}^\times$, then both $v_{f_1}$ and $v_{f_2}$ are at distance at least $4k+4$ of any vertex in $V_{\text{inc}} \cap V(H_i)$ in $H^*$. Indeed, if this were not the case, then w.l.o.g. the vertices on the boundary of $v_{f_1}$ would have distance at most $4k+6$ from some $w \in V_{\text{inc}} \cap V(H_i)$ in $H^*$, which would mean that $f_1$ is also a face in $\mathcal{H}_i^\times$. By the same distance-counting argument introduced at the end of the Proof of Point 1, This implies that no edge in a path $P$ of $G$ from a vertex $v \in H_i$ whose internal vertices all lie in $V_{\text{add}}$ can be drawn in any face of $\mathcal{H}^\times$ contained in $f$.

To complete the proof, let $G_1, \dots, G_p$, $p \le r$ be the connected components of $G'$ that contain a vertex in $H$ and $G_{p+1}, \dots, G_r$ the remaining connected components of $G'$. We obtain a solution $\mathcal{G}$ to the instance $I$ by simply taking the union of $\mathcal{H}$ and $\mathcal{G}_i$ for $i \in [p]$ and then for $i \in \{p+1, \dots, r\}$ shifting $\mathcal{G}_i$ so that $\mathcal{G}_i$ do not intersect any other part of the drawing. ◀

Since $|V_{\text{inc}}| \le 2k$, Lemma 5 allows us to restrict our attention to a subgraph of diameter at most $(4k+7) \cdot 2 \cdot 2k = 16k^2 + 28k$. This will be especially useful in view of the following known fact, that allows us to assume that the treewidth of our instances is bounded.

▶ **Proposition 6** ([34]). *A planar graph $G$ with radius at most $r$ has treewidth at most $3r+1$.*

▶ **Lemma 7.** *1-Planar Drawing Extension is* FPT *parameterized by $k + \text{tw}(H^*)$ if and only if it is* FPT *parameterized by $k$, where $H^*$ is the embedding graph of $\mathcal{H}$.*

**Proof.** The backward direction is trivial. For the forward direction, assume that that there exists an algorithm $\mathcal{B}$ which solves 1-Planar Drawing Extension in time $f(k + \text{tw}(H^*)) \cdot |V(G)|^c$ for some constant $c$ and computable function $f$. Now, consider the following algorithm $\mathcal{A}$ for 1-Planar Drawing Extension: $\mathcal{A}$ takes an instance $(G_0, H_0, \mathcal{H}_0)$ and constructs

$(G_1, H_1, \mathcal{H}_1)$ by applying Lemma 5. Recall that by Point 1 of Lemma 5, $(G_0, H_0, \mathcal{H}_0)$ is either NO-instance, in which case $\mathcal{A}$ correctly outputs "NO", or each connected component of $G_1$ contains at most one connected component of $H_1$.

Now let us consider a connected component $\mathcal{C}$ of $G_1$ and the embedding graph $H_1^*[\mathcal{C}]$ of $\mathcal{H}_1[\mathcal{C}]$ and let $v_f$ be a face-vertex in $H_1^*[\mathcal{C}]$. If $v_f$ is at distance at least $4k+9$ from every vertex in $V_{\text{inc}} \cap \mathcal{C}$ in $H_1^*[\mathcal{C}]$, then every vertex on the boundary of $f$ is at distance at least $4k+7$ from every vertex $w \in V_{\text{inc}} \cap \mathcal{C}$ in $H_1^*[\mathcal{C}]$. Let $v$ be an arbitrary vertex incident to $f$ in $\mathcal{H}_1[\mathcal{C}]$. Since each face of $\mathcal{H}_0^\times$ is completely contained in precisely one face of $\mathcal{H}_1^\times[\mathcal{C}]$, it follows that $v$ is at distance at least $4k+7$ from each vertex $w \in V_{\text{inc}} \cap \mathcal{C}$ in $H^*$. Because $v \in V(H_1[\mathcal{C}])$, this contradicts the fact that every vertex in $V(H_1)$ is at distance at most $4k+6$ from a vertex $w \in V_{\text{inc}}$ in $H^*$. Hence, every face-vertex in $H_1^*[\mathcal{C}]$ is at distance at most $4k+8$ from a vertex in $V_{\text{inc}} \cap \mathcal{C}$. Moreover, every vertex in $H_1^*[\mathcal{C}]$ is at distance at most 2 from some face-vertex and there are at most $2k$ vertices in $V_{\text{inc}} \cap \mathcal{C}$. Therefore, the radius, and by Proposition 6 the treewidth, of $H_1^*[\mathcal{C}]$ is bounded by $\mathcal{O}(k^2)$.

Now, for each connected component $\mathcal{C}$ of $G_1$, we solve the instance $(G_1[\mathcal{C}], H_1[\mathcal{C}], \mathcal{H}_1[\mathcal{C}])$ using algorithm $\mathcal{B}$. If $\mathcal{B}$ determines that at least one such (sub)-instance is a NO-instance, then $\mathcal{A}$ correctly outputs "NO". Otherwise, $\mathcal{A}$ outputs a solution for $(G_0, H_0, \mathcal{H}_0)$ that it computes by invoking the algorithm given by Point 2 of Lemma 5. To conclude, we observe that $\mathcal{A}$ is a fixed-parameter algorithm parameterized by $k$ and its correctness follows from Lemma 5. ◀

We now have all the ingredients we need to establish our tractability result.

▶ **Theorem 1.** *1-Planar Drawing Extension is* FPT *when parameterized by $k$.*

**Proof Sketch.** We prove the theorem by showing that 1-Planar Drawing Extension is fixed-parameter tractable parameterized by $k + \text{tw}(H^*)$, which suffices thanks to Lemma 7.

To this end, consider the following algorithm $\mathcal{A}$. Initially, $\mathcal{A}$ loops over all of the at most $\#\text{pat}(k)$ many patterns, tests whether each pattern is valid or not using Lemma 4, and stores all valid patterns in a set $\mathcal{P}$. Next, it branches over all valid patterns in $\mathcal{P}$, and for each such pattern $P = (S = \{s_1, \ldots, s_\ell\}, Q, C)$ it constructs an MSO formula $\Phi_P(\mathcal{F})$, where $\mathcal{F}$ is a set of at most $7k$ free variables specified later, the purpose of which is to find a suitable "placement" for $P$ in $\mathcal{H}$ by finding an interpretation in the embedding graph $H^*$. In particular, $\Phi_P$ uses the free variables in $\mathcal{F}$ to find a suitable face-vertex $x_i$ for each $s_i \in S$ and a suitable crossing point for each edge mapped to two elements of $S$, while also guaranteeing that the cyclic orders specified by $C$ are adhered to. Once we find a suitable placement for $P$ in $\mathcal{H}$, the algorithm constructs an extension by topologically "inserting" the pattern graph $G_P$ into the identified faces of $\mathcal{H}^\times$ and using the crossing points as well as vertices in $V_{\text{inc}}$ as "anchors". ◀

## 4.2 A More Efficient Algorithm for Extending by Edges Only

In this subsection we obtain a more explicit and efficient algorithm than in Theorem 1 for the case where $V(G) = V(H)$. The idea underlying the algorithm is to iteratively identify sufficiently many 1-planar drawings of each added edge into $\mathcal{H}$ that can either all be extended to a 1-planar drawing of $G$, or none of them can, which allows us to branch over a small number of possible drawings for that edge.

Let $X = \bigcup_{uv \in E_{\text{add}}} \{u, v\}$ be the set of all endpoints of edges in $E_{\text{add}}$, and let us fix an order of the added edges by enumerating $E_{\text{add}} = \{e_1, \ldots, e_k\}$. Now, consider a 1-planar drawing $\mathcal{H}_i$ of $H_i := H + \{e_1, \ldots, e_{i-1}\}$ and assume that we want to add $e_i$ as a curve

$\gamma(e_i)$. For a cell $f$ in $\mathcal{H}_i + \gamma(e_i)$ and vertices $x_1, x_2$ on the boundary of $f$, we denote by $b_{\gamma(e_i)}(f, x_1, x_2) \subseteq E(H_{i+1})$ the edges on the $x_1$-$x_2$-path along the boundary of $f$ which traverses this boundary in counterclockwise direction. We explicitly note that $b_{\gamma(e_i)}(f, x_1, x_2)$ does not contain any half-edges. In this way $b_{\gamma(e_i)}(f, x_1, x_2)$ is the set of edges of $H_{i+1}$ on the $x_1$-$x_2$-path along the boundary of $f$ that are not crossed in $\mathcal{H}_i + \gamma(e_i)$, and hence may still be crossed by drawings of $e_{i+1}, \ldots, e_k$ in a 1-planar extension of $\mathcal{H}_i + \gamma(e_i)$ to $G$.

Let $\gamma_1(e_i)$ and $\gamma_2(e_i)$ be two possible curves for $e_i$ to be drawn into $\mathcal{H}_i$. Then we call $\gamma_1(e_i)$ and $\gamma_2(e_i)$ $\{e_{i+1}, \ldots, e_k\}$*-partition equivalent* if there is a bijection $\pi$ from the cells of $\mathcal{H}_i + \gamma_1(e_i)$ to the cells of $\mathcal{H}_i + \gamma_2(e_i)$ such that

- the vertices in $X$ on the boundaries of the cells are invariant under $\pi$, i.e., for each cell $f$ whose boundary intersects $X$ precisely in $X'$ it must hold that $\pi(f)$ intersects $X$ precisely in $X'$ as well; and
- for each pair of cells $f, f'$ of $\mathcal{H}_i + \gamma_1(e_i)$ and ordered pairs of (not necessarily pairwise distinct) vertices $(x_1, x_2), (x_1', x_2') \in X^2$ that
  - are on the boundary of $f$ and $f'$, respectively, and
  - the counterclockwise $x_1$-$x_2$-path and the counterclockwise $x_1'$-$x_2'$-path along the boundaries of $f$ and $f'$, respectively, does not contain any inner vertices in $X$,

  the following must hold:

$$\begin{cases} \text{if } |b_{\gamma_1(e_i)}(f, x_1, x_2) \cap b_{\gamma_1(e_i)}(f', x_1', x_2')| \le k, \text{ then} \\ \quad |b_{\gamma_1(e_i)}(f, x_1, x_2) \cap b_{\gamma_1(e_i)}(f', x_1', x_2')| = |b_{\gamma_2(e_i)}(\pi(f), x_1, x_2) \cap b_{\gamma_2(e_i)}(\pi(f'), x_1', x_2')| \\ \text{otherwise} \\ \quad \text{also } |b_{\gamma_2(e_i)}(\pi(f), x_1, x_2) \cap b_{\gamma_2(e_i)}(\pi(f'), x_1', x_2')| > k. \end{cases}$$

Roughly speaking, the first condition guarantees that when extending $\mathcal{H}_i$ by $\{e_{i+1}, \ldots, e_k\}$-partition equivalent drawings of $e_i$, the topological separation of all vertices that might be important when drawing $e_{i+1}, \ldots, e_k$ is the same. The second condition ensures that when extending $\mathcal{H}_i$ by $\{e_{i+1}, \ldots, e_k\}$-partition equivalent drawings of $e_i$, the number of edges whose drawings might be crossed by drawings of $\{e_{i+1} \ldots, e_k\}$ is the same, or so large that they cannot all be crossed by drawings of $\{e_{i+1} \ldots, e_k\}$.

▶ **Lemma 8.** *For any $1 \le i \le k$, if two drawings $\gamma_1(e_i), \gamma_2(e_i)$ of $e_i$ into a drawing $\mathcal{H}_i$ of $H_i$ are $\{e_{i+1}, \ldots, e_k\}$-partition-equivalent, they either both can be extended to a 1-planar drawing of $G$, or none of them can.*

**Proof.** We show that we can obtain a 1-planar drawing extension of $\mathcal{H}_i + \gamma_2(e_i)$ to $G = H_i + \{e_{i+1}, \ldots, e_k\}$ from a 1-planar drawing extension of $\mathcal{H}_i + \gamma_1(e_i)$ to $G$. Then the claim immediately follows by a symmetric argument when $\gamma_1(e_i)$ and $\gamma_2(e_i)$ are interchanged.

Let $\pi$ be a bijection between the cells of $\mathcal{H}_i + \gamma_1(e_i)$ and the cells of $\mathcal{H}_i + \gamma_2(e_i)$ that witnesses $\{e_{i+1}, \ldots, e_k\}$-partition equivalence of $\gamma_1(e_i)$ and $\gamma_2(e_i)$. Assume we are given a 1-planar drawing extension $\mathcal{G}_1$ of $\mathcal{H}_i + \gamma_1(e_i)$ to $G$. From this, we will define a 1-planar drawing extension $\mathcal{G}_2$ of $\mathcal{H}_i + \gamma_2(e_i)$ to $G$. For $e \in E(H_i)$ set $\mathcal{G}_2(e) = \mathcal{H}_i(e)$ and set $\mathcal{G}_2(e_i) = \gamma_2(e_i)$. In this way, $\mathcal{G}_2$ is an extension of $\mathcal{H}_i$.

Note that for any cell $f$ of $\mathcal{H}_i + \gamma_1(e_i)$ the order in which the vertices of $X$ occur on the boundary of $f$ is the same (up to possibly reversal) in which they occur on the boundary of $\pi(f)$ (exactly the same such vertices occur because of $\{e_{i+1}, \ldots, e_k\}$-partition equivalence). This is due to the fact that $\mathcal{H}_i + \gamma_1(e_i)$ and $\mathcal{H}_i + \gamma_2(e_i)$ are obtained from the same drawing $\mathcal{H}_i$ and drawing edges into $\mathcal{H}_i$ merely subdivides cells and cannot permute the order on their boundaries.

Now we can define $\mathcal{G}_2(e_j)$ for $j \in \{i+1,\dots,k\}$ as follows: For $J \subseteq \{i+1,\dots,k\}$ such that $\mathcal{G}_1(e_j)$ intersects two cells $f$ and $g$ of $\mathcal{H}_i + \gamma_1(e_i)$ for every $j \in J$, it holds that each $\mathcal{G}_1(e_j)$ crosses the drawing $(\mathcal{H}_i + \gamma_1(e_i))(c_j)$ of an edge $c_j \in E(H_i) \cup \{e_i\}$. In particular, $c_j$ lies on the shared boundary of $f$ and $g$. Both $f$ and $g$ contain a vertex in $X$ in their boundary, as each of them contain at least one endpoint of $e_j$. Hence there are $x_1, x_2 \in X$ that are consecutive on the boundary of $f$ neglecting everything but $X$, and $y_1, y_2 \in X$ that are consecutive on the boundary of $g$ neglecting everything but $X$ such that $c_j \in b_{\gamma_1(e_i)}(f, x_1, x_2) \cap b_{\gamma_1(e_i)}(g, y_1, y_2)$. By partition-equivalence the boundaries of $\pi(f)$ and $\pi(g)$ each contain an endpoint of each $e_j$, and because $|J| \leq k$, we find distinct $c'_j \in b_{\gamma_2(e_i)}(\pi(f), x_1, x_2) \cap b_{\gamma_2(e_i)}(\pi(g), y_1, y_2)$ (or possibly $c'_j \in b_{\gamma_2(e_i)}(\pi(f), x_2, x_1) \cap b_{\gamma_2(e_i)}(\pi(g), y_2, y_1)$) for each $j \in J$. Without loss of generality the $c_j$ are indexed in the order in which they occur on the counterclockwise $x_1$-$x_2$-path along the boundary of $f$. We re-index the $c'_j$ to conform to the same order (up to reversal), also taking $x_1$ and $x_2$ into account, on $\pi(f)$. ◀

The next lemma shows that the number of non-equivalent drawings is bounded by a function of $k$, which in turn allows us to apply exhaustive branching to prove the theorem.

▶ **Lemma 9.** *For any $1 \leq i \leq k$, the number of ways to draw $e_i$ into a drawing $\mathcal{H}_i$ of $H_i$ that are pairwise not $\{e_{i+1},\dots,e_k\}$-partition-equivalent is at most $4(2k+1) \cdot 2(k+1) \in \mathcal{O}(k^2)$.*

▶ **Theorem 2.** *1-Planar Drawing Extension parameterized by $k$ can be solved in time $\mathcal{O}(k^{2k} \cdot n^{\mathcal{O}(1)})$ if $V(G) = V(H)$.*

**Proof.** We can pre-compute the intersection of the boundary of each cell of $\mathcal{H}$ with $X$ and for each pair of cells $f, f'$ of $\mathcal{H}$ and ordered pairs of vertices $x_1, x_2 \in X$ and $x'_1, x'_2$ that are consecutive on the boundaries of $f$ and $f'$ respectively if one neglects everything but $X$, the cardinality of the set of edges that are on the clockwise $x_1$-$x_2$-path along the boundary of $f$ and at the same time on the clockwise $x'_1$-$x'_2$-path along the boundary of $f'$ in polynomial time.

As described in the proof of Lemma 9, at any stage, for $1 \leq i \leq k$, we can branch on $\{e_{i+1},\dots,e_k\}$-partition-equivalent drawings $\gamma(e)$ of $e$ using the pre-computed information. This information can be modified within each branch according to the choice of $\gamma(e)$ in constant time because, as described in the proof of Lemma 9 the impact of $\gamma(e)$ involves only few values whose modifications can correctly be computed from the updated pre-computed information up to this stage and the chosen values determining $\gamma(e)$. Correctness of this branching follows from Lemma 8. ◀

## 5 Using Vertex+Edge Deletion Distance for IC-Planar Drawing Extension

In this section, we show that IC-Planar Drawing Extension parameterized by $\kappa$ is fixed-parameter tractable. We note that an immediate consequence of this is the fixed-parameter tractability of IC-Planar Drawing Extension parameterized by $k$.

On a high level, our strategy is similar to the one used to prove Theorem 1, in the sense that we also use a (more complicated) variant of the patterns along with Courcelle's Theorem. However, obtaining the result requires us to extend the previous proof technique to accommodate the fact that the number of edges incident to $V_{\text{add}}$, and hence the size of a pattern, is no longer bounded by $\kappa$. This is achieved by identifying so-called *difficult vertices* and *regions* that split up the neighborhood of each face-vertex in the embedding graph into a small number of sections (a situation which can then be handled by a formula in Monadic

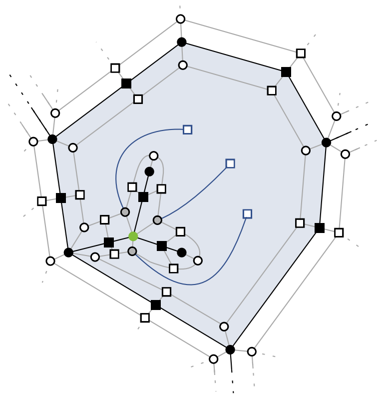


**Figure 2** An example of $H^*$ where a vertex $v' \in V_{\text{inc}}$ has several non-difficult shadow copies. Blue vertices are in $V_{\text{add}}$. The green vertex has no difficult shadow copy w.r.t. the blue face in $\mathcal{H}^\times$.

Second Order logic). Less significant complications are that we need a stronger version of Lemma 5 to ensure that the diameter of the resulting graph is bounded, and need to be more careful when using MSO logic in the proof of the main theorem.

Let $f$ be a face of $\mathcal{H}^\times$ and let $\mathcal{G}$ be a solution (i.e., an IC-planar drawing of $G$) for the instance $(G, H, \mathcal{H})$. Let $H^*$ be the embedding graph of $\mathcal{H}$, and without loss of generality let us assume (via topological shifting) that each edge between a vertex $a'$ on the boundary of $f$ and a vertex $b \in V_{\text{add}}$ placed by $\mathcal{G}$ in $f$ is routed "through" one shadow copy of $a'$[4]. Let $V^f_{\text{add}}$ be the subset of $V_{\text{add}}$ drawn by $\mathcal{G}$ in the face $f$.

Observe that, since shadow vertices are not part of the original instance and instead merely mark possible "parts" of the face that can be used to access a given vertex, it may happen that a solution routes several edges through one shadow vertex. We say that a shadow vertex $v \in N_{H^*}(v_f)$ (where $N_{H^*}(v_f)$ denotes the neighborhood of $v_f$ in $H^*$) is *difficult* w.r.t. $f$ if $\mathcal{G}$ routes at least two edges through $v$. Note that it may happen that a vertex $v' \in V_{\text{inc}}$ with more than one neighbor in $V_{\text{add}}$ has several shadow copies, none of which are difficult (see Figure 2).

▶ **Lemma 10.** *There are at most $3\kappa^2$ difficult vertices w.r.t. a face $f$ of $\mathcal{H}^\times$.*

**Proof.** We show that any two of the $\ell$ added vertices drawn into $f$ in $\mathcal{G}$ are both connected to at most 3 vertices in $N_{H^*}(v_f)$. Then the claim follows. Assume for contradiction that $v_1, v_2 \in V_{\text{add}}$ are drawn into $f$ in $\mathcal{G}$ and $w_1, w_2, w_3, w_4 \in N_{H^*}(v_f)$ are shadow vertices that each route two edges, one of which is incident to $v_1$ and one of which is incident to $v_2$. Since $H$ is connected, the boundary of $f$ is connected, and by construction of $H^*$, $w_1, w_2, w_3, w_4$ all lie on a cycle in $H^*$ that does not involve any of $\{v_{f'} \mid f' \text{ face in } \mathcal{H}^\times\}$. Hence the following graph $H'$ is a minor of $H^* - \{v_{f'} \mid f' \text{ face in } \mathcal{H}^\times\} + V_{\text{add}} + E_{\text{add}}$: $H' = (\{v_1, v_2, w_1, \ldots, w_4, v_f\}, \{v_i w_j \mid i \in \{1, 2, f\}, j \in \{1, \ldots, 4\} \cup \{w_i w_{(i \bmod 4)+1} \mid i \in \{1, \ldots, 4\})$. $H'$ does not admit a 1-planar drawing in which both $v_1$ and $v_2$ lie on the same side of the drawing of the $w_1$-$w_2$-$w_3$-$w_4$-cycle and $v_1$ and $v_2$ are each incident to at most one edge whose drawing is crossed. However the existence of $\mathcal{G}$ implies that exactly such a drawing of $H'$ exists. ◀

[4] The reason one distinguishes which shadow copy of $a'$ the edge is routed through is because this unambiguously identifies which part of the face the edge uses to access $a'$.

A region $R$ of a vertex $x \in V_{\text{add}}^f$ (or, equivalently, of a face $f$) is a maximal path $(r_1, \ldots, r_p)$ in $N_{H^*}(v_f)$ with the following properties: (1) $\mathcal{G}$ does not route through any shadow copy of an edge in $R$; (2) for each vertex $r$ in $R$, a uncrossed curve can be drawn in $\mathcal{G}$ inside $f$ between $r$ and $x$; (3) none of the vertices in $R$ are adjacent to $V_{\text{add}}^f \setminus \{x\}$; and (4) $r_1$ and $r_p$ are adjacent to $x$.

▶ **Lemma 11.** *There are at most $3\kappa$ regions of a face $f$.*

**Proof.** Consider a path $P$ in $H^*[N_{H^*}(v_f)]$ that traverses all $\ell$ regions of a vertex $v \in V_{\text{add}}^f$. It contains at least $\ell - 1$ pairwise disjoint subpaths $P_1, \ldots, P_{\ell-1}$ of paths connecting regions of $v$ that are consecutive in $P$.

For every $P_i$ ($i \in \{1, \ldots, \ell\}$), by the property that regions are inclusion maximal paths of vertices with certain properties, we find some vertex $x$ in $P_i$ that has to violate one of these properties. This can happen in three ways:

**(1)** $x$ is a shadow copy of an edge and $\mathcal{G}$ routes through $x$,
**(2)** (1) is not the case and the drawing of some edge $e \in E_{\text{add}}$ separates $x$ from $v$ in $\mathcal{H}$, or
**(3)** (1) and (2) are not the case and $x$ is adjacent to another vertex $w \in V_{\text{add}}^f \setminus \{v\}$.

In case (1) there is an edge $e \in E_{\text{add}}$ such that the drawing of $e$ crosses the boundary of $f$ in $\mathcal{H}$ and routes through $x$.

In case (2) the edge in question has both endpoints on $P$, thus $e \in E_{\text{add}}^H$ or the drawing of $e$ crosses either the boundary of $f$ in $\mathcal{H}$ or a drawing of another added edge.

There are at most $|E_{\text{add}}^H|$ edges in $E_{\text{add}}^H$ and at most $|V_{\text{add}}|$ edges in $E_{\text{add}} \setminus E_{\text{add}}^H$ that can cross another edge in an IC-planar drawing. Moreover, in both cases (1) and (2), the endpoints of $e$ cannot occur in any $P_j$ with $j \in \{1, \ldots, \ell\} \setminus \{i\}$.

In case (3) either $x$ is contained in a region of $w$ or the drawing of $xw$ in $\mathcal{H}$ crosses an edge and is the only drawing of an edge incident to $w$ that does so. If $x$ is in a region of $w$, $w$ is separated from $N_{H^*}(v_f) \setminus P_i$ by the edges from $v$ to the outermost vertices of the regions that $P_i$ connects and hence can have no region outside of $P_i$. There are at most $|V_{\text{add}}|$ many such $w$.

Thus we find at most $|E_{\text{add}}^H| + |V_{\text{add}}| + 2|V_{\text{add}}|$ such $x$ on $P$ in total and thus $\ell \leq |E_{\text{add}}^H| + |V_{\text{add}}| + 2|V_{\text{add}}| \leq 3\kappa$. This concludes the proof. ◀

The underlying intuition one should keep about regions and difficult vertices is that a solution $\mathcal{G}$ partitions the shadow vertices into those which **(a)** have no edges routed through them, **(b)** have precisely one edge routed through them (in which case they must be part of the respective region), and **(c)** have at least two edges routed through them (in which case they form a difficult vertex).

In the remainder of this section we give some intuition on how the techniques used to prove Theorem 1 need to be extended to obtain Theorem 3. Unlike in the proof of Theorem 1 the number of vertices in $V_{\text{inc}}$ is not bounded by $\kappa$. Consequently, we cannot track their exact placement through patterns as defined in Definition 1. To circumvent this, we define *extended patterns* that store the necessary details about the cyclic orders in which individual regions, difficult vertices together with crossings, and endpoints of edges in $E_{\text{add}}^H$ as well as the single edge per vertex in $V_{\text{add}}$ that is allowed to cross, are supposed to appear inside a face. Then, as for patterns, we define how an extended pattern is *derived* from potential solutions.

The proof then proceeds by following the strategy laid down in Subsection 4.1. In particular, we define a notion of validity along with *pattern graphs* for extended patterns (cf. Definition 3). We show that validity can be checked and pattern graphs can be constructed efficiently (cf. Lemma 4). We use analogues to Lemma 5 and Lemma 7 to prune our instances so that they have bounded treewidth. At this point, we have all the ingredients we need to prove our final Theorem 3 – the main complication here compared to the proof of Theorem 1 is that we cannot have explicit labels identifying individual vertices in $V_{\text{inc}}$.

▶ **Theorem 3.** *IC-Planar Drawing Extension is* FPT *parameterized by* $\kappa$.

## 6 Inserting Two Vertices into a 1-Plane Drawing

In this section we show that 1-Planar Drawing Extension is polynomial-time tractable in the case where we are only adding 2 vertices to the graph along with their incident edges (i.e., when $|V_{\text{add}}| = 2$ and $E^H_{\text{add}} = \emptyset$[5]). Already solving this, at first glance simple, case seems to require non-trivial insight into the problem. In the following we call the two vertices in $V_{\text{add}}$ the *red* and *blue* vertex, denoted by $r$ and $b$, respectively.

On a high level, our algorithm employs a "delimit-and-sweep" approach. First, it employs exhaustive branching to place the vertices and identify a so-called "initial delimiter" – a Jordan curve that isolates a part of our instance that we need to focus on. In the second step, it uses such an initial delimiter to solve the instance via a careful dynamic programming subroutine. As our very first step, we exhaustively branch to determine which cells $r$ and $b$ should be drawn in, in $\mathcal{O}(n^2)$ time, and in each branch we add $r$ and $b$ into the selected cell(s) (from now on, we consider these embeddings part of $\mathcal{H}$).

**The Flow Subroutine.** Throughout this section, we will employ a generic network-flow subroutine that allows us to immediately solve certain restricted instances of 1-Planar Drawing Extension. In particular, assuming we are in the setting where $r$ and $b$ have already been inserted into $\mathcal{H}$, consider the situation where:

- There is a partial mapping $\lambda$ from the faces of $\mathcal{H}^\times$ to $\{R, B\}$; and
- $r$ and $b$ are in different cells of $\mathcal{H}$.

We say a 1-planar extension of $\mathcal{H}$ to $G$ is $\lambda$-*consistent* if the drawing of any edge in $E(G) \setminus E(H)$ which is incident to $r$ intersects the interior of face $F$ of $\mathcal{H}^\times$ only if $\lambda(F) = R$, and correspondingly the drawing of any edge in $E(G) \setminus E(H)$ which is incident to $b$ intersects the interior of face $F$ of $\mathcal{H}^\times$ only if $\lambda(F) = B$ (i.e., $\lambda$ specifies precisely which kind of edges may enter which face). We use a reduction to network flows to show:

▶ **Lemma 12.** *Given $\lambda$ as above, it is possible to determine whether there exists a $\lambda$-consistent 1-planar extension of $\mathcal{H}$ to $G$ in polynomial time.*

**Proof Sketch.** Consider the max flow instance $\theta_1$ constructed as follows. $\theta_1$ contains a universal sink $t$ and a universal source $s$. We add one vertex for each vertex in $N_{E(G)\setminus E(H)}(r)$, and a capacity-1 edge from each such "$R$-vertex" to $t$. We add one "$f$-vertex" for each face $f$ in $\mathcal{H}^\times$ that $\lambda$ maps to $R$, and a capacity-1 edge from each such vertex to every $R$-vertex that lies on the boundary of $f$. We add an (unlimited-capacity) edge from $s$ to every $f$-vertex whose face contains $r$ (possibly on its boundary). Finally, we add an edge from every $f$-vertex whose face contains $r$ to each other $f'$-vertex of capacity equal to the number of crossable edges that lie on the shared boundary of $f$ and $f'$. The instance $\theta_2$ is constructed in an analogous fashion for $B$ and $b$. To conclude the proof, it suffices to show that the drawings of the edges in $E_{\text{add}}$ incident to $r$ in a $\lambda$-consistent extension $\mathcal{G}$ of $\mathcal{H}$ to $G$ correspond to $s$-$t$-flows of values $|N_{E(G)\setminus E(H)}(r)|$ and $|N_{E(G)\setminus E(H)}(b)|$ for $\theta_1$ and $\theta_2$, respectively. ◀

[5] We note that it is trivial to extend the result to the case where the number of added edges is bounded by a fixed constant, via simple exhaustive branching.

▶ **Corollary 13.** *1-Planar Drawing Extension for $|V_{add}| = 1$ and $E^H_{add} = \emptyset$ can be solved in polynomial time.*

**Finding an Initial Delimiter.** We begin by formally defining the following notion:

▶ **Definition 14.** *A Jordan curve $\omega$ in the plane is an* initial delimiter *for $\mathcal{H}$ if:*
1. *$\omega$ passes through both $r$ and $b$ but through no other vertex of $\mathcal{H}$,*
2. *whenever $\omega$ shares at most one point with the interior of an edge, this point is a proper crossing between $\omega$ and that edge,*
3. *the intersection between $\mathcal{H}$ and the exterior of $\omega$ (including $\omega$ itself) contains a single cell $c_r$ whose boundary contains $r$, and a single cell $c_b$ whose boundary contains $b$, and*
4. *the intersection of the boundary of $c_r$ (resp. $c_b$) and $\omega$ is a single simple curve containing $r$ (resp. $b$) as an interior point.*

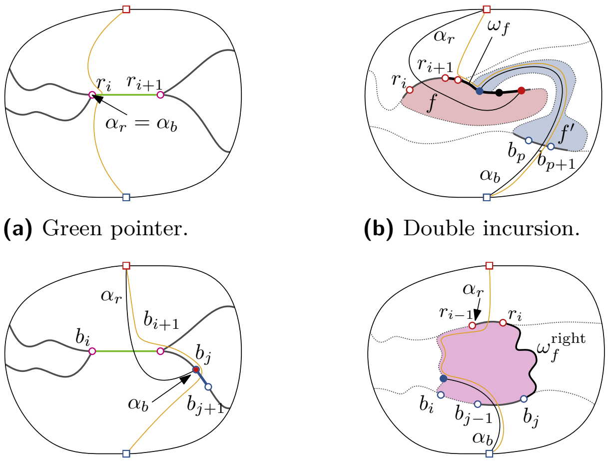
Intuitively, the third condition means that if we add $\omega$ onto $\mathcal{H}$, then there are unique cells in the exterior of $\omega$ for $r$ and $b$. A solution for our instance, i.e., a drawing $\mathcal{G}$ of $G$, is $\omega$-*compatible* if every edge from $E_{\text{add}}$ is drawn in the exterior region defined by $\omega$. We state the main result of this subsection below – intuitively, it provides us with a set of initial delimiters that we can exhaustively branch over, and in each branch we can restrict our attention to solutions that are compatible with the chosen initial delimiter. To avoid confusion, we note that the lemma covers the case where $r$ and $b$ are placed in the same cell $f$ (by branching on and placing a constant number of new edges that "separate" the boundary of $f$).

▶ **Lemma 15.** *For every instance $(G, H, \mathcal{H})$ where $|V_{add}| \leq 2$ and $E^H_{add} = \emptyset$, we can in polynomial time either solve $(G, H, \mathcal{H})$ or construct a set $Q$ of initial delimiters with the following property (or both): if $(G, H, \mathcal{H})$ admits a solution, then $Q$ contains at least one $\omega$ such that $(G, H, \mathcal{H})$ also admits an $\omega$-compatible solution.*

**Dynamic Programming.** We can now proceed with a very high-level sketch of how the algorithm proceeds once we have pre-selected (via branching) an initial delimiter. The general idea is to perform a left-to-right sweep of $\mathcal{H}$ by starting with the boundaries provided by the initial delimiter. The runtime of the algorithm is upper-bounded by the fact that it relies on dynamic programming where the maximum size of the records (representing possible "positions" on our sweep) is polynomial in the input size, and where the possibility of transitioning from one record to the next can be checked in polynomial time. In particular, the "steps" we use to move from one record to the next relies on a situational combination of exhaustive branching and the network-flow subroutine described in Lemma 12.

The intuitive reason we need to combine both of these techniques is that in some parts of our sweep, we will encounter faces where edges from both $r$ and $b$ may enter – there the interactions between these edges are too complicated to be modeled as a simple flow problem, but (as we will show) we can identify separating curves that cut our instance into parts for which we have a mapping $\lambda$ that can be applied in conjunction with Lemma 12.

We now formalize the records used by our algorithm: a *record* is a tuple $(\alpha_r, \alpha_b, T)$, where $\alpha_r$ is either a vertex in $\mathcal{H}$ or an edge $e \in E_{\text{add}}$ incident to $r$. $\alpha_b$ is defined symmetrically, and we describe such edges by specifying their endpoint and potential crossing point. $T$ is then an auxiliary element that specifies the "type" of a given record. Finally, we associate each record with a *delimiter* that iteratively pushes our "left" initial delimiter boundary towards the "right" one; everything to the left of the *delimiter* can be ignored, since the assumptions




**(a)** Green pointer. **(b)** Double incursion. **(c)** Left incursion.

**(d)** Right incursion. **(e)** Slice.

**Figure 3** The five types of records for the dynamic program. The computed delimiters are the orange curves. $r$ and $b$ are the rectangular vertices, colored disks are in $R$ and $B$, white disks are $r_i$ and $b_i$ depending on their border-color.

we use to select our records guarantee that it cannot be crossed by an edge of the targeted solution.

While we are forced to omit the majority of the details of the algorithm due to space constraints, below we at least provide a brief, intuitive summary of the 5 types of records used (see also Figure 3):

1. **Green Pointer.** $\alpha_r = \alpha_b$ is a vertex incident to an edge on the boundary of $c_r$ and $c_b$.
2. **Double Incursion.** One edge "covers" a part of one face that is accessed by the other edge from the other side.
3. **Left Incursion.** $\alpha_r$ (or $\alpha_b$) crosses into $c_b$ (or $c_a$) and heads "left".
4. **Right Incursion.** $\alpha_r$ (or $\alpha_b$) crosses into $c_b$ (or $c_a$) and heads "right".
5. **Slice.** One or both edges cross into a face other than $c_r$ and $c_b$.

Altogether, by using these records and carefully analyzing the cases that allow us to transition from one record to the other, we obtain a proof of:

▶ **Theorem 4.** *1-Planar Drawing Extension is polynomial-time tractable if $\kappa \leq 2$.*

## 7 Concluding Remarks

In this paper, we initiated the study of the problem of extending partial 1-planar and IC-planar drawings by providing several parameterized algorithms that target cases where only a few edges and/or vertices are missing from the graph. Our results follow up on previous seminal work on extending planar drawings, but the techniques introduced and used here are fundamentally different [2]. The by far most prominent question left open in our work concerns the (not only parameterized, but also classical) complexity of 1-Planar Extension w.r.t. $\kappa$. In particular, can one show that the problem is, at least, polynomial-time tractable for fixed values of $\kappa$? While the results presented in Section 6 are a promising start in this direction, it seems that new ideas are needed to push beyond the two-vertex case.

Follow-up work may also focus on extending other types of beyond planar drawings [17].

## References


1 Michael Albertson. Chromatic number, independence ratio, and crossing number. *ARS MATHEMATICA CONTEMPORANEA*, 1(1):1–6, 2008. doi:10.26493/1855-3974.10.2d0.

2 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.

3 Alan Arroyo, Martin Derka, and Irene Parada. Extending simple drawings. In Daniel Archambault and Csaba D. Tóth, editors, *27th International Symposium on Graph Drawing and Network Visualization (GD'19)*, volume 11904 of *LNCS*, pages 230–243. Springer, 2019. doi:10.1007/978-3-030-35802-0_18.

4 Alan Arroyo, Fabian Klute, Irene Parada, Raimund Seidel, Birgit Vogtenhuber, and Tilo Wiedera. Extending simple drawings with one edge is hard. *CoRR*, abs/1909.07347, 2019. Accepted for presentation at WG'20. arXiv:1909.07347.

5 Franz J. Brandenburg. Recognizing IC-planar and NIC-planar graphs. *J. Graph Algorithms Appl.*, 22(2):239–271, 2018. doi:10.7155/jgaa.00466.

6 Franz J. Brandenburg, Walter Didimo, William S. Evans, Philipp Kindermann, Giuseppe Liotta, and Fabrizio Montecchiani. Recognizing and drawing IC-planar graphs. *Theor. Comput. Sci.*, 636:1–16, 2016. doi:10.1016/j.tcs.2016.04.026.

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

8 Sergio Cabello and Bojan Mohar. Adding one edge to planar graphs makes crossing number and 1-planarity hard. *SIAM J. Comput.*, 42(5):1803–1829, 2013. doi:10.1137/120872310.

9 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.

10 Timothy M. Chan, Fabrizio Frati, Carsten Gutwenger, Anna Lubiw, Petra Mutzel, and Marcus Schaefer. Drawing partially embedded and simultaneously planar graphs. *J. Graph Algorithms Appl.*, 19(2):681–706, 2015. doi:10.7155/jgaa.00375.

11 Steven Chaplick, Paul Dorbec, Jan Kratochvíl, Mickaël Montassier, and Juraj Stacho. Contact representations of planar graphs: Extending a partial representation is hard. In Dieter Kratsch and Ioan Todinca, editors, *40th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'14)*, volume 8747 of *LNCS*, pages 139–151. Springer, 2014. doi:10.1007/978-3-319-12340-0_12.

12 Steven Chaplick, Radoslav Fulek, and Pavel Klavík. Extending partial representations of circle graphs. In Stephen K. Wismath and Alexander Wolff, editors, *21st International Symposium on Graph Drawing (GD'13)*, volume 8242 of *LNCS*, pages 131–142. Springer, 2013. doi:10.1007/978-3-319-03841-4_12.

13 Steven Chaplick, Grzegorz Guspiel, Grzegorz Gutowski, Tomasz Krawczyk, and Giuseppe Liotta. The partial visibility representation extension problem. *Algorithmica*, 80(8):2286–2323, 2018. doi:10.1007/s00453-017-0322-4.

14 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

15 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.

16 Giordano Da Lozzo, Giuseppe Di Battista, and Fabrizio Frati. Extending upward planar graph drawings. In Zachary Friggstad, Jörg-Rüdiger Sack, and Mohammad R. Salavatipour, editors, *16th International Symposium on Algorithms and Data Structures (WADS'19)*, volume 11646 of *LNCS*, pages 339–352. Springer, 2019. doi:10.1007/978-3-030-24766-9_25.

17 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Comput. Surv.*, 52(1):4:1–4:37, 2019. doi:10.1145/3301281.

18 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

19 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

20 Alexander Grigoriev and Hans L. Bodlaender. Algorithms for graphs embeddable with few crossings per edge. *Algorithmica*, 49(1):1–11, 2007. doi:10.1007/s00453-007-0010-x.

21 Martin Grohe. Computing crossing numbers in quadratic time. *Journal of Computer and System Sciences*, 68(2):285–302, 2004. Special Issue on STOC 2001. doi:10.1016/j.jcss.2003.07.008.

22 Seok-Hee Hong and Hiroshi Nagamochi. Convex drawings of graphs with non-convex boundary constraints. *Discret. Appl. Math.*, 156(12):2368–2380, 2008.

23 Pavel Klavík, Jan Kratochvíl, Tomasz Krawczyk, and Bartosz Walczak. Extending partial representations of function graphs and permutation graphs. In Leah Epstein and Paolo Ferragina, editors, *20th Annual European Symposium on Algorithms (ESA'12)*, volume 7501 of *LNCS*, pages 671–682. Springer, 2012. doi:10.1007/978-3-642-33090-2_58.

24 Pavel Klavík, Jan Kratochvíl, Yota Otachi, Ignaz Rutter, Toshiki Saitoh, Maria Saumell, and Tomás Vyskocil. Extending partial representations of proper and unit interval graphs. *Algorithmica*, 77(4):1071–1104, 2017. doi:10.1007/s00453-016-0133-z.

25 Pavel Klavík, Jan Kratochvíl, Yota Otachi, and Toshiki Saitoh. Extending partial representations of subclasses of chordal graphs. *Theor. Comput. Sci.*, 576:85–101, 2015. doi:10.1016/j.tcs.2015.02.007.

26 Pavel Klavík, Jan Kratochvíl, Yota Otachi, Toshiki Saitoh, and Tomás Vyskocil. Extending partial representations of interval graphs. *Algorithmica*, 78(3):945–967, 2017. doi:10.1007/s00453-016-0186-z.

27 Stephen G. Kobourov, Giuseppe Liotta, and Fabrizio Montecchiani. An annotated bibliography on 1-planarity. *Computer Science Review*, 25:49–67, 2017. doi:10.1016/j.cosrev.2017.06.002.

28 Vladimir P. Korzhik and Bojan Mohar. Minimal obstructions for 1-immersions and hardness of 1-planarity testing. *Journal of Graph Theory*, 72(1):30–71, 2013. doi:10.1002/jgt.21630.

29 Giuseppe Liotta and Fabrizio Montecchiani. L-visibility drawings of IC-planar graphs. *Inf. Process. Lett.*, 116(3):217–222, 2016. doi:10.1016/j.ipl.2015.11.011.

30 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.

31 Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyama. Layout adjustment and the mental map. *J. Visual Languages and Computing*, 6(2):183–210, 1995. doi:10.1006/jvlc.1995.1010.

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

33 Gerhard Ringel. Ein Sechsfarbenproblem auf der Kugel. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 29(1):107–117, 1965. doi:10.1007/BF02996313.

34 Neil Robertson and Paul D. Seymour. Graph minors. III. planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984. doi:10.1016/0095-8956(84)90013-3.

35 Wanshun Yang, Weifan Wang, and Yiqiao Wang. Acyclic coloring of IC-planar graphs. *Discret. Math.*, 342(12), 2019. doi:10.1016/j.disc.2019.111623.