On k-Plane Insertion into Plane Drawings

Julia Katheder[1], Philipp Kindermann[2], Fabian Klute[3], Irene Parada[3], and Ignaz Rutter[4]

1 **Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany**
`julia.katheder@uni-tuebingen.de`

2 **FB IV – Informatikwissenschaften, Universität Trier, Germany**
`kindermann@uni-trier.de`

3 **Universitat Politècnica de Catalunya, Spain**
`fabian.klute@upc.edu`, `irene.parada@upc.edu`

4 **Faculty of Computer Science and Mathematics, University of Passau, Germany**
`rutter@fim.uni-passau.de`

Abstract

We introduce the k-Plane Insertion into Plane drawing (k-PIP) problem: given a plane drawing of a planar graph G and a set of edges F, insert the edges in F into the drawing such that the resulting drawing is k-plane. In this paper, we focus on the 1-PIP scenario. We present a linear-time algorithm for the case that G is a triangulation, while proving NP-completeness for the case that G is biconnected and F forms a path or a matching.

1 Introduction

Inserting edges into planar graphs is a long-studied problem in Graph Drawing. Most commonly, the goal is to find a way to insert the edges while minimizing the number of crossings and maintaining the planarity of the prescribed subgraph. This problem is a core step in the planarization method to find graph drawings with few crossings [22]. Gutwenger et al. [15] have studied the case of a single edge. For multiple edges the picture is more complicated. In case the edges are all incident to one vertex previously not present in the graph, the problem can again be solved in polynomial time [7]. However, the problem is NP-hard even when the given drawing is fixed and its corresponding graph biconnected [23, 25]. Assuming a fixed drawing, Hamm and Hliněný presented an FPT-algorithm parameterized by the number of crossings [16]. Finally, Chimani and Hliněný [8] gave an FPT-algorithm for the fixed and variable embedding settings with the number of inserted edges as a parameter.

In this paper, we take a slightly different viewpoint and do not restrict the overall number of created crossings, but instead their structure. Moreover, we focus on the case when the drawing of the given planar graph is fixed. Then our goal is, given a plane drawing Γ and a set F of edges with its endpoints being vertices of this graph, to find a k-plane drawing containing Γ as a subdrawing plus the edges of F. Here, a *k-plane drawing* of a graph is one in which no edge is crossed more than k times. The class of *k-planar graphs*, which are those admitting a k-plane drawing, is widely studied in Graph Drawing [10, 17].

▶ Problem 1 (k-Plane Insertion into Plane drawing (k-PIP)). Given a plane drawing Γ of a graph $G = (V, E)$ and a set of edges F with endpoints in V, find a k-plane drawing of the graph $(V, E \cup F)$ that contains Γ as a subdrawing.

Our results. In this paper, we focus on 1-PIP. Note that we may assume that a solution to an instance of 1-PIP is a simple 1-plane drawing, i.e., no two edges share more than one point, since in a 1-plane drawing simplicity only affects the crossings of adjacent edges.

40th European Workshop on Computational Geometry, Ioannina, Greece, March 13–15, 2024.
This is an extended abstract of a presentation given at EuroCG'24. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

In Section 2, we present an $O(|V|)$ algorithm for the case that G is a triangulation. To accomplish this, we first reduce the number of possible ways one edge can be inserted into the given drawing to at most two per edge in F and then use a 2-SAT formulation to compute a solution if possible. In Section 3, we show that 1-PIP is NP-complete even if G is biconnected and the edges in F form a path or a matching.

Related work. k-PIP is related to the problem of extending a partial drawing of a graph to a drawing of the full graph. Usually, the goal in such problems is to maintain certain properties of the given drawing. For example, in works by Angelini et al. [1], Eiben et al. [13, 12], Ganian et al. [14], or Arroyo et al. [3] the input is a plane, 1-plane, k-plane, or simple drawing, respectively, and the desired extension must maintain the property of being plane, 1-plane, k-plane, or simple. Restrictions of the drawing such as it being straight-line [24], level-planar [4], upward [20], or orthogonal [2] have been explored. Other results consider the number of bends [6] or assume that the partially drawn subgraph is a cycle [5, 21].

2 Extending a triangulation

We assume standard notation and concepts from graph theory; compare, e.g., [11]. Given an instance (G, Γ, F) of 1-PIP, an edge $e \in F$ might be inserted into Γ in different ways. Note that e cannot be inserted without crossings in a triangulation. An *option* for e is an edge γ of G such that e can be inserted into Γ crossing only γ. Note that in a triangulation, a pair of faces uniquely defines an edge γ that must be crossed if e is inserted into said pair of faces. Thus, with the term option we might also refer to a pair of faces. An option for e is *safe* if, in case the instance admits a solution, there is one solution in which e is inserted according to this option. Two options for two edges e and e' of F *clash* if inserting e and e' according to these options violates 1-planarity. Examples of safe options are those of edges with a single option and an option without clashes. An immediate solution can be found if each edge in F has a non-clashing option. However, it is not sufficient for each edge in F to have a safe option in order to find a solution, e.g. in the case that two single options are clashing. Finally, observe that in a triangulation, each edge of Γ can only be an option for one edge of F and clashes with at most four other options.

▶ **Theorem 2.1.** *Given a plane drawing Γ of a triangulation $G = (V, E)$ and a set of edges F with endpoints in V, 1-PIP on (G, Γ, F) can be solved in time $O(|V|)$.*

Proof. The idea is to preprocess the instance until we are left with a set of edges $F' \subseteq F$ with two options each. The resulting instance can then be solved using a 2SAT formula. Begin by computing all options for every $e \in F$. Since Γ is a plane drawing, we can get the triangles incident to each $v \in V$ in cyclic order and also the options for edges in F incident to v. This way, we get the overall $O(|V|)$ options for edges in F in $O(|V|)$ time. For an edge $(u, v) \in F$, $u, v \in V$, with two or more options we say that two options are *consecutive* if the corresponding faces are consecutive in the cyclic order around u (or v); see the options for (u, v) in Fig. 1(c) for an illustration. We say a set of options is *cyclically consecutive* if the corresponding edges induce a cycle in G; see the options for (u, v) in Fig. 1(d).

Whenever an edge e has no option left, we stop and output `no` and if e has exactly one option left, we insert it into Γ. Every time we insert an edge, we need to remove at most four options of other edges plus all the options of the just inserted edge. Consider an edge $e = (u, v) \in F$, $u, v \in V$, that has three or more options. We consider four cases.

(a) There are at least three options for e such that no two of them are consecutive. Given one of these three options, say σ_i, we claim that it is either safe or never possible in a

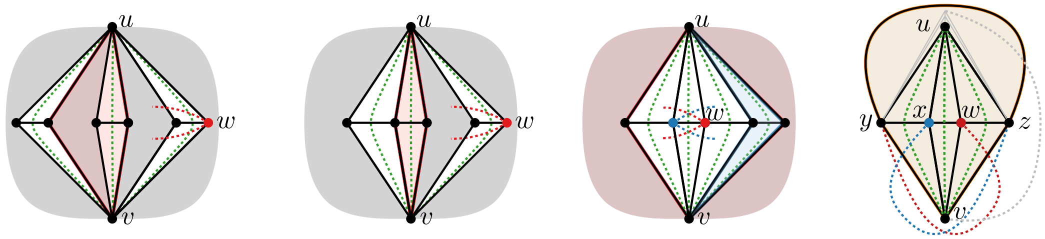

Figure 1 Cases with three or more options in a triangulation.

solution; see Fig. 1(a). If σ_i is not clashing with any other option, it is safe and we add it. Otherwise, let w and x be the two endpoints of σ_i. Option σ_i can only be clashing with two options for edges in F incident to w and two options for edges in F incident to x. Moreover, any option for those edges clashes with σ_i. To see this, consider w.l.o.g. such an edge in F incident to w (illustrated in red in Fig. 1(a)). The other options for e define a non-empty region in which this edge cannot be drawn (see red region in Fig. 1(a)), restricting its options to options that clash with σ_i.

(b) There are at least three options for e, and one of them, σ_i, is not consecutive to any of the other two; see Fig. 1(b). As in Case (a), σ_i is either safe or never possible.

(c) There are at least four consecutive non-cyclic options for e; see Fig. 1(c). Let σ_i be one of the central options. As in the previous cases, σ_i is either safe or never possible.

(d) There are three consecutive or four cyclically consecutive options for e; see Fig. 1(d). Consider the middle option σ_i (or any option if there were four). If it is safe, we just add it. Else, let w and x be the endpoints of σ_i and y, z the other endpoints of options for e. Assume, w.l.o.g., that σ_i clashes with an option of an edge e_w incident to w and to vertex y. For σ_i to be a possible option in a solution, e_w must have an option that does not clash with it. There is only one possibility, and it implies that v, y, z or u, y, z form a triangle. Assume, w.l.o.g., the former, so (y, z) is an edge in Γ. Let $\diamond$ be the set of vertices $\{u, v, w, x, y, z\}$ and $G_\diamond$ the octahedron subgraph of G induced by $\diamond$.
Edges in F with exactly one endpoint in $\diamond \setminus \{u\}$ have at most one option. Thus, we can insert them first and see whether we are still in Case (d). Edges incident to u and to a vertex not in $\diamond$ cannot clash with any option of an edge between vertices in $\diamond$. Thus, we can solve the constant-size subinstance consisting of inserting such edges into $G_\diamond$ independently, taking into account the single-option edges that we might have inserted.

Once each edge has exactly two options we create a 2SAT formula containing one variable per option and clauses that ensure exactly one option per edge in F' and exclude clashes. This formula has size $O(|V|)$ and is satisfiable iff the original instance has a solution. ◀

3 Inserting a path or a matching is NP-complete

We prove NP-hardness by reduction from Planar Monotone 3-SAT; the membership of the problem in NP is straightforward. Let ϕ be a Boolean formula in CNF with variables $V = \{x_1, \ldots, x_n\}$ and clauses $C = \{c_1, \ldots, c_m\}$. Each clause has at most three literals and is either *positive* (all literals are positive) or *negative* (all literals are negative). Furthermore, there is a rectilinear representation Γ_ϕ of the variable-clause incidence graph of ϕ, such that all variables and clauses are depicted as axis-aligned rectangles connected via vertical segments and all variables are positioned on the x-axis, all positive clauses lie above and all negative clauses lie below the x-axis; see Fig. 2 for an example. This problem is known to be NP-complete [9, 19].

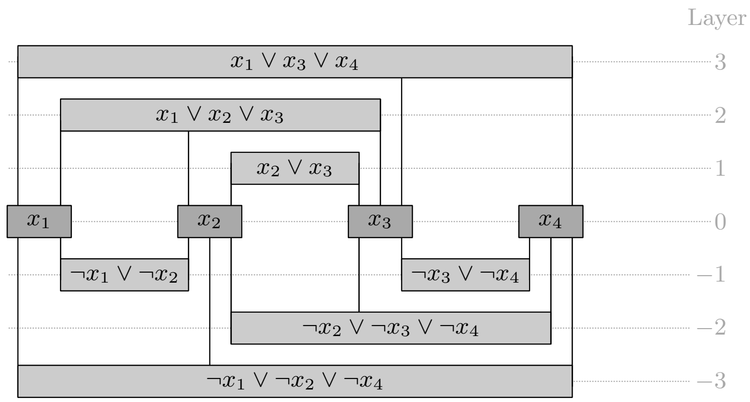

Figure 2 Rectilinear representation of the variable-clause incidence graph of a PLANAR MONOTONE 3-SAT instance.

In the following, starting from Γ_ϕ, we construct a graph $G = (V, E)$, its plane drawing Γ, and the edge set F, which will be inserted into Γ in a specific way. We start with the case of F forming a path (see Theorem 3.4) and describe the changes to our construction for F being a matching afterwards (see Corollary 3.5). The bars in Γ_ϕ can be layered decreasingly from top to bottom. We set the layer of the variables as layer zero and denote by $L(c)$ the layer of clause c; see Fig. 2. We denote by H the graph K_4 missing one arbitrary edge and create chains of copies of H, that are connected via their degree 2 vertices, such that the chord is drawn inside the C_4 cycle of H. This provides the structure for routing the path F in the drawing. We say that an edge $e \in F$ is *ℓ-spanning* if there are ℓ chords of H in the chain between its endpoints.

The variable gadget. We replace each bar of a variable x in Γ_ϕ by a variable gadget which consists of a H-chain of $4a + 1$ copies where a is the maximum over the number of positive and negative occurrences of x in ϕ. Let $u_1, \dots, u_{4a+2}$ be the vertices where the H copies are joined from left- to rightmost copy. Moreover, we mark for $i \in \{0, \dots, a-1\}$ the vertices u_{3+4i} as *variable endpoints* (c.f. the squares in Fig. 3). Each such vertex gets two additional incident edges, called *literal edges*, which are going to connect the variable gadgets to adjacent layers and encode the truth-value of the respective variables. We call a literal edge exiting its variable endpoint upwards (downwards) *positive* (*negative*).

For each $i \in \{0, \dots a-1\}$, we define the path F to pass through u_{1+4i}, u_{4+4i}, u_{3+4i}, u_{6+4i}, u_{5+4i}, except for $a-1$ where we omit the last vertex; cf. red path in Fig. 3. Then, for a variable gadget its edges in F consist of an alternation of 3-spanning and 1-spanning edges. By 1-planarity and simplicity, F cannot cross the H-chain. We refer to the 3-spanning edges that cross literal edges as *blocking 3-spanners* and the others as *forcing 3-spanners*.

For the remainder, we depict literal edges representing the value `true` in blue and the ones representing `false` in orange, while the edges in F are colored in red; c.f. Figs. 3 to 6. The proofs of statements marked with a ($\star$) are available in the long version on arXiv [18].

▶ **Lemma 3.1** ($\star$). *Let v be a variable gadget described as above, then in any 1-planar drawing containing it, its literal edges, and the edges $F_v \subseteq F$ incident to vertices in v either all negative or all positive literal edges are crossed.*

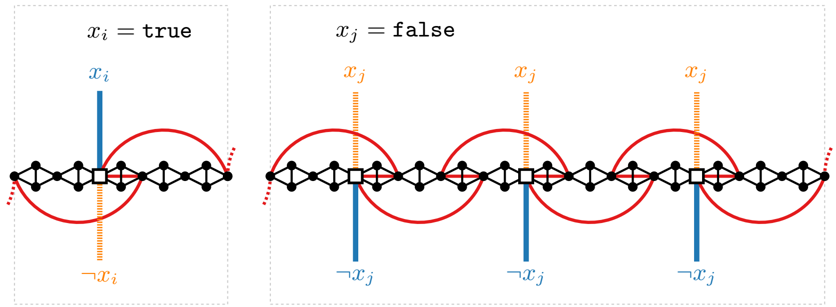

Figure 3 Drawing of the variable gadget.

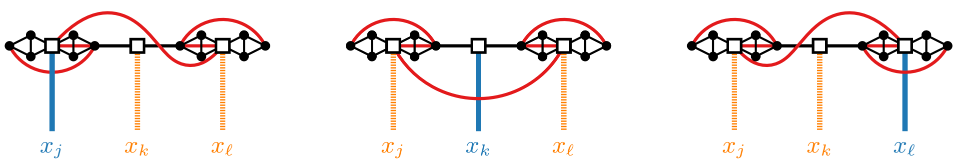

Figure 4 Drawing of the clause gadget.

If the negative literal edges are crossed, we think of the variable corresponding to the gadget as set to `true` and `false` otherwise. We connect the variable gadgets by adding one copy of H with a 1-spanning edge added to F in between them; see Fig. 5.

The clause gadget. We describe the construction only for the positive clauses, as the construction for the negative ones is symmetric. The clause gadget is depicted in Fig. 4. It consists of a chain of two copies of H, followed by two edges, followed by two more copies of H. We mark the middle vertices of each of the two copies and the two edges as *variable endpoints* and add one additional edge to them, their *literal edge*. Assume that all literal edges are drawn on the same side as shown in Fig. 4 and add edges to F as shown in red.

▶ **Lemma 3.2** ($\star$). *Let c be a clause gadget drawn as describe above, then in any 1-planar drawing containing c, its literal edges, and the edges $F_c \subseteq F$ incident to vertices of c, at least one literal edge has to be crossed by an edge in F_c.*

Propagating the variable state. In order to ensure that there is no interaction between the path F and other parts of the drawing, we insert H-chains with 1-spanning edges added to F on every layer of Γ_ϕ and insert the clause gadgets into the respective layers as shown in Fig. 5. We create further variable endpoints on the layers $i > 0$, in order to propagate the state of the variable gadgets to clauses in higher layers. For each pair of corresponding variable endpoints of a variable gadget and clause gadget, we create a variable endpoint at a merged vertex in the H-chain in each layer i with $0 < i < L(c)$ and prescribe F to span the two neighboring copies of H. Further, we connect every two consecutive variable endpoints on layer j and $j+1$ with $0 \leq j < L(c)$ via a literal edge, as illustrated in Fig. 5.

▶ **Lemma 3.3** ($\star$). *Let $P = e_1, \ldots, e_{L(c)}$ be a path of literal edges such that e_1 is incident to a variable endpoint of variable gadget v and $e_{L(c)}$ to one clause gadget c, if e_1 is crossed in v, then $e_{L(c)}$ is crossed by an edge of F incident to vertices on layer $L(c) - 1$.*

Note that if the first edge of P is not crossed in the variable gadget, this does not imply that its last edge is uncrossed in layer $L(c) - 1$. In fact, this is possible when multiple literals

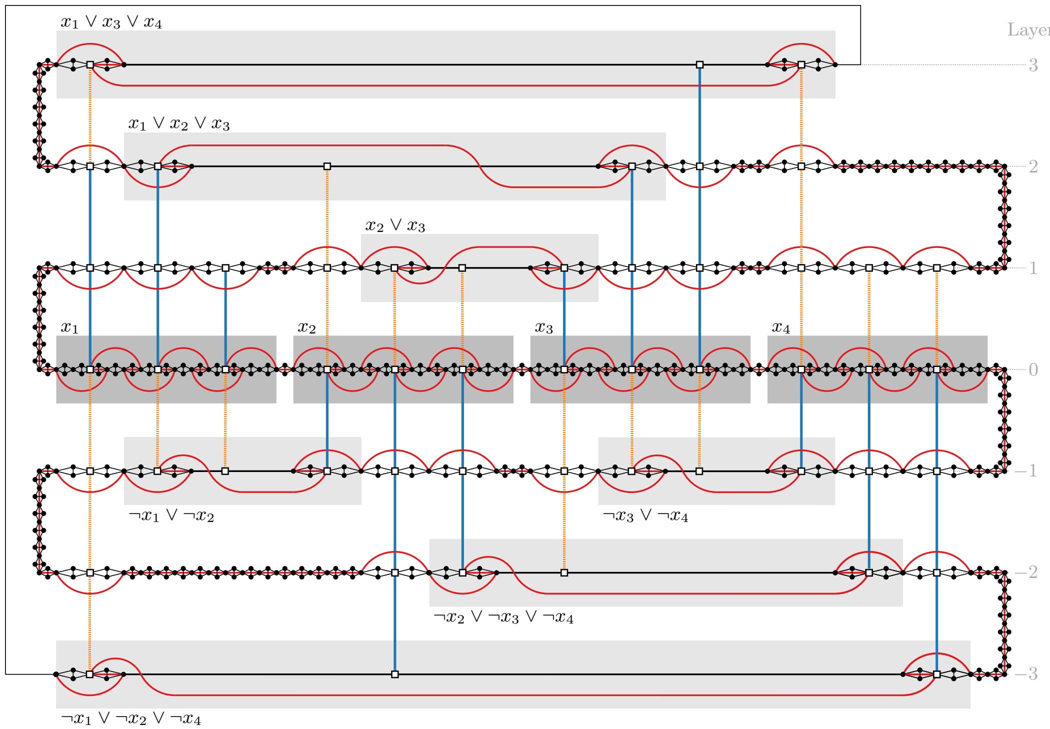

Figure 5 Solution (in red) of the 1-PIP instance coming from the graph given in Fig. 2.

evaluate to `true` for a clause gadget; e.g. the top- and leftmost orange edge in Fig. 5. If a variable is not contained in the same number of negative and positive clauses we ensure the alternating pattern of F on layer zero by connecting the remaining variable endpoints to ones on layer one, e.g., see x_1 in Fig. 5. Lastly, the subpath of F on every layer is joined to the one of the previous layer in an alternating fashion by an H-chain and 1-spanning edges.

▶ **Theorem 3.4** ($\star$). *1-PIP is* NP*-complete even if G is biconnected and F forms a path.*

We can use the same construction but replacing the alternating connections between the layers by edges to prove NP-hardness also for the case that F is a matching; see Fig. 6.

▶ **Corollary 3.5.** *1-PIP is* NP*-complete even if G is biconnected and F forms a matching.*

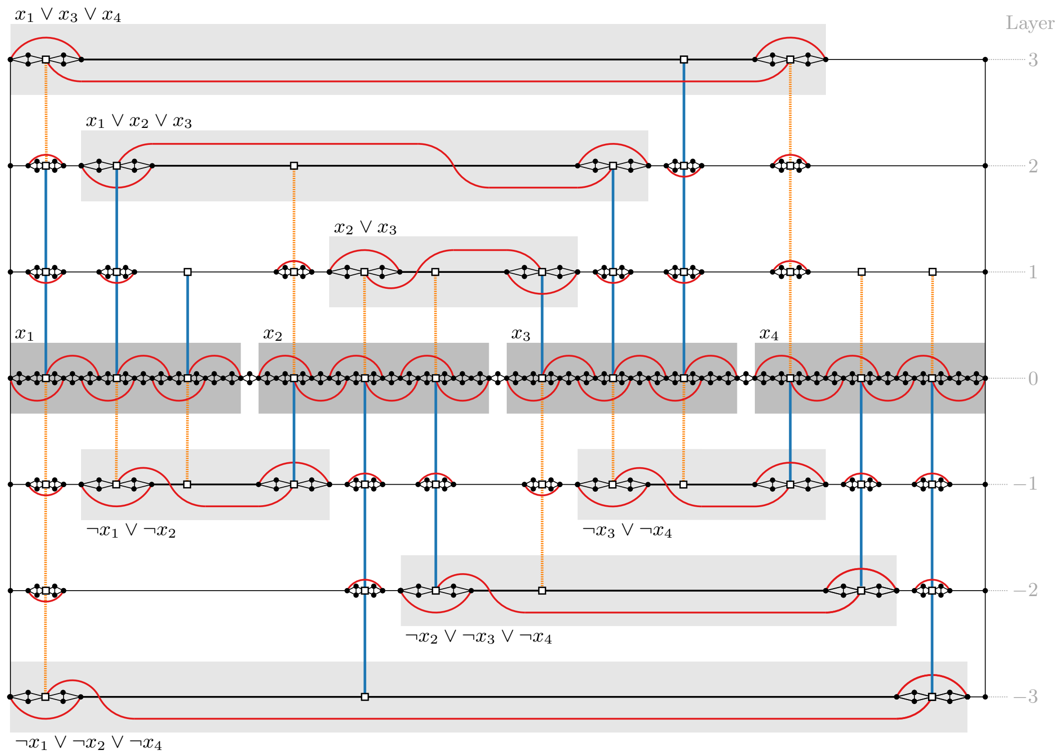

Figure 6 1-PIP instance where F is a matching reduced from the graph given in Fig. 2

4 Conclusion

We introduced the k-PIP problem and showed that it is NP-complete even when the given graph is biconnected and the inserted edges form a path or matching. We also presented a linear-time algorithm when the given graph is triangulated. This naturally raises the question if the triconnected case of k-PIP is polynomial-time solvable or NP-complete.

References

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

2 Patrizio Angelini, Ignaz Rutter, and Sandhya T. P. Extending partial orthogonal drawings. In *Proceedings of the 28th International Symposium on Graph Drawing and Network Visualization (GD'20)*, volume 12590 of *LNCS*, page 265–278. Springer, 2020. doi:10.1007/978-3-030-68766-3_21.

3 Alan Arroyo, Fabian Klute, Irene Parada, Birgit Vogtenhuber, Raimund Seidel, and Tilo Wiedera. Inserting one edge into a simple drawing is hard. *Discrete & Computational Geometry*, 69(3):745–770, 2023. doi:10.1007/S00454-022-00394-9.

4 Guido Brückner and Ignaz Rutter. Partial and constrained level planarity. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*, page 2000–2011. SIAM, 2017. doi:10.1137/1.9781611974782.130.

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

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

7 Markus Chimani, Carsten Gutwenger, Petra Mutzel, and Christian Wolf. Inserting a vertex into a planar graph. In Claire Mathieu, editor, *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA'09)*, page 375–383. SIAM, 2009. doi:10.1137/1.9781611973068.42.

8 Markus Chimani and Petr Hlinený. Inserting multiple edges into a planar graph. *Journal of Graph Algorithms and Applications*, 27(6):489–522, 2023. doi:10.7155/JGAA.00631.

9 Mark de Berg and Amirali Khosravi. Optimal binary space partitions for segments in the plane. *International Journal of Computational Geometry and Applications*, 22(3):187–206, 2012. doi:10.1142/S0218195912500045.

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

11 Reinhard Diestel. *Graph Theory, 5th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012. doi:10.1007/978-3-662-53622-3.

12 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending nearly complete 1-planar drawings in polynomial time. In *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science, (MFCS'20)*, volume 170 of *LIPIcs*, page 31:1–31:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.MFCS.2020.31.

13 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending partial 1-planar drawings. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP'20)*, volume 168 of *LIPIcs*, page 43:1–43:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.43.

14 Robert Ganian, Thekla Hamm, Fabian Klute, Irene Parada, and Birgit Vogtenhuber. Crossing-optimal extension of simple drawings. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, page 72:1–72:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.ICALP.2021.72.

15 Carsten Gutwenger, Petra Mutzel, and René Weiskircher. Inserting an edge into a planar graph. *Algorithmica*, 41(4):289–308, 2005. doi:10.1007/S00453-004-1128-8.

16 Thekla Hamm and Petr Hliněný. Parameterised partially-predrawn crossing number. In *Proceedings of the 38th International Symposium on Computational Geometry (SoCG'22)*, volume 224 of *LIPIcs*, page 46:1–46:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.SoCG.2022.46.

17 Seok-Hee Hong and Takeshi Tokuyama, editors. *Beyond Planar Graphs, Communications of NII Shonan Meetings*. Springer, 2020. doi:10.1007/978-981-15-6533-5.

18 Julia Katheder, Philipp Kindermann, Fabian Klute, Irene Parada, and Ignaz Rutter. On k-plane insertion into plane drawings. arXiv preprint, 2024. URL: https://arxiv.org/abs/2402.14552, arXiv:2402.14552.

19 Donald E. Knuth and Arvind Raghunathan. The problem of compatible representatives. *SIAM Journal on Discrete Mathematics*, 5(3):422–427, 1992. doi:10.1137/0405033.

20 Giordano Da Lozzo, Giuseppe Di Battista, and Fabrizio Frati. Extending upward planar graph drawings. *Computational Geometry: Theory and Applications*, 91:101668, 2020. doi:10.1016/j.comgeo.2020.101668.

21 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.

22 Petra Mutzel and Thomas Ziegler. The constrained crossing minimization problem. In Jan Kratochvíl, editor, *Proceedings of the 7th International Symposium on Graph Drawing (GD'99)*, volume 1731 of *Lecture Notes in Computer Science*, page 175–185. Springer, 1999. doi:10.1007/3-540-46648-7_18.

23 Petra Mutzel and Thomas Ziegler. The constrained crossing minimization problem a first approach. In *Proceedings of the 1999 International Conference on Operations Research*, page 125–134. Springer, 1999. doi:10.1007/978-3-642-58409-1_11.

24 Maurizio Patrignani. On extending a partial straight-line drawing. *International Journal of Foundations of Computer Science*, 17(5):1061–1070, 2006. doi:10.1142/S0129054106004261.

25 Thomas Ziegler. *Crossing minimization in automatic graph drawing*. PhD thesis, Saarland University, Saarbrücken, Germany, 2001.