On k-Plane Insertion into Plane Drawings

Julia Katheder
Universität Tübingen, Germany

Philipp Kindermann
Universität Trier, Germany

Fabian Klute
Universitat Politècnica de Catalunya, Barcelona, Spain

Irene Parada
Department of Mathematics, Universitat Politècnica de Catalunya, Barcelona, Spain

Ignaz Rutter
Universität Passau, Germany

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 F of edges, insert the edges in F into the drawing such that the resulting drawing is k-plane. In this paper, we show that the problem is NP-complete for every $k \geq 1$, even when G is biconnected and the set F of edges forms a matching or a path. On the positive side, we present a linear-time algorithm for the case that $k = 1$ and G is a triangulation.

2012 ACM Subject Classification Mathematics of computing → Combinatoric problems

Keywords and phrases Graph drawing, edge insertion, k-planarity

Digital Object Identifier 10.4230/LIPIcs.GD.2024.35

Related Version *Full Version*: `https://doi.org/10.48550/arXiv.2402.14552` [19]

Funding *Julia Katheder*: Funded by the Deutsche Forschungsgemeinschaft (DFG) – 364468267.
Fabian Klute: F. K. is supported by a “María Zambrano grant for attracting international talent” and by grant PID2019-104129GB-I00 funded by MICIU/AEI/10.13039/501100011033.
Irene Parada: I. P. is a Serra Húnter Fellow. Partially supported by grant 2021UPC-MS-67392 funded by the Spanish Ministry of Universities and the European Union (NextGenerationEU) and by grant PID2019-104129GB-I00 funded by MICIU/AEI/10.13039/501100011033.
Ignaz Rutter: Funded by the Deutsche Forschungsgemeinschaft (DFG) – 541433306.

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 [23]. Gutwenger et al. [16] 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 be solved in polynomial time [8]. However, the general problem is NP-hard even when the given drawing is fixed and the underlying graph is biconnected [24, 26]. Assuming a fixed drawing, Hamm and Hliněný presented an FPT-algorithm parameterized by the number of crossings [17]. Finally, Chimani and Hliněný [9] 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 G is fixed. Then our goal is, given a plane drawing Γ

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).
Editors: Stefan Felsner and Karsten Klein; Article No. 35; pp. 35:1–35:11
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of G and a set F of edges not present in G, 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 [11, 18].

▶ **Problem 1** (k-Plane Insertion into Plane drawing (k-PIP)). *Let Γ be a plane drawing of a graph $G = (V, E)$ and let $H = (V, E')$ be its complement. Given G, Γ and a set $F \subseteq E'$ of edges, find a k-plane drawing of the graph $(V, E \cup F)$ that contains Γ as a subdrawing.*

For any fixed $k \in \mathbb{N}$, an *instance* (G, Γ, F) of k-PIP consists of a graph G, a plane drawing Γ of G, and a set of edges F from the complement of G.

Our contribution. In addition to introducing this problem, we give two results. In Section 2, we present an $O(|V|)$ algorithm for 1-PIP 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 k-PIP is NP-complete for every $k \geq 1$ 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, 14], Ganian et al. [15], or Arroyo et al. [3, 4] 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 [25], level-planar [5], upward [21], or orthogonal [2] have been explored. Other results consider the number of bends [7] or assume that the partially drawn subgraph is a cycle [6, 22].

2 1-PIP: Efficiently inserting edges into a triangulation

We assume standard notation and concepts from graph theory; compare, e.g., [12]. Given an instance (G, Γ, F) of 1-PIP, often times an edge $e \in F$ can 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 adjacent faces uniquely defines an edge γ that must be crossed if e is inserted into said pair of faces. Thus, we also use the term option to refer to such a pair of

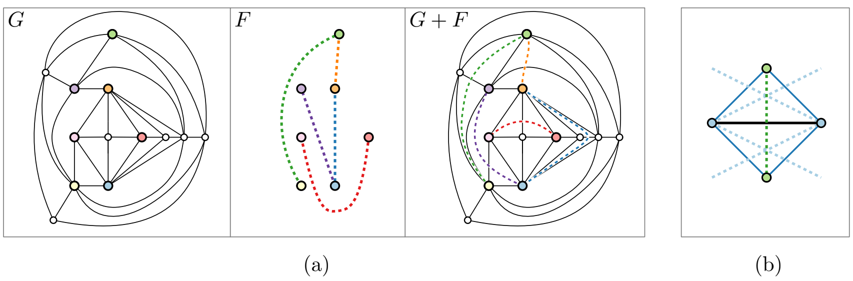

Figure 1 (a) The 1-PIP problem: a plane graph G, a set F of edges, and a 1-plane drawing of $G + F$. (b) In a triangulation, an edge in G (bold) can only be an option for a single edge in F (green) and clashes with at most four other options (blue).

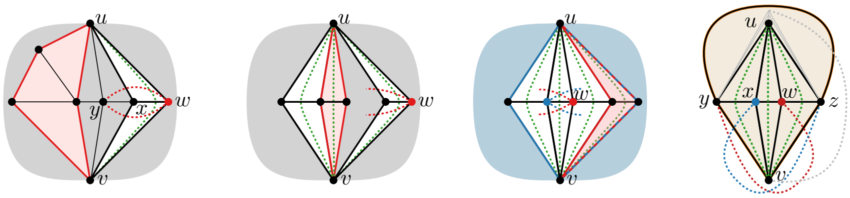

Figure 2 Cases with three or more options in a triangulation.

faces. An option for e is *safe* if, in case the instance admits a solution, there is a solution in which e is inserted according to this option. Two options for two edges e and e' of F *clash* if inserting both 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. 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; see Figure 1(b). Further, for a triangulation, we have the following property where a *blocking cycle* in the drawing forces an edge to have only clashing options; see Figure 2(a).

▶ **Property 2.** *Let $e = (u, v)$ be an edge in F and let $\sigma_i = (x, w)$ be one of its options. For an edge $e' = (w, y) \in F$ having at least one clashing option with σ_i, there is other non-clashing option with σ_i, if there is a cycle C in G such that $u, v \in C$ and $x, w, y \notin C$, .*

Proof. The cycle $C_{\sigma_i} = (u, x, v, w, u)$ and C share only the vertices u and v and since Γ is plane, C and C_{σ_i} partition Γ into four regions, where the edges in C and C_{σ_i} constitute the borders of said regions. The edge e' has an option that clashes with σ_i, i.e., this option is an edge in C_{σ_i}. Then the endpoint y of e' lies in the region bordered by edges in C and C_{σ_i}. By 1-planarity, e' cannot have an option not clashing with σ_i, as this would require crossing C twice. ◀

▶ **Theorem 3.** *1-PIP can be solved in linear time for instances (G, Γ, F) where G is a triangulation.*

Proof. The idea is to preprocess the instance until we are left with a set $F' \subseteq F$ of edges with two options each. The resulting instance can then be solved using a 2SAT formula. We begin by computing all options for every $e \in F$, resulting in $O(|V|)$ options, since each option is an edge in the plane drawing Γ, crossed by a unique edge in F. Since Γ is plane, 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. Hence, 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 Figure 2(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 Figure 2(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 three cases.

(a) There are at least three options for e, and at least one of them, σ_i, is not consecutive to any of the other two; see rightmost option in Figure 2(b). We claim that σ_i is either safe or never possible in a solution. 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 the cycle C formed by u, v, and the endpoints of another option for e other than σ_i (illustrated in red in Figure 2(b)). By σ_i being is a non-consecutive option, C fulfills Property 2 for clashing options of edges in F incident to x and w.

(b) There are at least four consecutive non-cyclic options for e; see Figure 2(c). Let σ_i be one of the inner options. Then, similar to the previous case, we can find a blocking cycle as follows. If the option clashes with the rightmost (leftmost) option, we can find a blocking cycle formed by u, v and the endpoints of the leftmost (rightmost) option. Otherwise, the cycle formed by u, v and the first and last endvertex in the path formed by the consecutive options of e forms a cycle fulfilling Property 2 for the endvertices of σ_i.

(c) There are three consecutive or four cyclically consecutive options for e; see Figure 2(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 $V_\diamond$ be the set of vertices $\{u, v, w, x, y, z\}$ and $G_\diamond$ the octahedron subgraph of G induced by $V_\diamond$.
Edges in F with exactly one endpoint in $V_\diamond \setminus \{u\}$ 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 $V_\diamond$ cannot clash with any option of an edge between vertices in $V_\diamond$. 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 $k_{\geq 1}$-PIP: Inserting a path or a matching is NP-complete

The membership of $k_{\geq 1}$-PIP in NP is straightforward; we prove NP-hardness by reduction from Planar Monotone 3-SAT. 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 or *bars* 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 Figure 3 for an example. This problem is known to be NP-complete [10, 20]. The bars in Γ_ϕ can be layered decreasingly from top to bottom. We set the layer of the variables as layer zero, and insert two empty layers, one directly above and one below the variable layer. We denote by $L(c)$ the layer of clause c; see Figure 3.

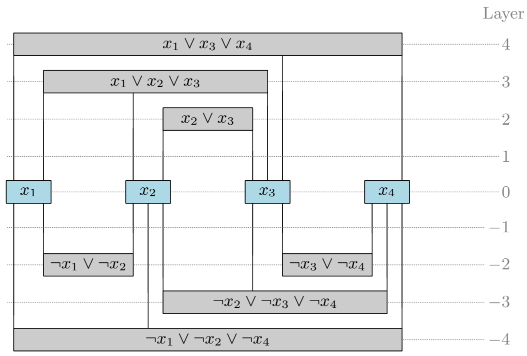

Figure 3 Rectilinear representation of the variable-clause incidence graph of a PLANAR MONOTONE 3-SAT instance.

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 7) and describe the changes to our construction for F being a matching afterwards (see Corollary 8).

We denote by H^+ the graph consisting of an axis-aligned $(k+1) \times (k+1)$-vertex grid, where all the $k+1$ vertices on the left side of the grid are connected to a vertex u, while all vertices on the right are connected to a vertex v. We create chains of copies of H^+, that are connected via the vertices u, v. Further, we denote by H^- the axis-aligned grid graph consisting of $(k-1) \times (k-1)$ vertices. In our construction of G, we create grids of H^- graphs, by connecting two opposing vertical or horizontal sides of their respective vertex grid via $k-1$ non-crossing edges. The grid construction can also be connected to copies of H^+ via $k-1$ non-crossing edges, leaving out the corner vertices of the H^+ vertex grid. If it is necessary to connect a single vertex v to an H^-, we connect v via a fan of $k-1$ edges to one side of the vertex grid. Note that for the case of $k = 1$, structures parameterized by $k-1$ such as H^- are meant to disappear from the construction. Figure 4(a) shows a structure consisting of multiple copies of H^+ and H^- and their schematic representation used in more complex figures. We say that an edge $e \in F$ is *ℓ-spanning* if there are ℓ different copies of H^+ in the chain between its endpoints.

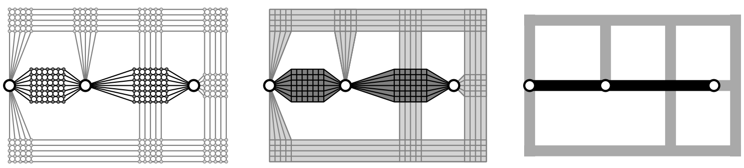

Figure 4 Different representations used in the drawings of our construction. (Left) every vertex and every edge, (middle) a simplification, and (right) a highly abstracted representation.

The variable gadget. We replace each bar of a variable x in Γ_ϕ by a variable gadget which consists of an H^+-chain of $4a+1$ copies, where a is the maximum over the number of positive and negative occurrences of x in ϕ. Let $u_1, \ldots, u_{4a+2}$ be the vertices that join the copies of H^+ as well as the two unjoined vertices of the first and last H^+ copy in the chain, from left to right. Moreover, we mark for $i \in \{0, \ldots, a-1\}$ the vertices u_{4i+3} as *variable endpoints* (squares in Figure 5). Each such vertex is incident to two *literal edges*, which are connecting 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*).

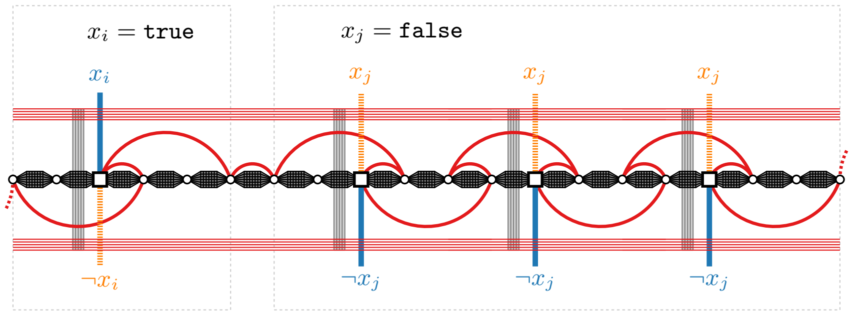

Figure 5 Drawing of the variable gadget illustrating Lemma 4.

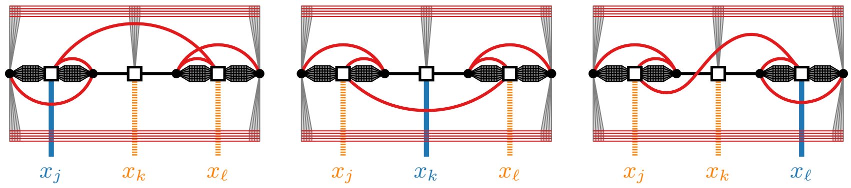

Figure 6 Drawing of the clause gadget illustrating Lemma 5.

For every copy of H^+ with position $4i+2$, $0 \le i < a$, we connect the top and the bottom side of its vertex grid to one copy of H^- each. For each side – top and bottom – of the H^+-chain, the copies of H^- will be connected by $k-1$ non-crossing edges in F as schematically shown in Figure 5. For an illustration showing all vertices and edges, see Figure 4(a).

For each variable gadget, its edges in F (bold red in Figure 5) then consist of alternating 3- and 1-spanning edges. Formally, for each $i \in \{0, \dots a-1\}$, the path F passes through the vertices u_{4i+1}, u_{4i+4}, u_{4i+3}, u_{4i+6}, u_{4i+5}, except for $a-1$, where we omit the last vertex.

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. Figures 5, 6, 8, and 10. The proofs of statements marked with a ($\star$) are available in [19].

▶ **Lemma 4** ($\star$). *Let v be a variable gadget described as above. Then, in any k-planar drawing containing v, 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.*

We think of the variable corresponding to the gadget as set to `true` if the negative literal edges are crossed, and to `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 Figure 5.

The clause gadget. We describe the construction only for the positive clauses; it works symmetrically for the negative ones. The clause gadget is depicted in Figure 6. 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 Figure 6 and add edges to F as shown in red. Further, we add three copies of H^- on the top and two on the bottom side of the gadget, and connect the left- and rightmost vertex in the gadget as well as the middle variable endpoint to the corresponding ones. Similar to the variable gadget, these H^- copies will be connected via edges in F as shown in Figure 6.

▶ **Lemma 5** (⋆). *Let c be a clause gadget drawn as described above. Then, in any k-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. Again, we only describe the construction for layers > 0 as the other side is symmetric. We insert H^+-chains with 1-spanning edges added to F on every layer > 0 of Γ_ϕ and insert the clause gadgets into the respective layers as shown in Figure 8. Further, we create variable endpoints on all layers > 0 in order to propagate the state of the variable gadgets to clauses in higher layers. Layer 1 thereby ensures that each variable endpoint can be connected to another endpoint, even if the respective literal edge is not used in a clause, as this is crucial to ensure the alternating pattern in the variable gadgets; see, e.g., x_1 in Figure 8. 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 insert *propagating edges*, by prescribing F to span the two neighboring copies of H^+. Further, we connect every two consecutive variable endpoints on layer j and $j+1$ with $1 \leq j < L(c)$ via a literal edge, as illustrated in Figure 8.

Both the variable and the clause gadget require each literal edge to have either $k-1$ or k crossings. Since the $k_{\geq 1}$-PIP problem requires Γ to be plane, it is not possible in our construction to create these crossings with edges in G, hence the path formed by edges F has to cross each literal edge $k-1$ times, in addition to one potential crossing by the propagation edges. To this end, we create a subpath P_i comprised of edges in F between each layer i and layer $i-1$ (if present), which is passing through copies of H^- and crosses the literal edges $k-1$ times; see Figure 7 for different levels of abstraction used in our illustrations. The subpath of F on every layer i is joined to P_i and P_{i+1} (if present) by an H^+-chain and 1-spanning edges. For $k = 1$ we simply connect the subpaths of F on each layer to the next by aH^+-chain and 1-spanning edges. Note that if k is even, the sides where the H^+-chain is located alternate, otherwise they connect on the same side of the drawing. Note that in our illustrations showing final the constructions for the graph given in Figure 3, we assume an even k. To ensure that the edges in each P_i do not exceed k crossings, we subdivide between each literal edge by inserting vertically connected copies of H^- as depicted in Figure 8.

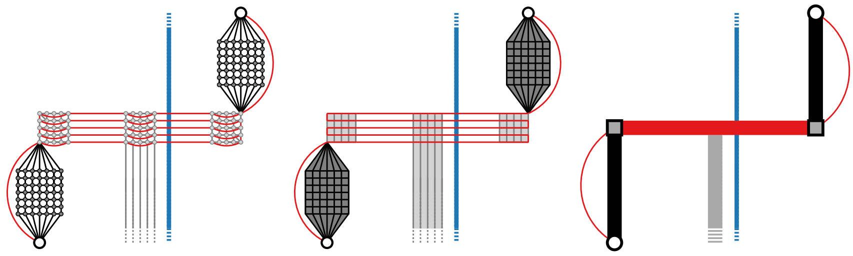

Figure 7 Different representations of the alternating path P_i between layer i and layer $i-1$.

▶ **Lemma 6** (⋆). *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 the first edge of P being uncrossed in the variable gadget does not necessarily lead to its last edge being uncrossed in layer $L(c)-1$. In fact, this is possible when multiple literals evaluate to `true` for a clause gadget; e.g., the top-left orange edge in Figure 8.

▶ **Theorem 7** ($\star$). *k-PIP is* NP*-complete for every $k \geq 1$, even if G is biconnected and F forms a path.*

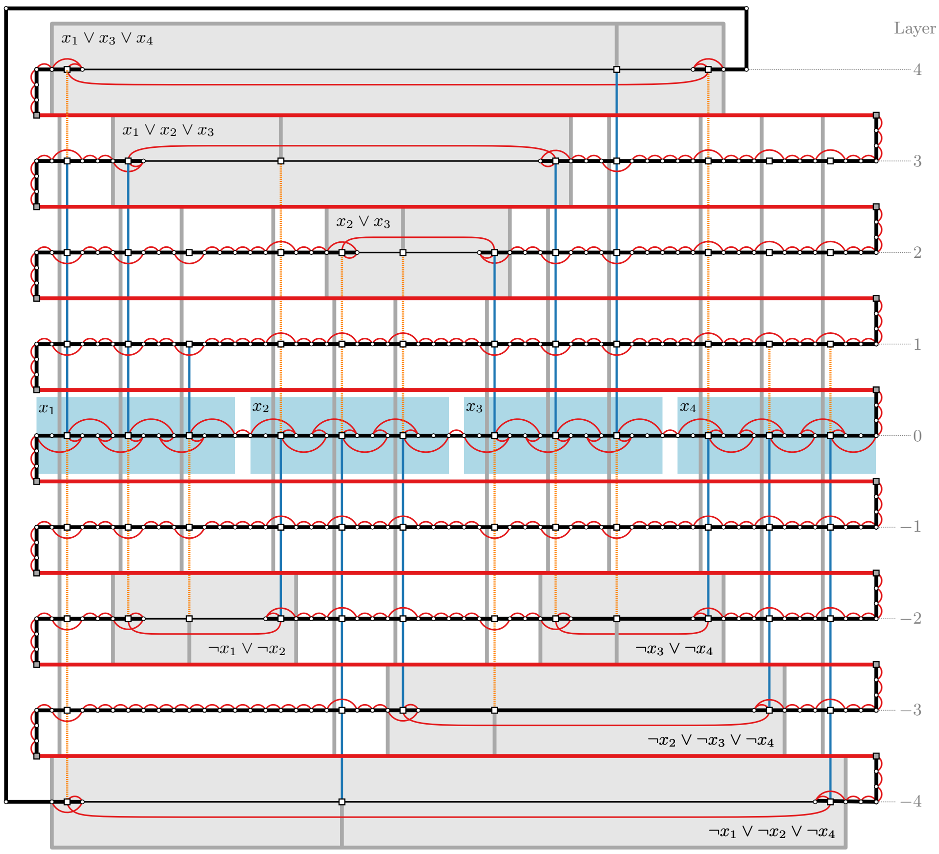

Figure 8 Solution (in red) of the k-PIP instance coming from the graph given in Figure 3.

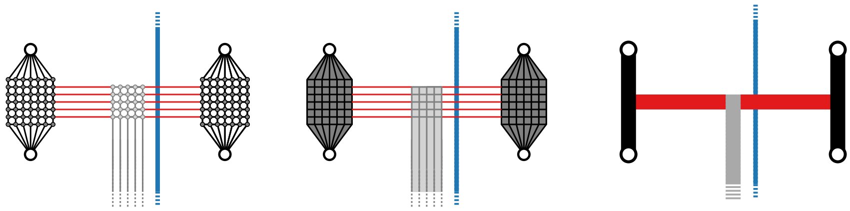

Figure 9 Different representations of the $k-1$ matching edges which substitute the alternating path P_i in the case that F is a matching.

We can use essentially the same construction, but replace the alternating connections between the layers by single edges to prove NP-hardness also for the case that F is a matching; see Figures 9 and 10.

▶ **Corollary 8.** *k-PIP is* NP*-complete for every $k \geq 1$, even if G is biconnected and F forms a matching.*

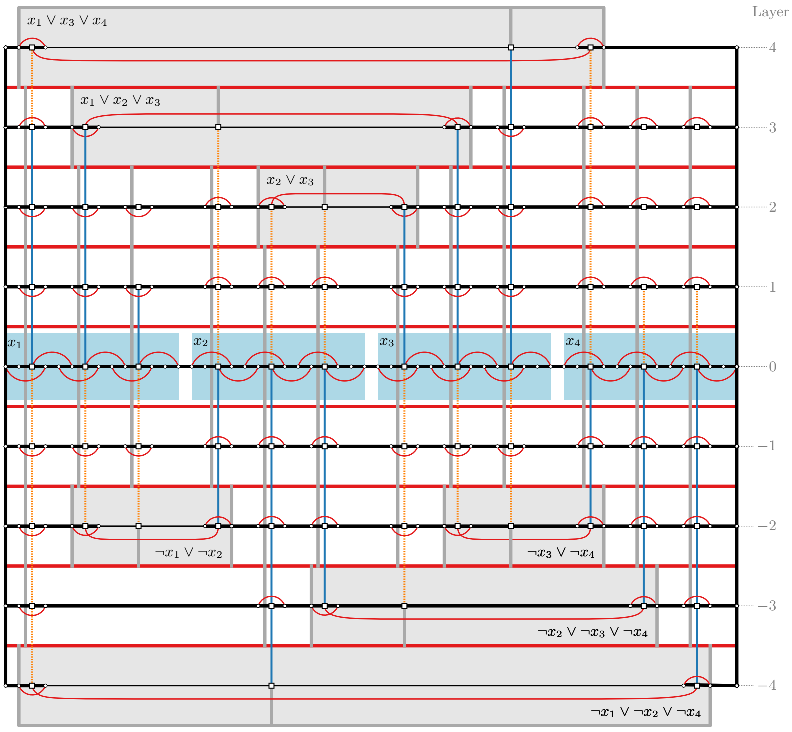

Figure 10 k-PIP instance where F is a matching reduced from the graph given in Figure 3.

4 Conclusion

We introduced the k-PIP problem and showed that it is NP-complete for every $k \geq 1$ even when the given graph is biconnected and the inserted edges form a path or matching. We also presented a linear-time algorithm for 1-PIP when the given graph is triangulated. This naturally raises the question if the triconnected case of 1-PIP is also polynomial-time solvable.

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, Martin Derka, and Irene Parada. Extending simple drawings. In *Proceedings of the 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, 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.
5. 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.

6 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.

7 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.

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

9 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.

10 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.

11 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.

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

13 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.

14 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.

15 Robert Ganian, Thekla Hamm, Fabian Klute, Irene Parada, and Birgit Vogtenhuber. Crossing-optimal extension of simple drawings. In *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.

16 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.

17 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.

18 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.

19 Julia Katheder, Philipp Kindermann, Fabian Klute, Irene Parada, and Ignaz Rutter. On k-plane insertion into plane drawings. *CoRR*, abs/2402.14552, 2024. doi:10.48550/arXiv.2402.14552.

20 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.

21 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.

22 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.

23 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.

24 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.

25 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.

26 Thomas Ziegler. *Crossing minimization in automatic graph drawing.* PhD thesis, Saarland University, Saarbrücken, Germany, 2001. URL: https://d-nb.info/961610808.