Extending simple drawings with one edge is hard

Alan Arroyo*[1], Fabian Klute†[2], Irene Parada‡[3], Raimund Seidel§[4], Birgit Vogtenhuber¶[3], and Tilo Wiedera‖[5]

[1]IST Austria, Austria
[2]TU Wien, Austria
[3]Graz University of Technology, Austria
[4]Universität des Saarlandes, Germany
[5]Osnabrück University, Germany

September 17, 2019

Abstract

A simple drawing $D(G)$ of a graph $G = (V, E)$ is a drawing in which two edges have at most one point in common that is either an endpoint or a proper crossing. An edge e from the complement of G can be inserted into $D(G)$ if there exists a simple drawing of $G' = (V, E \cup \{e\})$ containing $D(G)$ as a subdrawing. We show that it is NP-complete to decide whether a given edge can be inserted into a simple drawing, by this solving an open question by Arroyo, Derka, and Parada.

1 Introduction

A *simple drawing* of a graph G (also known as *good drawing* or as *simple topological graph* in the literature) is a drawing $D(G)$ of G in the plane such that every pair of edges share at most one point that is either a proper crossing (no tangent edges allowed) or a common endpoint. Moreover, no three edges intersect in the same point and edges must not contain other vertices in their relative interior. Simple drawings have received a great deal of attention in various areas of graph drawing, especially in connection with two long-standing open problems: the crossing number of the complete graph [22] and Conway's thrackle conjecture.

The complement of a graph $G = (V, E)$ is the simple graph $\overline{G}$ with the same vertex set as G and where two distinct vertices in $\overline{G}$ are adjacent if and only if they are not adjacent in G. Given a simple drawing $D(G)$ of a graph $G = (V, E)$ and an edge e from the complement $\overline{G}$ of G, we say that e can be *inserted* into $D(G)$ if there exists a simple drawing of $G' = (V, E \cup \{e\})$ that contains $D(G)$ as a subdrawing. In that case, we say that $D(G)$ can be *extended* with e.

Recently, Arroyo, Derka, and Parada [2] showed that it is NP-complete to decide if a simple drawing $D(G)$ of a graph G can be extended with a set of edges from the complement of G. A central question arising from their paper asks if it is possible to decide in polynomial time whether a given edge from the complement of G can be inserted into $D(G)$. Under the assumption that $\mathsf{P} \neq \mathsf{NP}$, we give a negative answer to this question by showing that deciding whether one given edge can be inserted into a simple drawing is already NP-complete.

*Email: alanmarcelo.arroyoguevara@ist.ac.at. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 754411.

†Email: fklute@ac.tuwien.ac.at.

‡Email: iparada@ist.tugraz.at. Partially supported by the Austrian Science Fund (FWF): W1230 and also within the collaborative DACH project *Arrangements and Drawings* as FWF project I 3340-N35.

§Email: rseidel@cs.uni-saarland.de.

¶Email: bvogt@ist.tugraz.at. Partially supported by Austrian Science Fund within the collaborative DACH project *Arrangements and Drawings* as FWF project I 3340-N35.

‖Email: tilo.wiedera@uos.de. Supported by the German Research Foundation (DFG) grant CH 897/2-2.

One of the implications of the results presented in this paper concerns so-called saturated drawings. A simple drawing $D(G)$ of a graph G is called *saturated* if no edge e from $\overline{G}$ can be inserted into $D(G)$. It is known that there are saturated simple drawings with a linear number of edges [11, 17]. However, no polynomial-time algorithm for deciding whether a simple drawing is saturated is known. Our hardness result implies that the straight-forward idea of testing if $D(G)$ is saturated by checking for every edge in $\overline{G}$ whether it can be inserted into $D(G)$ is not feasible unless $\mathsf{P} = \mathsf{NP}$.

Several questions concerning the number of crossings in planar graphs with one additional edge have been studied [5, 10, 21]. Furthermore, the question considered in this paper is strongly related to work on extending partial representations (not necessarily drawings) of graphs. Here, we are usually given a representation of a part of the graph G and are asked to extend it into a full representation of G such that the partial representation is a sub-representation of the full one. Recent years have seen a plethora of results in this area [1, 3, 4, 6, 7, 8, 9, 12, 13, 14, 15, 16, 19, 20].

Complementing our hardness result in Section 2, we show in Section 3 that, if the number of crossings in a simple drawing $D(G)$ of a graph G is bounded by a constant k, it is fixed-parameter-tractable with respect to k to decide if an edge e from the complement of G can be inserted into $D(G)$.

2 Inserting one edge is hard

Theorem 1. *Given a simple drawing $D(G)$ of a graph $G = (V, E)$ and an edge uv of the complement of G, it is NP-complete to decide if uv can be inserted into $D(G)$, even if $V \setminus \{u, v\}$ induces a matching in G and u and v are isolated vertices.*

The problem is in NP since it can be described combinatorially. See Arroyo et al. [2] for details.

We show NP-hardness via a reduction from $\mathsf{3SAT}$. Let $\phi(x_1, \dots x_n)$ be a $\mathsf{3SAT}$-formula with *variables* $x_1, \dots, x_n$ and set of *clauses* $\mathcal{C} = \{C_1, \dots, C_m\}$. An occurence of a variable x_i in a clause $C_j \in \mathcal{C}$ is called a *literal*. For convenience, we assume that in $\phi(x_1, \dots, x_n)$ each clause has three literals (possibly with duplicated literals). First, in a preprocessing step, we transform $\phi(x_1, \dots, x_n)$ into an equivalent formula, in which no clause has three positive or three negative literals.

Claim 1. *The following transformation of the clauses in a formula preserves satisfiability of the formula:*

$$x_i \vee x_j \vee x_k \Rightarrow \begin{cases} x_k \vee y \vee \texttt{false} & \textit{(i)} \\ x_i \vee x_j \vee \neg y & \textit{(ii)} \end{cases} \qquad \neg x_i \vee \neg x_j \vee \neg x_k \Rightarrow \begin{cases} \neg x_i \vee \neg x_j \vee y & \textit{(iii)} \\ \neg x_k \vee \neg y \vee \texttt{false} & \textit{(iv)} \end{cases}$$

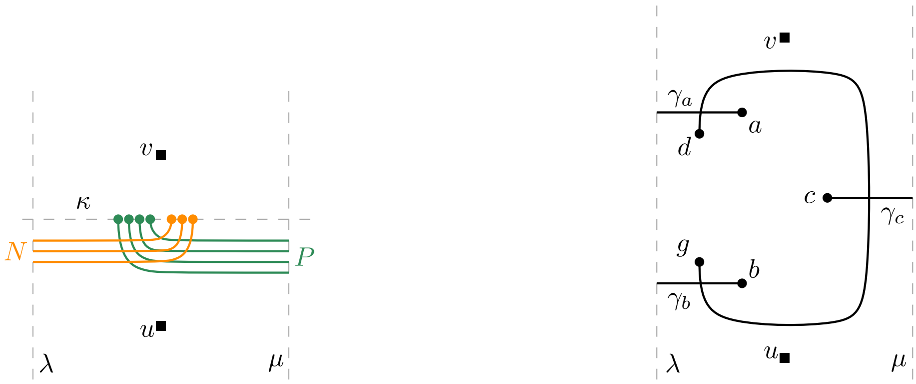
where y is a new variable for each transformed clause and `false` *is the constant truth value* `false`.

Proof. We prove the statement for the case in which the original clause has three positive literals, the other case is analogous. Assume x_i or x_j satisfies the original clause. Then it also satisfies Clause (ii) and y can be set to `true` to satisfy Clause (i). If x_k satisfies the original clause, then it also satisfies Clause (i) and y can be set to `false` to satisfy Clause (ii). If none of x_i, x_j, and x_k satisfies the original clause, then to satisfy Clause (ii) we have to set y to `false`, which implies that Clause (i) is not satisfied. □

Note that after transforming a formula, the clauses are of four types depending on the number of positive and negative literals (and `false` constants). Clauses (i)–(iv) in Claim 1 are each of one of these types. Consequently, we denote these types as Type (i)–(iv). This means that clauses of Type (i) have two positive literals and one constant `false`, clauses of Type (ii) have two positive and one negative literal, Type (iii) clauses contain two negative and one positive literal, and finally, a Type (iv) clause has two negative literals and one constant `false`.

Given a transformed $\mathsf{3SAT}$-formula $\phi(x_1, \dots, x_n)$ with set of clauses $\mathcal{C} = \{C_1, \dots, C_m\}$, the reduction uses gadgets consisting of simple drawings to represent the variables and clauses. Satisfiability of $\phi(x_1, \dots, x_n)$ will correspond to being able to insert a given edge uv into a simple drawing D of a matching constructed from the formula ϕ. The main idea of the reduction is that the variable and clause gadgets act as barriers inside a simple closed region R of D, in which we need to insert an arc γ from one side to the other to complete the connection between u and v.

To simplify the description, we first restrict our attention to the inside of the simple closed region R. We assume that γ cannot cross the boundary of R. In the following we use two lines, named λ and μ, to describe the variable and clause gadget. Later, these will be identified with opposite segments on the boundary of R.



(a) Variable gadget. The orange arcs belong to N, the green ones to P.

(b) Clause gadget.

Figure 1: The two main gadgets of the reduction.

Variable gadget. A variable gadget W includes two sets of arcs (parts of later-defined edges), P and N, that correspond to positive and negative appearances of a variable, respectively. The gadget is bounded on the left by a line λ and on the right by a line μ. Arcs in P and N have one endpoint on a horizontal line κ such that the endpoints of arcs in P are to the left of the endpoints of arcs in N and for both sets lie between λ and μ. The other endpoint of arcs in P and N lies below κ and on μ and λ, respectively. Notice that an arc in P intersects every arc in N, and vice versa; see Figure 1a for an illustration. Finally, we choose two points u and v such that u is below all arcs in W and v is above them.

Lemma 1. *Let W be a variable gadget. Any arc between the vertical lines λ and μ that connects u and v crosses either all arcs in P or all arcs in N.*

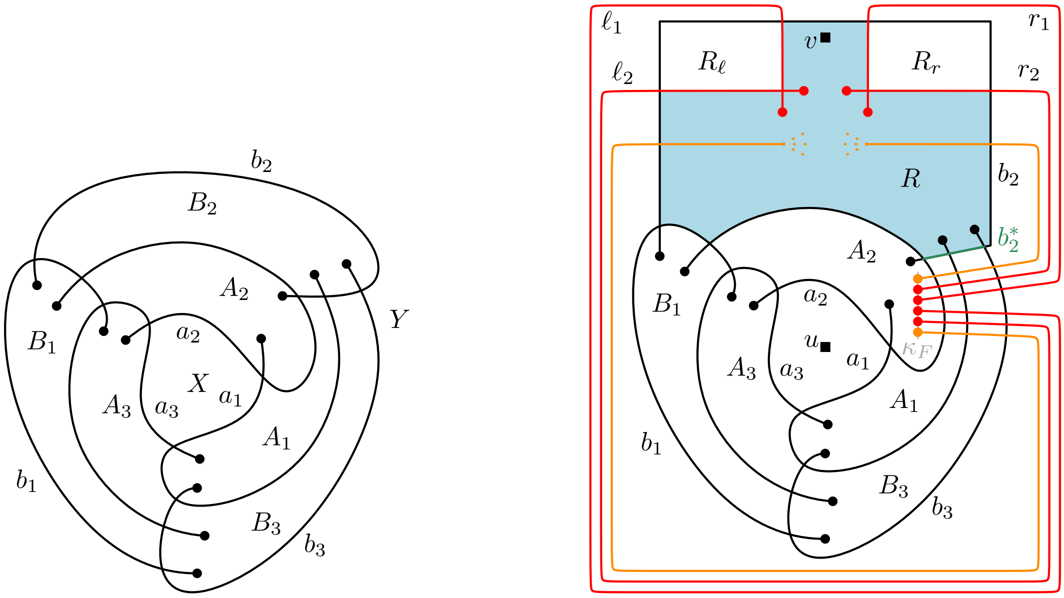
Proof. Assume that there is an arc connecting u and v neither crossing all the arcs in P nor all the arcs in N. Hence, there are two arcs $p \in P$ and $n \in N$ such that this arc neither crosses p nor n. By the construction of the gadget, p and n cross. Thus, their union together with λ and μ separates u from v. It follows that the arc has to cross either p or n, a contradiction. □

Clause gadget. A clause gadget K includes three arcs γ_a, γ_b, and γ_c (parts of later-defined edges) incident to three points a, b, and c, respectively, and an arc (edge) dg incident to two other points d and g. As in the variable gadget, the clause gadget is bounded on the left by a line λ and on the right by a line μ. The arcs γ_a and γ_b have their other endpoint on λ and γ_c has its other endpoint on μ. None of these three arcs intersect. The arc dg is placed such that it crosses γ_a, γ_c, and γ_b in that order as we traverse it from d to g; see Figure 1b for an illustration. Notice that we do not require any specific rotation of the crossings of dg with γ_a and γ_b (where the rotation is the clockwise order of the endpoints of the crossing arcs). Finally, we choose two points u and v such that u is below all arcs in K and v is above them.

Lemma 2. *Let K be a clause gadget. Any arc uv between the vertical lines λ and μ that connects u and v crosses either dg twice or at least one of the arcs γ_a, γ_b, and γ_c.*

Proof. Let $\times$ be the crossing point of γ_c and dg. This point splits the arc dg into two arcs $d\times$ and $g\times$. Assume that the arc uv does not cross the arcs γ_a, γ_b, and γ_c. The union of γ_a and γ_c together with $d\times$ and the lines λ and μ separates u from v. Since the arcs γ_a and γ_c are not crossed by uv, uv must cross $d\times$ in a point that is not $\times$. Analogously, the union of γ_b, γ_c, together with $g\times$ and the lines λ and μ separates u from v. Thus, uv has to cross $g\times$ in a point that is not $\times$. This implies that uv crosses dg twice. □

The reduction. Let $\phi(x_1, \ldots, x_n)$ be a 3SAT-formula transformed as described in Claim 1 and with clause set $\mathcal{C} = \{C_1, \ldots, C_m\}$ (each clause being of one of the four types described above). To build our



(a) The simple drawing ◎ presented by Kynčl [18]. It is impossible to insert an edge between a point in X and one in Y.

(b) A schematic overview of the edges in F (red and orange) and how they are combined with ◎.

Figure 2: Last gadget for the reduction acting as a frame.

reduction we need one more gadget. First, we introduce the following simple drawing described by Kynčl et al. [18, Figure 11] and depicted in Figure 2a. Here, we denote this drawing with ◎. Following the notation by Kynčl et al., we denote its six arcs with a_1, a_2, a_3, b_1, b_2, and b_3 and its eight cells with X, A_1, A_2, A_3, B_1, B_2, B_3, and Y; see Figure 2a. The core property $\mathcal{P}$ of ◎ is that it is not possible to insert an edge between a point in cell X and another point in cell Y such that the result is a simple drawing [18, Lemma 15].

For our reduction we first choose two arbitrary points u and v in the cells X and B_2 and insert them as vertices to ◎. Let ◎′ be the simple drawing in which we inserted vertices u and v into ◎. Finally, let b_2^* be the part of the arc b_2 between the crossing point of b_2 and a_2 and the crossing point of b_2 and b_3.

Lemma 3. *The edge uv cannot be inserted into ◎′ without crossing b_2^*.*

Proof. Assume for contradiction that uv can be inserted not crossing b_2^* and let γ_{uv} be such an arc. If γ_{uv} did not cross b_2, then we would be able to prolong it and cross b_2 to reach Y, a contradiction to property $\mathcal{P}$. Thus, γ_{uv} crosses b_2. Further, we may assume w.l.o.g. that γ_{uv} does not cross b_2 inside A_2 or B_1, as otherwise it would be possible to modify γ_{uv} to not cross b_2. Thus, γ_{uv} intersects B_2 on one side of the crossing with b_2. Since γ_{uv} cannot intersect Y, this crossing must be on b_2^*. □

The final piece we need for our a reduction is a set F of $m^I + m^{IV} + 4$ arcs that we insert into ◎′, where m^I is the number of clauses of Type (i) and m^{IV} the number of clauses of Type (iv). For an arc $f \in F$ we will place one of its endpoints on a vertical line κ_F inside A_2 and the other one inside B_2. The only crossings of f with ◎′ are with the arcs a_2, a_1, b_3, and b_2, in that order when traversing f from its endpoint on κ_F to its endpoint in B_2. Furthermore, f, traversed in that direction, crosses from A_2 to A_1, from A_1 to B_3, from B_3 to Y, and from Y to B_2.

Consider the $m^I + m^{IV} + 4$ endpoints on κ_F sorted from top to bottom. We denote with f_j the arc in F incident to the j-th such endpoint. When traversing b_2 from its endpoint in A_2 to its endpoint in B_1, the crossings of arcs in F with b_2 appear in the same order as their endpoints on κ_F. More precisely, the crossings of b_2 when traversed in that direction are with a_2, a_1, b_3, f_1, f_2, …, $f_{|F|}$, and b_1.

The arcs f_{m^I+1}, f_{m^I+2}, f_{m^I+3}, and f_{m^I+4} will behave differently than the other arcs in F. In the following, we denote these four arcs with r_2, r_1, ℓ_1, and ℓ_2, respectively. There are only two crossings between arcs in F, namely of r_1 and r_2, and of ℓ_1 and ℓ_2, and both these crossings are inside B_2. These four crossing arcs divide B_2 into three regions. We denote the region with b_2^* on its boundary with R, the (other) region with the crossing of r_1 and r_2 on its boundary with R_r, and the (other) region with the crossing of ℓ_1 and ℓ_2 on its boundary with R_ℓ. Arcs r_1, r_2, ℓ_1, and ℓ_2 must be drawn such that the vertex v lies in R; see Figure 2b for an illustration. The precise endpoints of the edges in $F \setminus \{r_1, r_2, \ell_1, \ell_2\}$ will be fixed when we insert the clause gadgets.

Lemma 4. *The edge uv cannot be inserted into $◎'$ without crossing every arc in F inside A_1 or B_3.*

Proof. Assume for contradiction that there is an arc $f \in F$ such that uv does not cross f. From Lemma 3 we know that uv has to cross b_2^*. Consider the region bounded by b_2^*, b_3, f, and a_2. Observe that, since b_2^* is fully contained on the boundary of this region, uv has to cross at least one of the three other arcs as well. By assumption, uv does not cross f. Crossing b_3 is impossible by property $\mathcal{P}$, as the part contained on this region's boundary separates B_3 from Y. Finally, crossing the arc which is part of a_2 is not possible, since this would imply the existence of a point v' in A_2 such that uv passes through v' without having crossed a_2. Hence, we could prolong the arc uv' that is part of uv by crossing a_2 such that it reaches B_2 without having crossed b_2^*, a contradiction to Lemma 3. Furthermore, as we do not allow more than two arcs to cross in one point, the statement follows. □

It remains to insert inside R the clause and variable gadgets and precisely define the endpoints of arcs in $F \setminus \{\ell_1, \ell_2, r_1, r_2\}$. For simplicity, we first insert the variable gadgets and then the clause gadgets. The idea is that each clause and variable gadget is inserted in R separating b_2^* from v. This is done by identifying the endpoints that were lying on λ or μ with points on ℓ_1, ℓ_2, r_1, r_2, or b_2. As a result, Lemmas 1 and 2 can be applied to the arc that we aim to insert connecting u and v in the final simple drawing, since it has to cross b_2^* by Lemma 3.

We insert now the variable gadgets into R. Let $W^{(i)}$ be the variable gadget corresponding to variable x_i. For a gadget $W^{(i)}$, the arcs in N are drawn such that the endpoints on λ, lie on the part of ℓ_1 that bounds R. The arcs in P are drawn similarly, but with the endpoints on μ lying on the part of r_1 that bounds R. Moreover, we identify vertex v in the gadget with vertex v in $◎'$. Gadgets corresponding to different variables are inserted without crossing each other. We now specify how they are inserted relative to each other. As we traverse ℓ_1 from its endpoint on κ_F to its endpoint in R we encounter the endpoints of arcs in $W^{(i)}$ before the endpoints of arcs in $W^{(i+1)}$. Analogously, as we traverse r_1 from its endpoint on κ_F to its endpoint in R we encounter the endpoints of arcs in $W^{(i)}$ before the endpoints of arcs in $W^{(i+1)}$. An illustration is presented in Figure 3.

The clause gadgets are inserted in a similar way. Let $K^{(j)}$ be the clause gadget corresponding to clause C_j. If C_j is of Type (i), $K^{(j)}$ is inserted such that the endpoints on λ lie on the part of ℓ_2 that bounds R. If C_j is the j'-th clause of Type (i), we identify c with the endpoint of the arc $f_{j'}$. Similarly, if C_j is of Type (iv), $K^{(j)}$ is inserted such that the endpoints on λ lie on the part of r_2 that bounds R. If C_j is the j'-th clause of Type (iv), we identify c with the endpoint of the arc $f_{m^I+4+j'}$. If C_j is of Type (ii), $K^{(j)}$ is inserted such that the endpoints on λ lie on the part of ℓ_2 that bounds R and the endpoint on μ lies on the part of r_2 that bounds R. Similarly, if C_j is of Type (iii), $K^{(j)}$ is inserted such that the endpoint on μ lies on the part of ℓ_2 that bounds R and the endpoints on λ lie on the part of r_2 that bounds R. The crossings in R of arcs from different clause gadgets are of arcs with an endpoint in r_2 with arcs in $\{f_j : 1 \leq j \leq m^I\}$.

We now specify how different clause gadgets are inserted relative to each other. As we traverse ℓ_2 from its endpoint on κ_F to its endpoint in R we encounter the endpoints of arcs corresponding to clauses of Type (iii) before the ones corresponding to clauses of Type (ii), and those before the ones corresponding to clauses of Type (i). Analogously, as we traverse r_2 from its endpoint on κ_F to its endpoint in R we encounter the endpoints of arcs corresponding to clauses of Type (iv) before the ones corresponding to clauses of Type (iii), and those before the ones corresponding to clauses of Type (ii). Moreover, as we traverse ℓ_2 and r_2 in the specified directions, the endpoints of arcs corresponding to the j'-th clause of a certain type are encountered before the endpoints of arcs corresponding to the $j' - 1$-st clause of this type. An illustration is presented in Figure 3.

Finally, we connect arcs from variable and clause gadgets inside the regions R_ℓ and R_r. This is done such that if a literal in a clause is x_k then the corresponding arc in the clause gadget, that has an endpoint on ℓ_2, is connected with an arc in N of the gadget $W^{(k)}$, that has an endpoint on ℓ_1. Thus, these connections can lie in R_ℓ. Analogously, if a literal in a clause is $\neg x_k$ then the corresponding arc in the clause gadget, that has an endpoint on r_2, is connected with an arc in P of the gadget $W^{(k)}$, that has an endpoint on r_1. Thus, these connections can lie in R_r. Since, without loss of generality, we can assume that R_ℓ and R_r are convex regions and the endpoints we want to connect are in general position (no three on the same line), the connections can be drawn as straight-line segments. (For clarity, in Figure 3 these connections have one bend per arc.) Therefore, there is at most one crossing between each pair of connecting arcs.

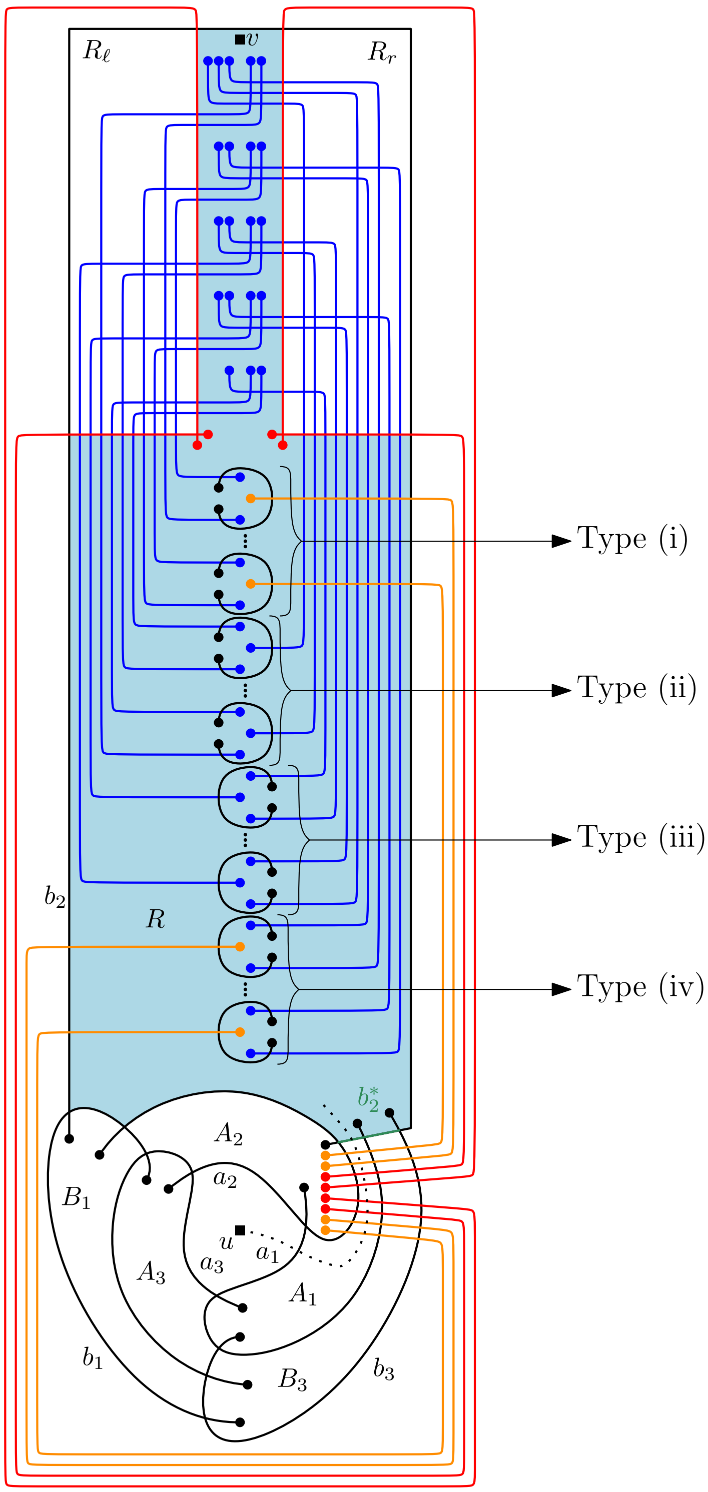


Figure 3: Illustration of the reduction.

Each connecting arc can be concatenated with the arcs in a variable and in a clause gadget that it joins. These concatenated arcs are edges in our drawing that have one endpoint in a variable gadget and the other in a clause gadget. By construction, each of them corresponds to a literal in the formula ϕ and each pair of these edges crosses at most once. Similarly, the arcs in $F \setminus \{\ell_1, \ell_2, r_1, r_2\}$ have one endpoint in a clause gadget and also define a set of edges in our final drawing that we denote with the same name as the corresponding arcs.

We now have all the pieces that constitute our final drawing. It consists of (i) the simple drawing $\circledcirc'$; (ii) the edges $f_i \in F$ drawn as the described arcs (with their endpoints as vertices); (iii) the edges corresponding to literals (with their endpoints as vertices); and (iv) the edges dg in each clause gadget (with d and g as vertices). Observe that the constructed drawing is a simple drawing, as it is the drawing of a matching (plus the vertices u and v) and, by construction, two edges cross at most once.

Correctness. It is now straight-forward to show that the presented construction is a valid reduction.

Lemma 5. *The above construction is a polynomial-time reduction from* 3SAT *to the problem of deciding if an edge can be inserted into a simple drawing.*

Proof. Given a 3SAT formula $\phi(x_1, \ldots, x_n)$ with clauses $C_1, \ldots, C_m$ we construct a simple drawing D as above and aim to insert the edge uv into it. This construction can clearly be computed in polyonmial time and space, specially since only the combinatorial description of the drawing is needed.

Assume uv can be inserted into D and let uv be the resulting arc. By Lemmas 3 and 4 we know that uv has to cross b_2^* and every arc in F. Let u^* be the point where uv crosses b_2^*. Each clause and variable gadget separates u^* from v and thus, Lemmas 1 and 2 can be applied. This means that in a variable gadget $W^{(i)}$ either all arcs in P or all arcs in N are crossed. In the former case we assign to variable x_i the value `true`, and otherwise the value `false`. Assume that this truth assignment does not satisfy $\phi(x_1, \ldots, x_n)$. Then there exists a clause C_j for which all three literals evaluate to `false`. Consider the clause gadget $K^{(j)}$. By Lemma 2 we must cross in it an edge corresponding to one of its literals. However, by Lemma 4 an edge corresponding to the constant value `false` cannot be crossed (again) in a clause gadget. By construction and the truth assignment of the variables, the edges corresponding to the other literals of C_j cannot be crossed either.

Conversely, assume we are given a satisfying assignment of $\phi(x_1, \ldots, x_n)$. We then can insert uv into D as follows. Starting from u, edge uv crosses a_1 to enter region A_1, then crosses all arcs in F, and crosses b_2^* to enter R; see also the dotted line in Figure 3. In each clause gadget, edge uv crosses one edge corresponding to a literal evaluating to `true`, none corresponding to a literal evaluating to `false`, and the edge dg in the gadget if necessary. By construction, this leaves in each variable gadget all arcs either in P or in N free to be crossed by uv. Moreover, this allows us to connect u and v without crossing any edge twice. □

Remarks. The presented reduction from 3SAT constructs a simple drawing of a matching, and thus, the problem remains NP-hard when G is as sparse as possible (isolated vertices that are not the starting or ending vertices of the edge that we aim to insert can be disregarded for our problem, and thus we can restrict our attention to graphs without such vertices). We remark that if we don't require G to be a matching, our variable gadget can be simplified by identifying all the vertices on κ and removing the crossings between edges in N and P. Moreover, the disconnectedness of the produced instance is not a restriction. If an instance $D(G)$ is a simple drawing of a disconnected graph G we can transform it to an equivalent instance consisting of a simple drawing of a connected graph by inserting an apex vertex into any cell of the drawing and subdividing its incident edges that connect to all the vertices of $D(G)$.

3 FPT-algorithm for bounded number of crossings

In this section we show that for drawings with a bounded number of crossings it can be decided in FPT-time if an edge can be inserted.

Theorem 2. *Given a simple drawing $D(G)$ of a graph $G = (V, E)$ and an edge uv of the complement of G, there is an* FPT*-algorithm in the number k of crossings in $D(G)$ for deciding whether uv can be inserted into $D(G)$.*

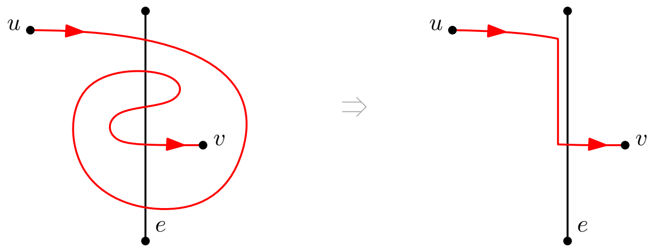


Figure 4: Rerouting uv when it crosses an edge uncrossed in $D(G)$ more than once.

Proof. Let $n = |V|$ and $m = |E|$ be the number of vertices and edges of G, respectively. We consider a subdrawing $D'(G')$ of $D(G)$ consisting of the edges incident to u and v, the (at most $2k$) edges which are crossed, and the vertices incident to all these edges together with u and v if they are isolated (and thus, not yet included). We first show that we can decide in FPT-time in k whether uv can be inserted into $D'(G')$. We then argue that uv can be inserted into $D'(G')$ if and only if it can be inserted into $D(G)$.

As in [2], we reformulate the problem of inserting an edge into a simple drawing as a problem in the dual graph of its planarization, in which crossings are replaced by vertices resulting in a plane drawing. Given a simple drawing $D(G)$ of a graph G, the *dual graph* $G^*(D)$ is the plane dual of the planarization of $D(G)$. Thus, every vertex in $G^*(D)$ corresponds to a cell in $D(G)$ and every edge in $G^*(D)$ corresponds to a segment of an edge in $D(G)$. We assign to each edge in $D(G)$ a different color (label) and define a coloring χ of the edges of $G^*(D)$, where every edge in $G^*(D)$ inherits the color of the edge in $D(G)$ containing the segment to which it corresponds. Given two vertices $u, v \in V$, let $G^*(D, \{u, v\})$ be the subgraph of $G^*(D)$ obtained by removing from it the edges corresponding to segments of edges incident to u or to v. We denote with χ' the coloring of the edges of $G^*(D, \{u, v\})$ that coincides with χ in every edge. The problem of extending $D(G)$ with one edge uv is then equivalent to the existence of a path in $G^*(D, \{u, v\})$ between a vertex corresponding to a cell incident to u and a vertex corresponding to a cell incident to v in which no color given by χ is repeated (that is, it is *heterochromatic*).

The number of segments of crossed edges in $D'(G')$ is at most $4k$. Thus, $G^*(D', \{u, v\})$ has at most $4k$ edges (but the number of vertices might not be bounded by a function of k). There are $O(n)$ cells with u on their boundary, and we consider the vertices of $G^*(D', \{u, v\})$ corresponding to them all as the possible starting cells of a valid heterochromatic path. Thus, the algorithm checking whether uv can be inserted into $D'(G')$ runs in $O(nk2^{4k})$ time.

We now argue that uv can be inserted into $D'(G')$ if and only if it can be inserted into $D(G)$. Since $D'(G')$ is a subdrawing of $D(G)$, it is clear that if uv cannot be inserted into $D'(G')$ then it cannot be inserted into $D(G)$. Suppose that uv can be inserted into $D'(G')$ and let γ be a *valid drawing* of uv in $D'(G')$ resulting in a simple drawing. We orient γ from u to v. If γ is not a valid drawing of uv in $D(G)$ then it must cross more than once an edge e uncrossed in $D(G)$. We can modify γ such that it is routed close to e between its first and last crossings with e, producing at most one intersection; see Figure 4 for an illustration. Repeating this process for every edge uncrossed in $D(G)$ and crossed by γ more than once we obtain a valid drawing of uv in $D(G)$. □

4 Conclusions

In this paper we showed that given a simple drawing $D(G)$ of a graph G it is NP-hard to decide if a particular edge from the complement of G can be inserted into $D(G)$ such that the result is a simple drawing. On the positive side, we proved that the problem is FPT with respect to the number of crossings of $D(G)$.

In the light of our results, checking whether a simple drawing $D(G)$ is saturated by trying to insert every edge of the complement of G is hopeless (unless P = NP). Thus, it is an interesting open problem whether there is a polynomial algorithm for deciding if a simple drawing is saturated.

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.

[2] 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)*, 2019. To appear.

[3] Alireza Bagheri and Mohammadreza Razzazi. Planar straight-line point-set embedding of trees with partial embeddings. *Information Processing Letters*, 110(12-13):521–523, 2010.

[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)*, pages 2000–2011, 2017.

[5] Sergio Cabello and Bojan Mohar. Adding one edge to planar graphs makes crossing number and 1-planarity hard. *SIAM Journal on Computing*, 42(5):1803–1829, 2013.

[6] Steven Chaplick, Paul Dorbec, Jan Kratochvíl, Mickaël Montassier, and Juraj Stacho. Contact representations of planar graphs: extending a partial representation is hard. In *Proceedings of the 40th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'14)*, pages 139–151. Springer, 2014.

[7] Steven Chaplick, Radoslav Fulek, and Pavel Klavík. Extending partial representations of circle graphs. *Journal of Graph Theory*, 91(4):365–394, 2019.

[8] Steven Chaplick, Grzegorz Guśpiel, Grzegorz Gutowski, Tomasz Krawczyk, and Giuseppe Liotta. The partial visibility representation extension problem. *Algorithmica*, 80(8):2286–2323, 2018.

[9] Giordano Da Lozzo, Giuseppe Di Battista, and Fabrizio Frati. Extending upward planar graph drawings. In *Proceedings of the 16th International Symposium on Algorithms and Data Structures (WADS'19)*, pages 339–352, 2019.

[10] Carsten Gutwenger, Petra Mutzel, and René Weiskircher. Inserting an edge into a planar graph. *Algorithmica*, 41(4):289–308, 2005.

[11] Péter Hajnal, Alexander Igamberdiev, Günter Rote, and André Schulz. Saturated simple and 2-simple topological graphs with few edges. *Journal of Graph Algorithms and Applications*, 22(1):117–138, 2018.

[12] Vít Jelínek, Jan Kratochvíl, and Ignaz Rutter. A Kuratowski-type theorem for planarity of partially embedded graphs. *Computational Geometry: Theory and Applications*, 46(4):466–492, 2013. Special Issue on the 27th Annual Symposium on Computational Geometry (SoCG'11).

[13] Pavel Klavík, Jan Kratochvíl, Tomasz Krawczyk, and Bartosz Walczak. Extending partial representations of function graphs and permutation graphs. In *Proceedings of the 20th European Symposium on Algorithms (ESA'12)*, pages 671–682, 2012.

[14] 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.

[15] Pavel Klavík, Jan Kratochvíl, Yota Otachi, and Toshiki Saitoh. Extending partial representations of subclasses of chordal graphs. *Theoretical Computer Science*, 576:85–101, 2015.

[16] Pavel Klavík, Jan Kratochvíl, Yota Otachi, Toshiki Saitoh, and Tomás Vyskocil. Extending partial representations of interval graphs. *Algorithmica*, 78(3):945–967, 2017.

[17] Jan Kyncl. Improved enumeration of simple topological graphs. *Discrete & Computational Geometry*, 50(3):727–770, 2013.

[18] Jan Kynčl, János Pach, Rados Radoicic, and Géza Tóth. Saturated simple and k-simple topological graphs. *Computational Geometry: Theory and Application*, 48(4):295–310, 2015.

[19] Tamara Mchedlidze, Martin Nöllenburg, and Ignaz Rutter. Extending convex partial drawings of graphs. *Algorithmica*, 76(1):47–67, 2015.

[20] Maurizio Patrignani. On extending a partial straight-line drawing. *International Journal of Foundations of Computer Science*, 17(5):1061–1070, 2006.

[21] Adrian Riskin. The crossing number of a cubic plane polyhedral map plus an edge. *Studia Scientiarum Mathematicarum Hungarica*, 31(4):405–414, 1996.

[22] Marcus Schaefer. *Crossing numbers of graphs*. CRC Press, 2018.