Università degli Studi di Roma Tre
Dipartimento di Informatica e Automazione
Via della Vasca Navale, 79 – 00146 Roma, Italy

On Extending a Partial Straight-Line Drawing

Maurizio Patrignani

RT-DIA-99-2005 **July 2005**

Dipartimento di Informatica e Automazione,
Università di Roma Tre,
Rome, Italy.
patrigna@dia.uniroma3.it

Work partially supported by European Commission - Fet Open project DELIS - Dynamically Evolving Large Scale Information Systems - Contract no 001907, by "Project ALGO-NEXT: Algorithms for the Next Generation Internet and Web: Methodologies, Design, and Experiments", MIUR Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale, and by "The Multichannel Adaptive Information Systems (MAIS) Project", MIUR–FIRB.

ABSTRACT

In this paper we investigate the computational complexity of the following problem. Given a planar graph in which some vertices have already been placed in the plane (i.e., a *partial embedding*), place the remaining vertices to form a planar straight-line drawing of the graph. We show that the above extensibility problem, mentioned in the 2003 paper "Selected Open Problems in Graph Drawing" [1], is NP-complete.

1 Introduction

A *(simple) graph* $G(V, E)$ consists of a set V of *vertices* and a set E of vertex pairs called *edges*. A *drawing* of G is a mapping of each vertex $v \in V$ to a distinct point of the plane and of each edge $e \in E$ to a Jordan curve connecting its end-vertices. A drawing of G is *planar* if no pair of edges intersect except, possibly, at common end-vertices. A graph G is *planar* if it admits a planar drawing. A *straight-line drawing* of G is a drawing of G where each edge is mapped to a straight segment. Every planar graph admits a straight-line drawing, as independently established by Steinitz and Rademacher [8], Wagner [9], Fary [4], and Stein [7], and such a drawing can be computed in linear time.

In this paper we show that finding a straight-line planar drawing for a graph that is already partially drawn is an NP-complete problem. This extensibility problem was proposed in [1] and thought to be related to the problem of drawing with fixed vertex positions, a problem that was solved by Cabello [2].

Formally, the Partial Drawing Extensibility problem can be stated as follows.

Problem: **Partial Drawing Extensibility (PDE)**
Instance: A planar graph $G(V, E)$ and a mapping ψ between a subset V' of its vertices and a set of distinct points of the plane.
Question: Can coordinates be assigned to the vertices in $V - V'$ such that the resulting straight-line drawing of $G(V, E)$ is planar?

In Section 2 we show that the PDE problem is NP-hard, while in Section 3 we prove that it is in NP. Therefore, the following theorem holds:

Theorem 1 *The* Partial Drawing Extensibility *problem is NP-complete.*

Section 4 contains our conclusions and open problems.

2 Partial Drawing Extensibility is NP-hard

In order to show the NP-hardness of the PDE problem we produce a reduction from the Planar 3-Satisfiability (P3SAT) problem, which is strongly NP-complete [6]. P3SAT is defined as follows:

Problem: **Planar 3-Satisfiability (P3SAT)**
Instance: A set of clauses $C_1, \ldots, C_m$ each one having three literals from a set of Boolean variables $v_1, \ldots, v_n$. A plane bipartite graph $G(V_A, V_B, E)$ where nodes in V_A correspond to the variables while nodes in V_B correspond to the clauses (hence, $|V_A| = n$ and $|V_B| = m$). Edges connect clauses to the variables of the literals they contain. Moreover, $G(V_A, V_B, E)$ is drawn without intersections on a rectangular grid of polynomial size in such a way that nodes in V_A are arranged in a horizontal line that is not crossed by any edge (see Fig. 1).
Question: Can truth values be assigned to the variables $v_1, \ldots, v_n$ such that each clause has a true literal?

First, we describe how the PDE instance is constructed starting from a P3SAT instance (Section 2.1). Second, we show the correctness of the construction and produce the NP-hardness proof (Section 2.2).

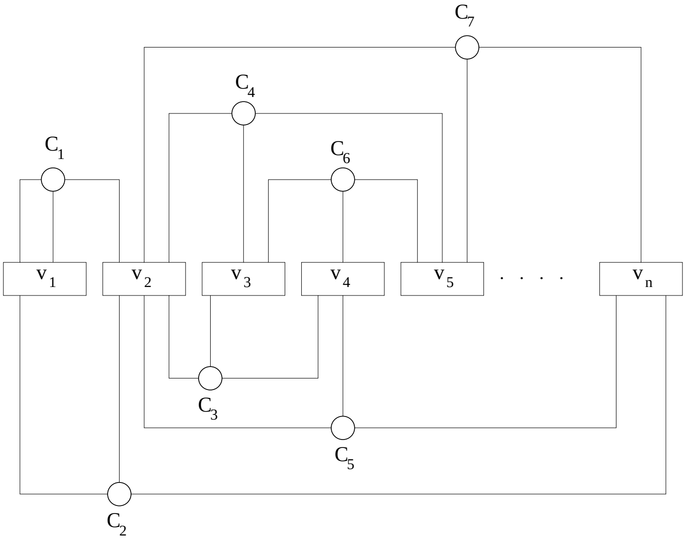

Figure 1: A planar embedding of graph $G(V_A, V_B, E)$ for a P3SAT instance.

2.1 PDE Instance Construction Rules

In this section we describe how a PDE instance is constructed starting from an instance of the P3SAT problem, consisting of the set of clauses $C_1, \ldots, C_m$, each one having three literals from the Boolean variables $v_1, \ldots, v_n$, and a drawing of the graph $G(V_A, V_B, E)$.

We briefly introduce some further definitions. Let $\langle G(V, E), \psi \rangle$ be an instance of the PDE problem and let V' be the domain of ψ. We call *fixed* vertices those in V', i.e., those that have assigned coordinates, and we call *free* vertices those in $V - V'$, whose coordinates have to be chosen in order to obtain a planar straight-line drawing of $G(V, E)$. The fixed vertices and the edges between them are introduced into the PDE instance to account for variables and clauses of the P3SAT instance as described in Sections 2.1.2 and 2.1.3, respectively, while Section 2.1.4 describes how to add the free vertices and the remaining edges. Section 2.1.1 introduces the basic tools that are used for the construction of the two kinds of gadgets.

2.1.1 Basic construction tools

For the construction of the PDE instance we make use of the *basic gadget* depicted in Fig. 2. The basic gadget only has fixed vertices, which form the boundary of a chamber. The chamber has two openings on the bottom side, called *true gate* and *false gate*, respectively, and labeled with a ‘T’ and an ‘F’ in Fig 2. On the top side the chamber has an even number of openings, that we call *exits.* The vertices and edges near the exits form narrow corridors pointing internally towards one of the two gates, and are called *true exits* or *false exits* depending on which gate they point to.

It is trivial to prove the following property:

Property 1 *Given the basic gadget depicted in Fig. 2, it is possible for a path p to traverse the basic gadget from one gate to one exit while leaving only two vertices inside the chamber only if p enters a true (false) gate and exits a true (false) exit.*

In particular, the two vertices of p must be placed in the spots where dashed circles are drawn in Fig. 2.

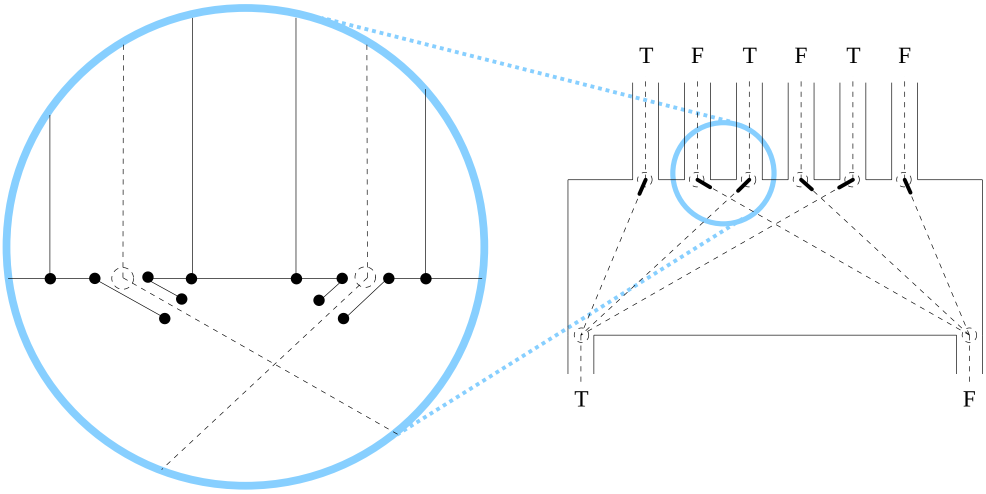

Figure 2: The basic gadget used to construct the instance of the PDE problem.

2.1.2 Variable gadget construction

For each variable v_i of the P3SAT instance we build a *variable gadget* depicted in Fig. 3. The variable gadget is composed by two basic gadgets, one of which mirrored with respect to the horizontal axis. The two basic gadgets are glued together in such a way that their true gates and false gates are attached together. The number of the exits of the top (bottom) basic gadget is equal to two times the number of the edges of E that are incident to the node of V_A corresponding to v_i from above (below) in the planar drawing of $G(V_A, V_B, E)$. Also, the small corridors near the exits point alternatively to the true and to the false gate of each gadget.

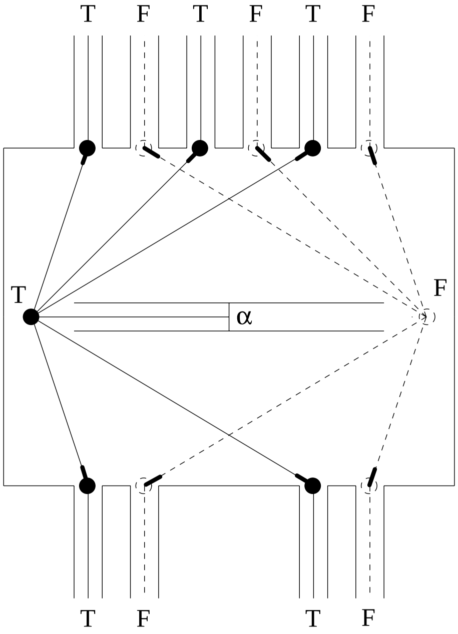

Figure 3: The variable gadget for a variable which is attached to three clauses from above and to two clauses from below.

2.1.3 Clause gadget construction

For each clause $C_h = (l_1 \vee l_2 \vee l_3)$, where l_1, l_2, and l_3 are literals of the variables v_{l_1}, v_{l_2}, and v_{l_3}, respectively, we build a *clause gadget* by using three basic gadgets as depicted in Fig. 4. Each basic gadget corresponds to a literal l_i and is attached to a true and a false exit of the variable gadget for v_{l_i} with two "pipes", called the *true* and *false* pipe, respectively, each one bending two times before reaching the variable gadget. Also, the exits of the three basic gadgets point to the same eight points $p_1, \ldots, p_8$, while, internally, the small corridors near the exits of the chambers point to the true gate or the false gate in such a way that each point $p_1, \ldots, p_8$ corresponds to a different combination of the truth values of the basic gadget exits. Further, consider the truth assignment for v_{l_1}, v_{l_2}, and v_{l_3} that does not satisfy the clause and the point p_{false} corresponding to it. The corridors pointing to p_{false} are closed with an edge.

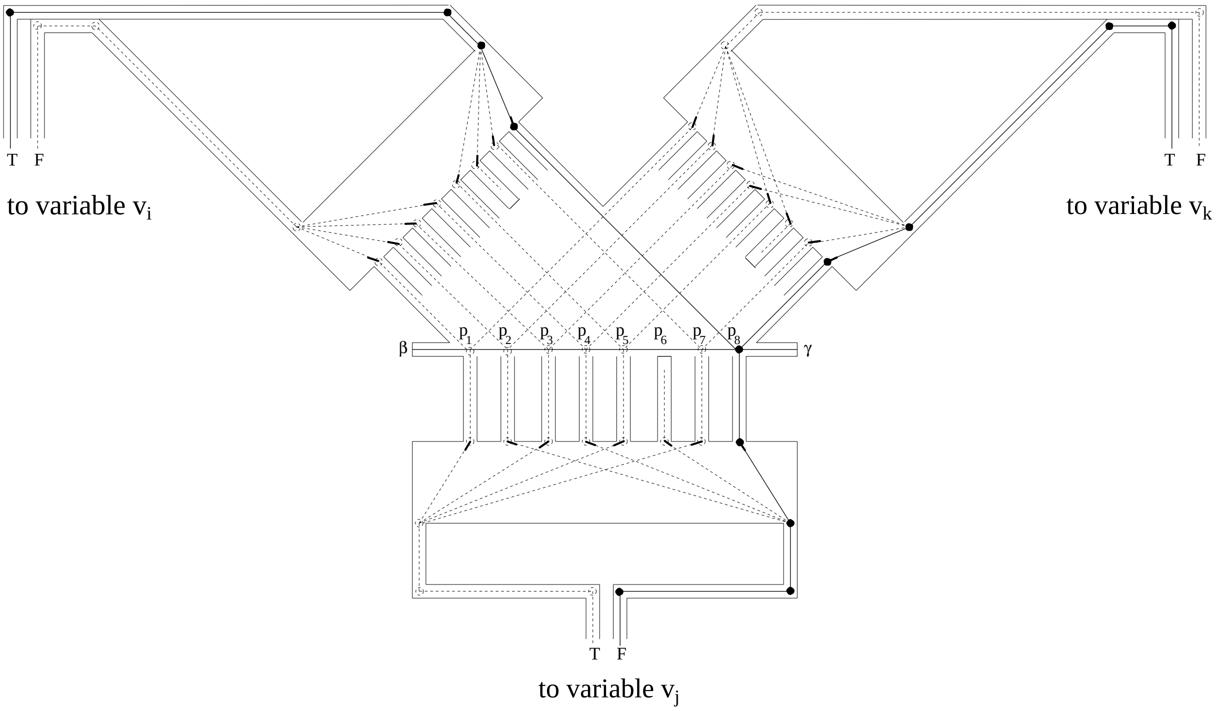

Figure 4: The clause gadget for a clause $C_h = (\overline{v}_i \vee v_j \vee v_k)$.

2.1.4 Adding free vertices

The free vertices of the PDE instance, i.e., those vertices that need to be placed while preserving planarity, are the following. For each variable v_i, we introduce one free vertex $n_{i,\alpha}$ which is adjacent to the fixed vertex of the variable gadget of v_i labeled α in Fig. 3. For each clause C_h, we introduce one free vertex $n_{h,\beta,\gamma}$ which is adjacent to the two fixed vertices of the clause gadget corresponding to C_h labeled β and γ in Fig. 4. If one literal of variable v_i occurs in clause C_h, vertices $n_{i,\alpha}$ and $n_{h,\beta,\gamma}$ are joined with a path of six edges, that is, containing other five free vertices (see Fig. 5).

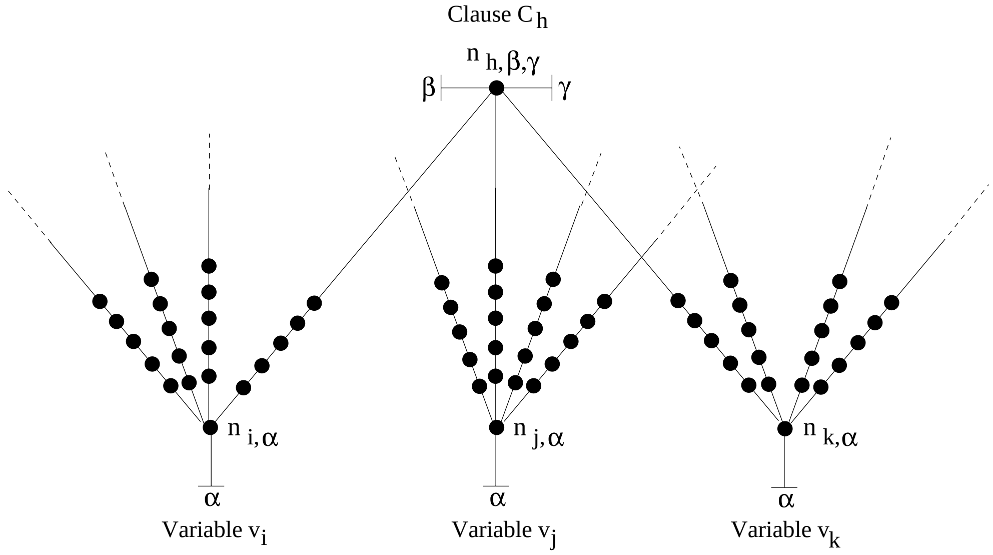

Figure 5: The free vertices of the EE instance introduced for a clause C_h with literals of the three variables v_i, v_j, and v_k.

2.2 Correctness

In this section we show the correctness of the PDE instance construction and prove the NP-hardness of the PDE problem.

Lemma 1 *Given a* P3SAT *instance, the construction of the corresponding* PDE *instance described in Section 2.1 can be performed in polynomial (actually linear) time.*

Proof: Let m be the number of clauses and n the number of variables of the P3SAT instance. The PDE instance has n variable gadgets and m clause gadgets. All clause gadgets have the same number of fixed vertices. The number of fixed vertices composing a variable gadget depends on the number of clauses which the corresponding variable participates in. However, since each clause has three literals, the overall number of true (false) exits of the variable gadgets is $3m$. Therefore, the number of fixed vertices of the PDE instance is linear in the size of the P3SAT instance. Finally, from Fig. 5 the number of free vertices is equal to $m + n + m(5 \times 3) = 16m + n$. □

Lemma 2 *If a* P3SAT *instance admits a truth assignment such that each clause has a true literal, the corresponding* PDE *instance admits a planar drawing.*

Proof: Starting from the truth assignmen of the P3SAT instance, a straight-line drawing of the corresponding PDE instance can be found as follows. Depending on the truth value of variable v_i place vertex $n_{i,\alpha}$ of the variable gadget for v_i on the true gate or false gate, and let each path to a vertex $n_{h,\beta,\gamma}$ exit from the corresponding true or false exit leading to clause gadget C_h. For each clause C_h with literals l_1, l_2, and l_3, place vertex $n_{h,\beta,\gamma}$ on an arbitrary point p_{true} different from p_{false}, and let each path to $n_{i,\alpha}$ pass through the (unblocked) corridor of the corresponding basic gadget. The obtained straight-line drawing is planar. In fact, edges between fixed vertices do not intersect. Also, due to Property 1, if v_i is true (false), for each clause C_h containing a literal of v_i, the five free vertices between each $n_{i,\alpha}$ and $n_{h,\beta,\gamma}$ can be placed inside the variable gadget for

v_i, the clause gadget for C_h, and the true (false) pipe linking them, without introducing intersections. □

Lemma 3 *A* P3SAT *instance admits a truth assignment such that each clause has a true literal if the corresponding* PDE *instance admits a planar drawing.*

Proof: Suppose that the free vertices of the PDE instance can be placed in such a way that the resulting straight-line drawing is planar. A truth assignment for the P3SAT instance such that each clause has a true literal can be easily computed as follows. Assign to each variable v_i a true (false) value if the corresponding variable gadget has the vertex $n_{i,\alpha}$ near the true (false) gate. We claim that the truth assignment so computed is such that each clause contains at least a true literal. In fact, consider the clause gadget of clause C_h. Since the paths attached to $n_{h,\beta,\gamma}$ have five internal vertices only, and since each pipe bends two times, the planarity of the drawing implies that $n_{h,\beta,\gamma}$ is placed on a point p_{true} different from p_{false} and that at least one of the three paths joining at $n_{h,\beta,\gamma}$ comes from a variable that has a truth assignment satisfying clause C_h. □

Theorem 2 *The* Partial Drawing Extensibility *problem is NP-hard.*

Proof: Due to Lemmas 2 and 3, a P3SAT instance admits a truth assignment such that each clause has a true literal if and only if the corresponding PDE instance admits a staight-line drawing. Since, starting from a P3SAT instance, the construction of the corresponding PDE instance can be done in polynomial time (Lemma 1), the statement follows. □

3 Partial Drawing Extensibility is in NP

To prove that the PDE problem is in NP we produce a nondeterministic Turing machine that decides it in polynomial time. Observe that the trivial Turing machine that nondeterministically assigns coordinates to the free vertices in all possible ways and then checks each obtained drawing for planarity is not guaranteed to work in polynomial time. In fact, in order to fully explore the space of the solutions, it would be needed to assign coordinates with an arbitrary precision. To guaratee polynomial time computations we need to transform the space of the solutions into a discrete set. To this aim, in Section 3.1 we introduce some basic computational geometry ingredients, which are used in Section 3.2 to construct a nondeterministic Turing machine that decides the PDE problem.

3.1 Equivalent Vertex Placements

In order to limit the exploration of the solution space, we aim at determining those vertex placements that are "equivalent", in the sense that one of them is representative of the whole set with respect to the possibility of finding a solution.

Consider a set V of vertices with assigned coordinates and the edges between them. Each point p of the plane that does not overlap with such vertices and edges can be associated with the set of vertices that are *visible* from p, that is, that could be joined to p with a straight segment without intersecting any other vertex or edge. Analogously, points overlapping with vertices in V and their edges can be associated with the empty set.

Two points have the same *visibility* if the sets of vertices that are visible from them are equal. A vertex w, adjacent to vertices w_1, w_2, $\dots$, $w_k \in V$ can be added to the drawing without introducing intersections if and only if it can be identified a point $p = (x, y)$ from which vertices w_1, w_2, $\dots$, w_k are visible. Fig. 6.a and 6.b show three vertices on the plane and the partition of the plane induced by the visibility relation.

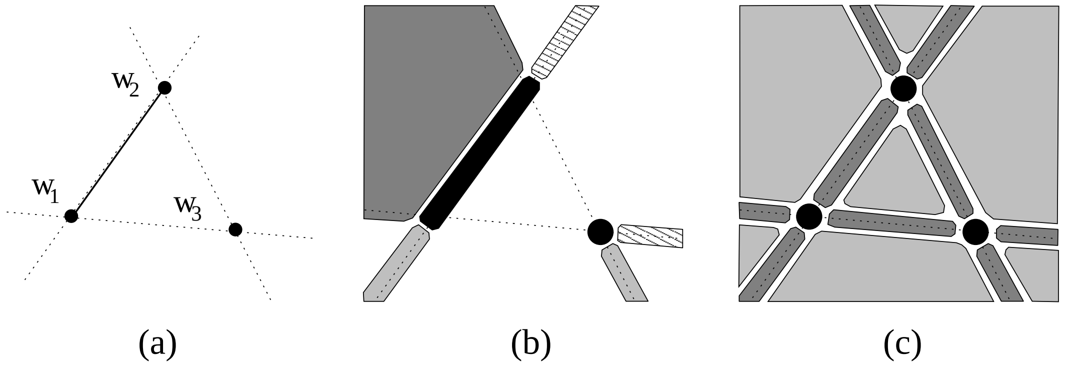

Figure 6: (a) A set of three vertices $V = \{w_1, w_2, w_3\}$ on the plane, where w_1 and w_2 are linked by an edge. (b) The partition of the plane induced by the visibility relationship: black areas have visibility $\emptyset$; white areas have visibility $\{w_1, w_2, w_3\}$; the dark gray area has visibility $\{w_1, w_2\}$; light gray areas have visibility $\{w_1, w_3\}$; and areas filled with stipes have visibility $\{w_2, w_3\}$. (c) faces of dimension two (light gray), one (dark gray), and zero (black) of arrangement $\mathcal{A}(V)$.

Given two (distinct) vertices w_1 and w_2 on the plane, we denote by l_{w_1,w_2} their common straight line. Let V be a set of vertices on the plane, their *arrangement*, denoted by $\mathcal{A}(V)$, is the set of lines l_{w_i,w_j}, one for each pair of distinct vertices w_i and w_j of V. Without loss of generality, we may assume that no line in $\mathcal{A}(V)$ is vertical, that is, each line l_{w_i,w_j} is composed by the points $p = (x, y)$ that satisfy $y = ax+b$. We say that a point $p_1 = (x_1, y_1)$ is *above*, *on*, or *below* line l if its y coordinate is greater than, equal to, or less than $ax + b$, respectively. For each line l in $\mathcal{A}(V)$, a point p_1 may be associated with a label in $\{+, 0, -\}$ depending on whether p_1 is above, on, or below l. The *position vector* of p_1 is a string in $\{+, 0, -\}^{|\mathcal{A}(V)|}$ that specifies the labels of p_1 with respect to each line of $\mathcal{A}(V)$. Two points are *equivalent* if they have the same position vector. The equivalence classes thus defined are called the *faces of* arrangement $\mathcal{A}(V)$. Notice that faces may have dimension two, one, or zero, and that they subdivide the plane into convex regions (without their boudaries), segments and half lines (without their end-points) and points [3]. Also notice that, apart from the case in which all vertices in V are collinear, each vertex corresponds to a face of dimension zero of the arrangement. We say that an arrangement $\mathcal{A}(V)$ where all vertices in V are collinear is *degenerate*. Degenerate arrangements are addressed separately at the end of the section. Fig. 6.c provides an example of the faces of an arrangement of three vertices.

Given a non-degenerate arrangement $\mathcal{A}(V)$, a data structure which represents each face of $\mathcal{A}(V)$ and each incidence between two faces explicitely can be constructed in $O(|V|^4)$ time by using, for example, the algorithm described in [3].

The following lemma holds:

Lemma 4 *The points of the same face of a non-degenerate arrangemnt $\mathcal{A}(V)$ have the same visibility.*

Proof: The statement is proved by showing that if two points have the same position vectors (that is, belong to the same face) they have the same visibility. Suppose by contradiction that two points p_1 and p_2 have the same position vector but have not the same visibility.

First, consider the case in which p_1 and p_2 belong to the same bidimensional face. Assume, without loss of generality, that vertex w_1 is not visible from p_1 but it is visible from p_2. This implies that segment $\overline{p_1 w_1}$ intersect one vertex or edge of the drawing, while segment $\overline{p_2 w_1}$ does not. Since faces are convex, the points of the segment $\overline{p_1 p_2}$ belong to f. Moving from p_1 to p_2 along the segment $\overline{p_1 p_2}$ consider the last point p_l such that $\overline{p_l w_1}$ intersects some objects of the drawing. Among such objects there is necessarily a vertex $w_2 \in V$. Thus, vertex p_2 is in one half-plane of the line l_{w_1,w_2} of $\mathcal{A}(V)$, while vertex p_1 is in the other half-plane (or on l_{w_1,w_2} if $p_l = p_1$), contradicting the fact that p_1 and p_2 have the same position vector.

Now consider the case in which p_1 and p_2 are two internal points of the same segment or half line of arrangement $\mathcal{A}(V)$. Without loss of generality, assume that p_1 and p_2 lie on line l_{w_1,w_2}, where w_1 and w_2 are two vertices of V. Segment $\overline{p_1 p_2}$ does not contain any vertex $w_3 \in V$, otherwise, since vertices in V are not all collinear, there would be a line l_{w_3,w_4}, with w_4 not belonging to l_{w_1,w_2}, such that p_1 and p_2 lie on the two opposite half-planes determined by l_{w_3,w_4}, contradicting the fact that they have the same position vector. It follows that p_1 and p_2 have the same visibility of the vertices of V that lie on line l_{w_1,w_2}. Therefore, we can identify a vertex $w_5 \notin l_{w_1,w_2}$ which is visible from p_2 but not from p_1 and apply the same technique used for points of a bidimensional face, that is, move on segment $\overline{p_1 p_2}$ until we find a point p_l between p_1 and p_2 that necessarily lies on a line l_{w_5,w_6}, contradicting the hypotheses. □

Due to Lemma 4, the faces of the arrangement are contained into the regions of the plane with the same visibility (see Fig. 6.c and Fig. 6.d for an example). Also, based on Lemma 4, each face of a non-degenerate arrangement $\mathcal{A}(V')$ can be equipped with the set of vertices visibile from that face (which is empty if that face is a vertex or an edge).

Two non-degenerate arrangements $\mathcal{A}'(V)$ and $\mathcal{A}''(V)$ are said to be *(combinatorially) equivalent* if there is a one-to-one correspondence between their faces that preserves position vectors and vertex placements. Observe that, since points of the same face have the same visibility (Lemma 4) and vertex placements are preserved by the correspondence, the corresponding faces of two equivalent arrangements have also the same visibility.

In order to determine those vertex placements that produce equivalent arrangements, we need to further subdivide the faces of an arrangement into smaller regions, that we call *cells*. The cells of arrangement $\mathcal{A}(V)$ are the faces of the arrangment $\mathcal{A}(V^*)$, where V^* is the set of faces of dimension zero of $\mathcal{A}(V)$. Since V^* contains V, faces of $\mathcal{A}(V^*)$ (i.e., cells of $\mathcal{A}(V)$) are contained into faces of $\mathcal{A}(V)$. Also, since $\mathcal{A}(V)$ has $O(|V|^2)$ lines, and since k lines may have $O(k^2)$ intersections, the cardinality of V^* is $O(|V|^4)$, which implies that $\mathcal{A}(V^*)$ has $O(|V|^8)$ lines and thus $O(|V|^{16})$ faces (cells of $\mathcal{A}(V)$).

The following lemma holds:

Lemma 5 *Let $\mathcal{A}(V)$ be a non-degenerate arrangement. All the arrangements obtained by adding a vertex w and placing it into the same cell c are equivalent.*

Proof: The statement is trivial if cell c has dimension zero (it has a single point). Otherwise, consider the two arrangements $\mathcal{A}_1(V \cup \{w\})$ and $\mathcal{A}_2(V \cup \{w\})$ obtained by adding a vertex w and placing it on point p_1 or p_2 of the same cell c. Arrangements $\mathcal{A}_1(V \cup \{w\})$ and $\mathcal{A}_2(V \cup \{w\})$ are obtained from $\mathcal{A}(V)$ by adding a line l_{w,w_i} for each $w_i \in V$. Since p_1 and p_2 belong to the same cell, lines l_{p_1,w_i} and l_{p_2,w_i} intersect the same set of faces of dimension two of arrangement $\mathcal{A}(V)$, and thus the obtained arrangements are equivalent. □

Lemma 6 *Let $\langle G(V,E), \psi\rangle$ be a* PDE *instance, V' be the domain of ψ, and w be a vertex in $V - V'$. All the* PDE *instances obtained by $\langle G(V,E), \psi\rangle$ by assigning w to the points of a cell c of $\mathcal{A}(V')$ admit a solution or they all do not.*

Proof: The statement is trivial if cell c has dimension zero. Otherwise, the statement is proved by showing that placing vertex w on point p_1 or p_2 of c, yields two arrangements $\mathcal{A}_1(V \cup \{w\})$ and $\mathcal{A}_2(V \cup \{w\})$ which are equivalent (by Lemma 5) and whose corresponding faces have the same visibility. □

Now we consider degenerate arrangements.

Lemma 7 *Let $I = \langle G(V,E), \psi\rangle$ be a* PDE *instance, and let V' be the domain of ψ. Suppose that vertices of V' all lie on the same line l. Instance I admits a solution if and only if at least one among the $|V - V'|$ instances I_w obtained by assigning a vertex $w \in V - V'$ to an arbitrary point $p_w \notin l$ does.*

Proof: Suppose I admits a solution. If all vertices of V lie on l, then I also admits a solution where an arbitrary vertex w is moved to an arbitrary point $p_w \notin l$, and the statement is trivially true. Otherwise, the solution has a vertex, call it w, which is placed in $p'_w \notin l$. Consider the position p_w of w in instance I_w. It's easy to find a transformation of the plane such that (i) points of l correspond to themselves, (ii) point p'_w is moved to p_w, and (iii) straight lines are preserved. This transformation yields a solution for I_w.

Conversely, suppose that I_w admits a solution for one $w \in V - V'$. This solution is also a solution for instance I. □

3.2 A Nondeterministic Turing Machine for PDE

The nondeterministic Turing machine T_N discribed in this section searches for a solution in a constructive way, assigning coordinates to the free vertices one at a time. At the beginning, only the fixed vertices of the PDE instance have coordinates assigned. At each step T_N assigns coordinates to a free vertex, using nondeterminism to carefully explore the solution space and taking advantage of the equivalent vertex placements described in Section 3.1.

Namely, given an instance $I = \langle G(V,E), \psi\rangle$ of the PDE problem, with V' domain of ψ, the Turing machine T_N first checks if nodes in V' lie all on the same line l. If they do, T_N nondeterministically places each vertex $w \in V - V'$ on an arbitrary point $p_w \notin l$. Therefore, in the remaining part of the computation, it may be assumed that points with assigned coordinates are not all collinear. Also, due to Lemma 7, it is assured that at least one branch will lead to a solution if and only if I admits one.

At each step, T_N constructs a data structure to explicitely represent the $O(|V|^{16})$ cells of arrangement $\mathcal{A}(W)$, where W is the set of vertices with assigned coordinates, and equips them with their visibility set. Then, T_N takes one vertex w in $V - W$ and nondeterministically places it into one of the $O(|V|^{16})$ cells of $\mathcal{A}(W)$. If the chosen cell has no visibility to all the vertices of W adjacent to w the branch of the computation terminates, otherwise a further nondeterministic step is perfomed.

After a suitable number of steps either all the vertices in V' have been placed without introducing intersections or all the branches of the nondeterministic computation were precociously terminated. In the first case, each non-terminated branch of the computation provides a solution to instance I. In the second case, Lemmas 6 and 7 ensure that instance I does not admit a solution.

Now we can prove the following:

Theorem 3 *The Partial Drawing Extensibility problem is in NP.*

Proof: At each step, the construction of the data structure to explicitely represent the cells of $\mathcal{A}(W)$, where W is the set of vertices with assigned coordinates, can be done in $O(|V|^{16})$ time [3]. It is trivial to show that cells can be equipped with visibility information in polynomial time. The number of steps to be perfomed is linear in the number of vertices of V. Therefore, a computation of T_N takes polynomial time. □

4 Conclusions

We showed that the Partial Drawing Extensibility problem is NP-complete. For simplicity, in the NP-hardness proof we used a reduction from the P3SAT problem producing non-connected PDE instances. We observe that it is not difficult to modify the construction in such a way that the resulting graph is connected. For example, edges can be added to connect each vertex $n_{i,\alpha}$ of the variable gadget for variable v_i to the middle point of each horizontal segment of the same variable gadget (see Fig. 7). Analogous changes performed on clause gadgets will produce a connected graph.

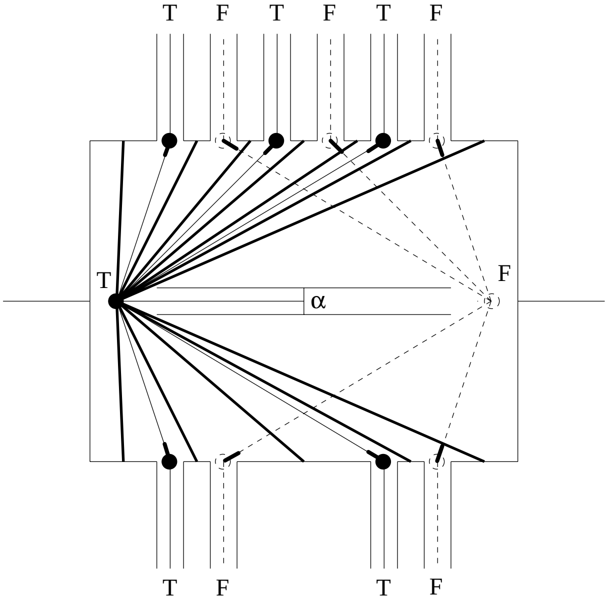

Figure 7: A variable gadget with extra edges (drawn thick) added in order to produce a connected PDE instance.

A similar problem to the one addressed in this paper comes up in mesh generation [5], where the already-placed vertices are usually assumed to form a simple polygon and the graph is assumed to have all interior faces triangles. Do these assumptions simplify the problem?

Acknowledgements

We thank the authors of [1] for their help focusing the problem and Giuseppe Di Battista for interesting conversations and constant encouragement.

References

[1] F. Brandenburg, D. Eppstein, M.T. Goodrich, S. Kobourov, G. Liotta, and P. Mutzel. Selected open problems in graph drawing. In G. Liotta, editor, *Graph Drawing (Proc. GD 2003)*, volume 2912 of *Lecture Notes Comput. Sci.*, pages 515–539. Springer-Verlag, 2004.

[2] S. Cabello. Planar embeddability of the vertices of a graph using a fixed point set is NP-hard. In *Proc. 20th European Workshop on Computational Geometry*, 2004.

[3] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science.* Springer-Verlag, Heidelberg, West Germany, 1987.

[4] I. Fary. On straight lines representation of planar graphs. *Acta Sci. Math. Szeged*, 11:229–233, 1948.

[5] L.A. Freitag and P.E. Plassman. Local optimization-based untangling algorithms for quadrilateral meshes. In *Proc. 10th int. Meshing Roundtable*, pages 397–406, 2001.

[6] D. Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11(2):329–343, 1982.

[7] S. K. Stein. Convex maps. *Proc. Amer. Math. Soc.*, 2(3):464–466, 1951.

[8] E. Steinitz and H. Rademacher. *Vorlesungen über die Theorie der Polyeder.* Julius Springer, Berlin, 1934.

[9] K. Wagner. Bemerkungen zum Vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–32, 1936.