

Simultaneous Graph Embeddings with Fixed Edges

Elisabeth Gassner², Michael Jünger³, Merijam Percan³, Marcus Schaefer¹, and Michael Schulz³

¹ DePaul University, School of CTI, 243 South Wabash, Ste 401,
60604 Chicago, IL, USA, mschaefer@cs.depaul.edu

² Technische Universität Graz, Institut für Mathematik B, Steyrergasse 30/II,
8010 Graz, Austria, gassner@opt.math.tu-graz.ac.at

³ Universität zu Köln, Institut für Informatik, Pohligstraße 1, 50969 Köln, Germany,
{mjuenger,percan,schulz}@informatik.uni-koeln.de, partially supported by Marie-Curie Research Training Network (ADONET) and by the German Science Foundation (JU204/10-1).

Abstract. We study the problem of simultaneously embedding several graphs on the same vertex set in such a way that edges common to two or more graphs are represented by the same curve. This problem is known as *simultaneously embedding graphs with fixed edges*. We show that this problem is closely related to the *weak realizability* problem: Can a graph be drawn such that all edge crossings occur in a given set of edge pairs? By exploiting this relationship we can explain why the simultaneous embedding problem is challenging, both from a computational and a combinatorial point of view.

More precisely, we prove that simultaneously embedding graphs with fixed edges is NP-complete even for three planar graphs. For two planar graphs the complexity status is still open.

1 Introduction

Cologne offers various methods of public transport, including buses, trains, and even a cable car to the zoo. Suppose your task is to design a system map displaying all of the transport systems simultaneously.

We can easily model each transport system by itself as a graph: the bus system has bus stops as vertices and edges correspond to direct connections between two stops on a bus route. Similarly, for the train system we model train stations as vertices and connections between train stations as edges. Let us make the assumption that each system by itself (bus, train, cable car, etc.) is planar; for instance, two bus routes do not intersect except at bus stops. We also assume that all graphs share the same vertex set, so a bus stop can coincide with a train stop or a cable car stop. To draw the full system map we need to embed vertices and edges in the plane such that each of the systems—bus, train, cable car—is planar by itself, but allowing, for instance, train lines and bus routes to cross outside stations.

More formally, a *simultaneous embedding* of a set of graphs sharing the same vertex set is a mapping of the vertices into the plane and a planar embedding of each graph on the vertex set [1]. Notice that edges belonging to different graphs are allowed to intersect. There are several natural ways to restrict the model; for example, we could require all edges to be drawn as straight-line segments. Such a drawing is called a *simultaneous geometric embedding* of the graphs [1].

Returning to the transport system example, it appears that asking for a simultaneous geometric embedding of the individual systems is too strict a requirement: we would be willing to accept Jordan curves to represent edges. However, requiring only a simultaneous embedding does not seem satisfactory either: if a train and bus route run in parallel, we would like to see them drawn in parallel; that is, the two routes should be represented by the same curve. This restriction leads to the notion of *simultaneous embeddings with fixed edges* introduced by Erten and Kobourov [2], and it is this notion that will mostly concern us in this paper: if several graphs share the same edge, that edge is represented by the same curve in the drawing. We see that every simultaneous geometric embedding is a simultaneous embedding with fixed edges, which, in turn, is a simultaneous embedding.

While different variants of simultaneous embeddings have been investigated from both theoretical and practical points of view [1–7], there is not much known about simultaneous embeddings with fixed edges. Erten and Kobourov [2] give an algorithm that constructs a simultaneous embedding with fixed edges of a tree and a path in linear time and uses at most one bend per edge. They point out that it is not known whether two trees can always be simultaneously embedded with fixed edges such that the number of bends is a small constant.

Studying the problem from a complexity-theoretic point of view gives a different challenge: it was not even clear whether the problem is decidable, let alone in NP or polynomial time. As we will see in Section 3 there is good reason for this: simultaneously embedding graphs with fixed edges is a different (and interesting) way to look at the weak realizability problem [9, 10]. The complexity of the weak realizability problem is related to that of the string graph problem, whose complexity was only settled recently after being open for thirty years [8, 11]. The relationship implies that the simultaneous embedding problem can be decided in NP. We settle the computational complexity question for three or more graphs in Section 3 by showing that it is NP-complete. The complexity of the case for two graphs remains open.

2 Preliminaries

Given a graph $G = (V, E)$ and a set $R \subseteq \binom{E}{2} = \{\{e, f\} \mid e, f \in E, e \neq f\}$, we call a drawing D of G in the plane in which all intersecting edge pairs belong to R a *weak realization* of (G, R) .

Notice that not all edge pairs in R are required to intersect in D , R just defines the set of allowed crossings. (G, R) is *weakly realizable* if such a drawing D exists.

Problem: *Weak Realizability*
Instance: A graph $G = (V, E)$ and a set $R \subseteq \binom{E}{2}$.
Question: Is (G, R) weakly realizable?

The weak realizability problem is closely related to the string graph problem [8]. Both problems are now known to be NP-complete even though this question was open for a long time [11].

The simultaneous embedding with fixed edges problem was introduced by Brass et al. [1]. It has become a major theoretical problem in simultaneous graph drawing. We will present an example of two planar graphs which do not have a simultaneous embedding with fixed edges. Brass et al already gave an example of two outerplanar graphs as well as three paths with no simultaneous embedding with fixed edges.

Problem: *Simultaneous Embedding with Fixed Edges (SEFE)*
Instance: A set of planar graphs $G_i = (V, E_i)$ on the same vertex set V .
Question: Are there plane drawings D_i of G_i , such that each vertex is mapped to the same point in the plane in all D_i and every shared edge $e \in E_i \cap E_j$ is represented by the same simple open Jordan curve in D_i and D_j ?

In Figure 1 two planar graphs G_1, G_2 are given for which no SEFE exist: the triangle induced by v_1, v_2, v_3 is equal to $G_1 \cap G_2$ (visualized by bold edges) and makes it impossible to include the edge $(v_4, v_5) \in G_1$ into the unique embedding (up to a homomorphism of the plane) of G_2 without crossing an edge of $G_1 \cap G_2$. This indicates that the SEFE problem is not trivial.

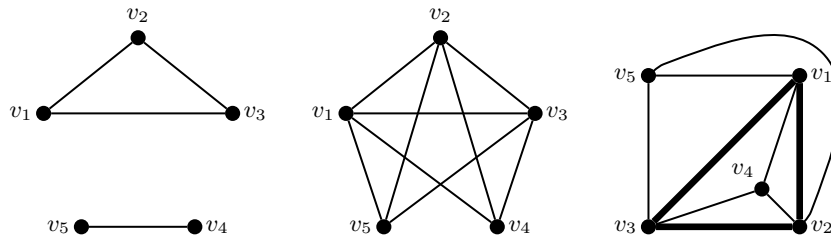


Fig. 1. Graph G_1 (left), graph G_2 (middle), unique planar embedding of G_2 (right).

3 Complexity Results on Simultaneous Embeddings

We start with a result dealing with the relationship between the two problems introduced in the previous section. This theorem implies the NP-completeness of SEFE in the general case. We will show later that the problem remains NP-complete in the case of three graphs.

Theorem 1. *The weak realizability problem is polynomially equivalent to the problem of simultaneous embedding with fixed edges.*

Proof. Let $((V, E), R)$ be an instance of the weak realizability problem. For every pair of edges $\{e, f\} \in \binom{E}{2} \setminus R$ we construct a graph $G_{e,f} = (V, \{e, f\})$. It is easy to verify that the weak realizability problem is solvable if and only if there exists a simultaneous embedding with fixed edges for the set of constructed graphs $G_{e,f}$.

Let now $G_i = (V, E_i)$, $i = 1, \dots, n$, be a set of graphs on the same vertex set V . Let $E = \bigcup_{i=1}^n E_i$ be the set of edges. Define R as the set of those pairs of edges which are not contained in one graph G_i . Then the problem of simultaneously embedding the graphs G_i is equivalent to the weak realizability problem of the pair (G, R) with $G = (V, E)$. \square

Kratochvíl [8] and Schaefer, Sedgwick and Štefankovič [11] showed that weak realizability is NP-complete. Hence we get the following corollary.

Corollary 1. *Simultaneous embedding with fixed edges is NP-complete.*

In particular, SEFE lies in NP for every fixed number k of graphs. As we will show next, it is even NP-complete in that case, as long as $k \geq 3$.

Theorem 2. *Deciding whether three graphs have a simultaneous embedding with fixed edges is NP-complete.*

Proof. We have already seen that SEFE for a fixed number of graphs is in NP. We will now show that there exists a polynomial transformation from 3SAT (which is well-known to be NP-complete) to SEFE for three planar graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$ and $G_3 = (V, E_3)$. Given an instance of 3SAT, we will construct an instance (G_1, G_2, G_3) of SEFE. Then we will prove that the instance of 3SAT is satisfiable if and only if there exists a simultaneous embedding with fixed edges of (G_1, G_2, G_3) .

The decision problem 3SAT is given in the following way:

Problem: 3SAT
Instance: A set U of boolean variables and a collection C of clauses in conjunctive normal form over U such that each clause $c \in C$ has exactly three literals.
Question: Is there a truth assignment to U that satisfies C ?

Construction: Let $C = \{c_1, \dots, c_m\}$ be the set of clauses and let $U = \{u_1, \dots, u_n\}$ be the variable set of a 3SAT-instance. Each clause is of the form $c_j = l_1^j \vee l_2^j \vee l_3^j$ with literals either $l_i^j = u_h$ or $l_i^j = \bar{u}_h$ for some $h = 1, \dots, n$ and $i = 1, 2, 3$.

Our construction of a SEFE-instance is made up of several components. For our construction we assume an ordering of the clauses, say (c_1, c_2, \dots, c_m) . Furthermore we choose an order of the three literals in each clause c_j and hence get an order of all literals in the following way $(l_1^1, l_2^1, l_3^1, l_1^2, \dots, l_3^m)$.

For each clause c_j we define a *clause box* by introducing vertices r_1^j, \dots, r_6^j , $y^{1,j}, y^{2,j}, y^{3,j}$. These vertices are connected by edges in E_2 and E_3 to form the cycle $(r_1^j, \dots, r_6^j, y^{3,j}, y^{2,j}, y^{1,j})$.

Next, we introduce two global vertices R_1 and R_2 . We add an edge (R_1, R_2) which is part of all three graphs G_1 , G_2 and G_3 . Furthermore, R_1 is connected to each clause box by four edges to the vertices r_2^j, r_3^j, r_4^j and r_5^j . These edges are part of all three graphs. R_2 is connected to each clause gadget by two edges to vertices r_1^j and r_6^j in E_3 .

To make the graphs more rigid we glue together neighboring clause boxes. This is done by identifying r_2^{j+1} with r_5^j and r_1^{j+1} with r_6^j for $j = 1, \dots, m-1$. Figure 2 shows the construction so far. We remark that all edges that have been constructed so far belong (among others) to E_3 .

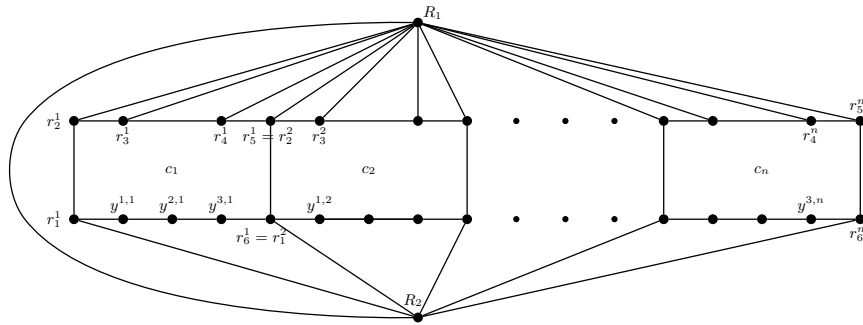


Fig. 2. The figure shows all clause boxes, the global vertices R_1 and R_2 and all connecting edges of E_3 .

For every literal l_i^j in clause c_j we define a *literal gadget* consisting of the vertices $V^{i,j} = \{x_k^{i,j}, z_k^{i,j} \mid k = 1, \dots, 6\} \cup \{y^{i,j}\}$. The edge set of the gadget is shown in Figure 3. From now on edges of G_1 are illustrated by solid lines, edges of G_2 are dashed and edges of G_3 are dotted.

Furthermore, let $l_{i(1)}^j, \dots, l_{i(\omega_h)}^j$ be all literals that belong to a variable u_h , that is $l_{i(\alpha)}^j = u_h$ or $l_{i(\alpha)}^j = \bar{u}_h$ for $\alpha = 1, \dots, \omega_h$. Assume that the literals are given in the order defined above. We connect each pair of adjacent literals in this ordered list by the following edges of E_2 (see Figure 4):

$$(z_1^{i(k),j(k)}, z_6^{i(k+1),j(k+1)}), (z_2^{i(k),j(k)}, z_5^{i(k+1),j(k+1)}), (z_3^{i(k),j(k)}, z_4^{i(k+1),j(k+1)})$$

with $k = 1, \dots, \omega_h - 1$.

For each clause we define a *clause gadget* consisting of three literal gadgets, the clause box and some additional vertices and edges. Let c_j be a clause with literals l_1^j, l_2^j and l_3^j . Notice that the three literal gadgets are already connected to the clause box via the vertices $y^{i,j}$ with $i = 1, 2, 3$. Further connections are given by the additional edges $(r_3^j, x_2^{1,j}) \in E_3$ and $(r_4^j, x_2^{3,j}) \in E_2$. We also add two vertices s^j, t^j .

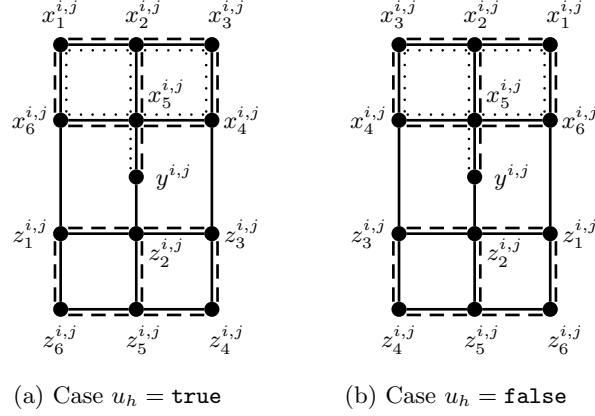


Fig. 3. Literal gadget for literal l_i^j (with corresponding variable u_h) of clause c_j . The edges in E_1 are solid, those in E_2 are dashed and those in E_3 are dotted. The two different drawings (a) and (b) will become important later.

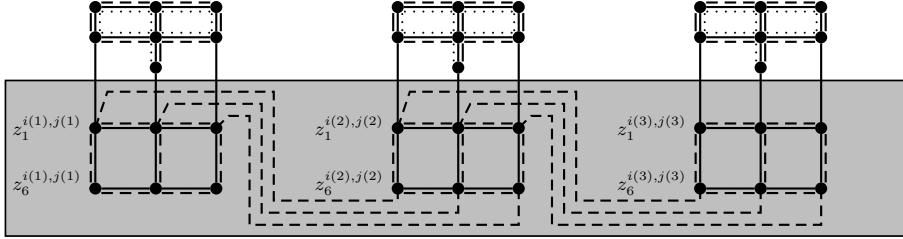


Fig. 4. Three literal gadgets that belong to variable u_h are linked with edges in E_2 .

In order to distinguish between negated and non-negated variables we define connections between the literal gadgets. The gadget of l_1^j is connected along s^j with the gadget of l_2^j . And the gadget of l_2^j is connected along t^j with the gadget of l_3^j . More precisely, we add edges $(p_1^j, s^j) \in E_3$ (where p_1^j is a vertex of the gadget of l_1^j), $(s^j, p_2^j) \in E_1 \cap E_2$ and $(q_2^j, t^j) \in E_1 \cap E_3$ (where p_2^j and q_2^j are vertices of the gadget of u_{i_2}) and $(t^j, q_3^j) \in E_2$ (where q_3^j is a vertex of the gadget of l_3^j). If l_1^j is not negated, we set $p_1^j = x_3^{1,j}$ (i. e., $x_3^{1,j}$ is connected with s^j), otherwise, if l_1^j is negated, we set $p_1^j = x_1^{1,j}$. Analogously we set $p_2^j = x_1^{2,j}$ and $q_2^j = x_3^{2,j}$ if l_2^j is not negated and $p_2^j = x_3^{2,j}$ and $q_2^j = x_1^{2,j}$ if l_2^j is negated. And finally, we set $q_3^j = x_1^{3,j}$ if l_3^j is not negated and $q_3^j = x_3^{3,j}$ if l_3^j is negated. See Figure 5 for a clause gadget.

We complete the construction by adding an edge from R_2 to $z_5^{i,j}$ to graph G_3 for each literal l_i^j .

1. Assume that the 3SAT-instance is satisfiable. We will prove that there exists a simultaneous embedding with fixed edges of the constructed instance.

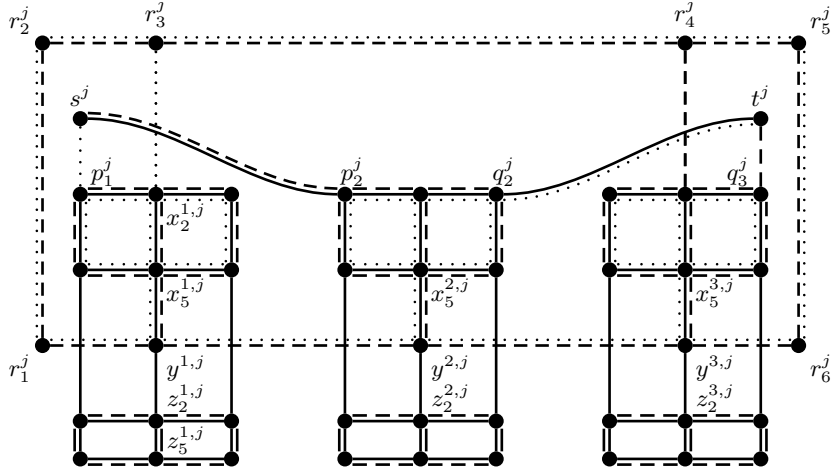


Fig. 5. Clause gadget for the case where $u_{k(2)}$ is the only variable that makes c_j true.

If variable $u_h = \text{true}$ we draw each of its literal gadgets as shown in Figure 3(a). Otherwise we draw the gadgets as shown in Figure 3(b).

The clause gadgets are drawn side by side as illustrated in Figure 2 for the clause boxes. Furthermore, the x -vertices of each literal gadget lie inside the cycle of the corresponding clause boxes and the z -vertices lie outside (see Figure 5). Moreover, every variable gets its own horizontal level for the z -variables of its literal gadgets (in Figure 4 the horizontal level is marked gray). Since all literal gadgets that belong to one variable u_h are drawn in the same way we can draw the edges that connect literal gadgets which belong to the same variable without crossings (see Figure 4).

It remains to show that we can draw the edges inside the clause gadgets without crossings of edges of the same graph. We say that a variable u makes a clause c true if either $u \in c$ and $u = \text{true}$ or if $\bar{u} \in c$ and $u = \text{false}$. Since the instance of 3SAT is satisfiable there exists at least one variable u in each clause c that makes c true.

Consider clause c_j with literals l_1^j , l_2^j , and l_3^j . Let $u_{k(i)}$ be the corresponding variable to literal l_i^j , thus either $l_i^j = u_{k(i)}$ or $l_i^j = \bar{u}_{k(i)}$. Assume first that $u_{k(1)}$ makes c_j true. Then there are two possibilities. Either $u_{k(1)} \in c_j$ and $u_{k(1)}$ is set to **true** or $\bar{u}_{k(1)} \in c_j$ and $u_{k(1)} = \text{false}$. In the first case the literal gadget is drawn like Figure 3(a) and $p_1^j = x_3^{1,j}$ and in the other case we have the situation of Figure 3(b) and $p_1^j = x_1^{1,j}$. But in both of these cases the vertex p_1^j is the upper right vertex of literal gadget l_1^j (see Figure 6). Figure 6 shows that in the case where $u_{k(1)}$ is the only variable that makes c_j true there is a simultaneous embedding with fixed edges of the corresponding clause gadget. Simple modifications yield a simultaneous embedding for the case where $u_{k(1)}$ is not the only variable that makes c_j true. Due to symmetry an analogue drawing can be found for the case where $u_{k(3)}$ makes c_j true.

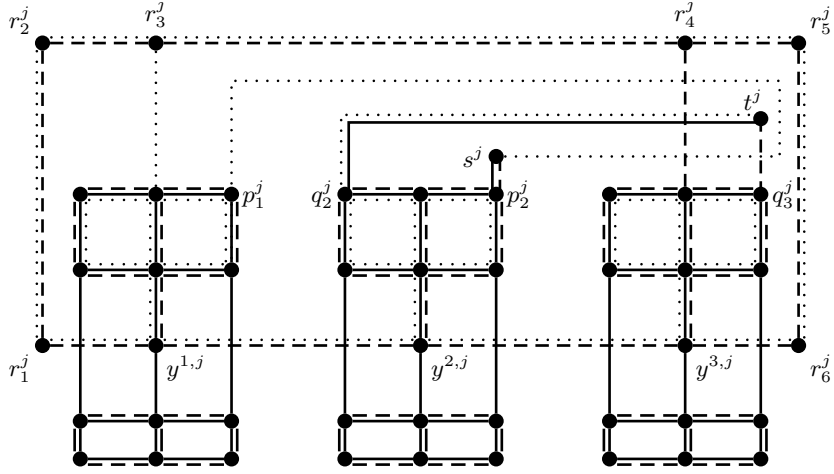


Fig. 6. Simultaneous embedding of the clause gadget when $u_{k(1)}$ is the only variable that makes c_j true.

Finally, if $u_{k(2)}$ makes c_j true we can find a simultaneous embedding as shown in Figure 5. Hence, we have found a simultaneous embedding with fixed edges of the constructed instance.

2. Now assume that we are given a simultaneous embedding with fixed edges. We will show that we can find a satisfying truth assignment for the 3SAT-instance.

Notice that the subgraph of G_3 shown in Figure 2 is a triconnected subdivision. Consequently, it has a unique combinatorial embedding. We choose the planar embedding with the edge (R_1, R_2) on the boundary of the outer face such that the cycle $(R_1, r_2^1, r_1^1, R_2, r_6^1, r_5^1)$ has the same order as visualized in Figure 2.

Due to the edge $(R_2, z_5^{i,j})$ the vertex $z_5^{i,j}$ lies outside the clause box of clause c_j for each literal l_i^j . Hence, all z -vertices of every literal lie outside all clause boxes.

Moreover, all x -vertices of a literal l_i^j lie inside the clause box of clause c_j . Notice first that both paths $(r_3^j, x_2^{1,j}, x_5^{1,j}, y^{1,j})$ and $(r_4^j, x_2^{3,j}, x_5^{3,j}, y^{3,j})$ have to be completely within the clause box of clause c_j due to vertex R_1 and its incident edges. Then all $x_k^{1,j}$ and $x_k^{3,j}$, $k = 1, \dots, 6$, have to lie within the clause box. Thus, s^j is also forced to lie within the clause box. Otherwise, the edge $(p_1^j, s^j) \in E_3$ crosses an edge of the clause box boundary which is also in E_3 . Hence, the edge (s^j, p_2^j) and then all $x_k^{2,j}$, $k = 1, \dots, 6$, have to lie within the clause box, otherwise there would be a crossing of two edges of graph G_2 .

Observe further, that each literal gadget contains (among others) two cycles, $C_1^{i,j} = \{x_k^{i,j} \mid k = 1, \dots, 6\}$ and $C_2^{i,j} = \{z_k^{i,j} \mid k = 1, \dots, 6\}$. Since the literal gadgets in G_1 are planar graphs, that are a triconnected subdivision, these gadgets have a unique planar embedding. Therefore, the clockwise ordering of the vertices of $C_1^{i,j}$ and $C_2^{i,j}$ in a simultaneous embedding is either $(x_1^{i,j}, x_2^{i,j}, \dots, x_6^{i,j})$ and

$(z_1^{i,j}, z_2^{i,j}, \dots, z_6^{i,j})$ or $(x_6^{i,j}, x_5^{i,j}, \dots, x_1^{i,j})$ and $(z_6^{i,j}, z_5^{i,j}, \dots, z_1^{i,j})$. In the first case, we say the literal gadget has a *positive* ordering and in the second case it has a *negative* ordering. Thus, the clockwise ordering of $C_1^{i,j}$ implies the clockwise ordering of $C_2^{i,j}$ and vice versa.

Again, let $l_{i(1)}^{j(1)}, \dots, l_{i(\omega_h)}^{j(\omega_h)}$ be all literals that belong to the same variable u_h . Notice that the subgraph of G_2 induced by the vertices $z_q^{i(k),j(k)}$ for $q = 1, \dots, 6$ and $k = 1, \dots, \omega_h$ is also a triconnected subdivision. Since G_2 is drawn planar, the ordering of all cycles $C_2^{i(k),j(k)}$ has to be the same. We conclude that all cycles C_1 and C_2 that belong to one variable u_h have the same ordering. Therefore, we can let

$$u_h = \begin{cases} \mathbf{true} & \text{if the literal gadgets of } u_h \text{ have a positive ordering,} \\ \mathbf{false} & \text{if the literal gadgets of } u_h \text{ have a negative ordering.} \end{cases}$$

We define the interior of cycle $C' = (r_1^j, r_2^j, r_3^j, x_2^{1,j}, x_5^{1,j}, y^{1,j})$ as region F_1^j , the interior of cycle $(r_3^j, r_4^j, x_2^{3,j}, x_5^{3,j}, y^{2,j}, y^{1,j}, x_5^{1,j}, x_2^{1,j})$ as region F_2^j and the interior of cycle $C'' = (r_4^j, r_5^j, r_6^j, y^{3,j}, x_5^{3,j}, x_2^{3,j})$ as region F_3^j (see Figure 7).

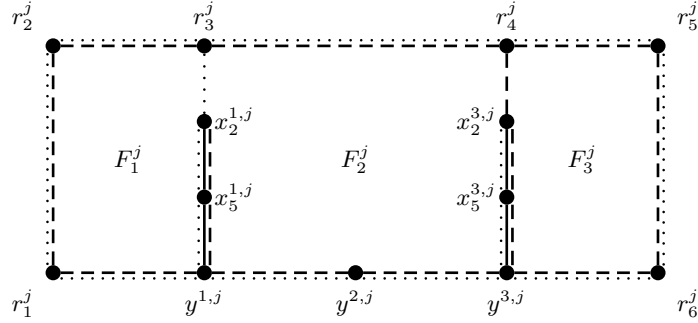


Fig. 7. p_1^j and s^j have to be embedded in region F_1^j , p_2^j and q_2^j have to be embedded in region F_2^j and q_3^j and t^j have to be embedded in region F_3^j .

Assume that there exists a clause c_j that is not true. This implies that none of the three corresponding clause variables $u_{k(1)}, u_{k(2)}, u_{k(3)}$ makes c_j true. If $u_{k(1)} \in c_j$ then the literal gadget of l_1^j has a negative ordering and $p_1^j = x_3^{1,j}$ is connected with s^j (see Figure 8). Otherwise, if $\bar{u}_{k(1)} \in c_j$ then the variable gadgets of $u_{k(1)}$ have a positive ordering and $p_1^j = x_1^{1,j}$ is connected with s^j . But in both cases the vertex p_1^j has to be embedded in region F_1^j . With similar arguments for $u_{k(2)}$ and $u_{k(3)}$ we conclude that the clause gadget of c_j is of the form illustrated in Figure 8. To be more specific, vertex q_3^j has to lie within region F_3^j and both p_2^j and q_2^j have to be embedded within region F_2^j . For the last two vertices we have an additional restriction: running through cycle $C_1^{2,j}$

the different curves as representations of the same edge. And while straight-line embeddings resolve this issue, they exclude the possibility of simultaneously embedding even very simple planar graphs, such as trees [5].

It has been known that the weak realizability problem is related to several famous problems, such as string graphs, topological inference, and Euler diagrams [9, 11], but the connection to fixed edge embeddings is the most purely combinatorial connection we have seen so far.

The main open question of complexity is the question whether two planar graphs allow a simultaneous embedding with fixed edges.

Acknowledgments

We would like to thank Dániel Marx and Stephen G. Kobourov for helpful discussions. Moreover, we are grateful to Stefan Hachul and Alejandro Estrella Balderrama for proof-reading this paper.

References

1. Peter Brass, Eowyn Cenek, Christian A. Duncan, Alon Efrat, Cesim Erten, Dan Ismailescu, Stephen G. Kobourov, Anna Lubiw, and Joseph S. B. Mitchell. On simultaneous planar graph embeddings. In *Workshop on Algorithms and Data Structures. Lecture Notes in Computer Science*, volume 2748, pages 243–255. Springer-Verlag, 2003.
2. Cesim Erten and Stephen G. Kobourov. Simultaneous embedding of planar graphs with few bends. In *12th Intl. Symp. on Graph Drawing. Lecture Notes in Computer Science*, volume 3383, pages 195–205. Springer-Verlag, 2004.
3. Cesim Erten and Stephen G. Kobourov. Simultaneous embedding of a planar graph and its dual on the grid. *Theory of Computing Systems*, 38:313–327, 2005.
4. Cesim Erten, Stephen G. Kobourov, Vu Le, and Armand Navabi. Simultaneous graph drawing: Layout algorithms and visualization schemes. In *11th Intl. Symp. on Graph Drawing. Lecture Notes in Computer Science*, volume 2912, pages 437–449. Springer-Verlag, 2003.
5. Markus Geyer, Michael Kaufmann, and Imrich Vrt'o. Two trees which are self-intersecting when drawn simultaneously. In *13th Intl. Symp. on Graph Drawing. Lecture Notes in Computer Science*, volume 3843, pages 201–210. Springer-Verlag, 2005.
6. Emilio Di Giacomo and Giuseppe Liotta. A note on simultaneous embedding of planar graphs. In *21st European Workshop on Comp.Geometry*, pages 207–210, 2005.
7. Stephen G. Kobourov and Chandan Pitta. An interactive multi-user system for simultaneous graph drawing. In *12th Intl. Symp. on Graph Drawing. Lecture Notes in Computer Science*, volume 3383, pages 492–501. Springer-Verlag, 2004.
8. Jan Kratochvíl. String graphs II Recognizing string graphs is NP-hard. *Journal of Combinatorial Theory, B* 52:67–78, 1991.
9. Jan Kratochvíl. Crossing number of abstract topological graphs. In *6th Intl. Symp. on Graph Drawing. Lecture Notes in Computer Science*, volume 1547, pages 238–245. Springer-Verlag, 1998.

10. Jan Kratochvíl, Anna Lubiw, and Jaroslav Nešetřil. Noncrossing subgraphs in topological layouts. *SIAM Journal on Discrete Mathematics*, 4(2):223–244, May 1991.
11. Marcus Schaefer, Eric Sedgwick, and Daniel Štefankovič. Recognizing string graphs in *NP*. *Journal Comput. Syst. Sci.*, 67(2):365–380, 2003.