

Chapter 13

Simultaneous Embedding



Ignaz Rutter

Abstract Given two planar graphs G_1 and G_2 that share some vertices and edges, a *simultaneous embedding with fixed edges* (SEFE) is a pair of planar topological drawings Γ_i of G_i , for $i = 1, 2$, that coincide on the shared graph $G_1 \cap G_2$. Despite much progress in the last years, the complexity of the corresponding decision problem is still open. This chapter surveys the developments in this area from the last decade. We first describe the recently discovered relations between the SEFE problem (which asks to decide whether a given pair of graphs admits a SEFE) and several other graph drawing problems, which show that SEFE is one of the most general problems in the context of planarity. Afterward, we survey algorithmic approaches to the SEFE problem, give an overview of recent results, and discuss their limitations. We close with a brief discussion of some recent variations of the simultaneous embedding problem.

13.1 Introduction

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs that potentially share some vertices and edges. We call the graph $G = G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$ the *shared graph* of G_1 and G_2 , and we typically denote its vertex set by V and its edge set by E . Given the graphs G_1 and G_2 , the problem SIMULTANEOUS EMBEDDING WITH FIXED EDGES (SEFE for short) asks whether there exists a planar topological drawing Γ_1 of G_1 and Γ_2 of G_2 such that the drawings Γ_1 and Γ_2 coincide on the shared graph G . Such a pair of drawings is called a *simultaneous planar drawing* or *simultaneous embedding with fixed edges* (SEFE for short).¹ The SEFE problem can be naturally generalized to $k > 2$ input graphs, where one requires that the topological planar

¹Note that both the drawing one seeks and the problem of deciding whether given input graphs admit such a drawing are called SEFE. This is a somewhat unfortunate double-meaning. On the other hand, the meaning is typically clear, and we follow this convention from the literature.

I. Rutter (✉)

Faculty of Computer Science and Mathematics, University of Passau, Passau, Germany
e-mail: rutter@fim.uni-passau.de

drawings Γ_i and Γ_j coincide on the graph $G_i \cap G_j$. An important special case of this is the *sunflower case*, where it is assumed that the shared graph is the same for any two different input graphs, i.e., $G_i \cap G_j$ is the same graph for each pair $i, j \in \{1, \dots, k\}, i \neq j$.

There are several other variants of the simultaneous embedding problem that take the same input, but place different requirements on the drawings. The problem SIMULTANEOUS EMBEDDING only requires that the shared vertices are mapped to the same points, whereas the same edge may be represented by different curves in the different drawings. For the problem SIMULTANEOUS GEOMETRIC EMBEDDING, the planar drawings are required to be straight-line drawings. It is well known that every pair of planar graphs admits a simultaneous embedding² [42]; so in that case the focus is mostly on the number of bends per edge that such a drawing requires. The problem SIMULTANEOUS GEOMETRIC EMBEDDING on the other hand is known to be NP-hard even for two input graphs [23]. This holds even if the combinatorial embedding of the input graphs is provided as part of the input, though in that case it is only known for $k \geq 14$ graphs [6]. We are not aware of any new results in this area over the last couple of years, and therefore refer to the survey of Bläsius et al. [14]. Further variants, such as simultaneous RAC drawings, where crossings have to occur at right angles [10, 12], and simultaneous orthogonal drawings [1] have been studied. The main focus of this chapter is the SEFE problem, and for the rest of the chapter, the terms simultaneous planar drawing or simultaneous embedding refer to a SEFE, unless explicitly stated otherwise.

The survey [14] provides a thorough overview of the results up to 2012. The purpose of this chapter is to review the recent progress on this topic and also to discuss the status of the open questions from [14].

The SEFE problem has long been motivated by applications in dynamic graph drawing, where it models the requirements that a visualization of an evolving network needs to provide both an aesthetic visualization of each individual snapshot (planarity of the individual drawings) and stability over time (coinciding with the other drawing on the shared graph). In the last years, however, SEFE has taken on an even more central role. Namely, besides the usual concept of planarity of a graph, there are several variants of it such as *level planarity* [38] (where edges have fixed y-coordinates, and edges have to be drawn as y-monotone curves), a radial variant of it [11], *partially embedded planarity*, where part of the drawing of a graph is already prescribed and the question is whether it can be completed without crossings [7], and clustered planarity [24] (where one seeks a drawing that respects a given planar clustering). As it turns out, each of these problems can be reduced to SEFE in polynomial time. Thus, a polynomial-time algorithm for SEFE would not only solve the original problem itself, but at the same time it would provide a unified planarity testing algorithm that encompasses almost all known efficiently testable planarity variants. The

²Pach and Wenger [42] show that any planar graph with a fixed combinatorial embedding can be drawn with fixed vertex positions and a linear number of bends per edge. Fixing the positions of all vertices arbitrarily at distinct points in the plane and applying the result by Pach and Wenger independently for both graphs yields the desired drawing.

relation to planarity variants along with the implications for complexity questions will be discussed in Sect. 13.2.

Afterward, we will turn toward the algorithmic problem of testing whether two graphs admit a simultaneous planar drawing. In the literature there exist essentially three approaches for showing that two graphs admit a simultaneous planar drawing. The oldest, and arguably most natural approach, is simply based on constructing the corresponding drawings. From here on two independent directions have been developed.

The first is based on so-called Hanani–Tutte characterizations, and tries to relax the properties required on simultaneous planar drawings by weakening them to finding a drawing of the union of both graphs such that the only pairs of edges that may cross an odd number of times are exclusive edges from distinct graphs. As it turns out, the existence of such a drawing can be tested algebraically; the crux is that the Hanani–Tutte characterization, which claims that if a pair of graphs admit such a drawing, then it admits a simultaneous planar drawing, has only been conjectured, but so far not proven in the general case. We will review the approach and the current state in Sect. 13.3.1.

The second approach is based on the work by Jünger and Schulz [39], who show that the existence of a SEFE is equivalent to the existence of *compatible embeddings* of the input graphs, i.e., combinatorial embeddings that induce the same combinatorial embedding on the shared graph. Here a combinatorial embedding refers to the information about the *rotation scheme*, i.e., the circular ordering of the edges around the vertices, and the *relative positions*, which describe the relative positioning of the connected components to each other by specifying which faces of the connected components are identified. Two combinatorial embeddings of the shared graph are considered to be the same if both these information coincide. This has enabled what we refer to as the *constraint-based approach*. The idea is to consider all the possible planar embeddings of the shared graph and to study the constraints that the two input graphs impose on these embeddings. A compatible embedding is precisely one that satisfies both sets of constraints. The main issue is that while the structure of all the planar embeddings of a graph is sufficiently well understood, also in the presence of constraints, there is no known data structure that allows to intersect two such sets of constraints efficiently. However, solutions have been found for increasingly general cases over the course of the last years, and we will review these and the open questions deriving from them in Sect. 13.3.2.

Finally, it is worth noting that both the constraint-based approach and the Hanani–Tutte approach will only construct compatible combinatorial embeddings of the input graphs. While, technically, this answers the question of the existence of a simultaneous planar drawing, these algorithms only provide a certificate of existence for such a drawing. Thus, the shift away from the classical drawing-based approach toward the Hanani–Tutte and the constraint-based approach has given rise to a new type of problem; constructing a simultaneous planar drawing from a pair of compatible embeddings. Here the two most prominent questions are how many bends per edge

such drawings require and the number of times a pair of edges may need to cross. We survey results of this type in Sect. 13.4 along with some recent developments in the area of dynamic graph drawing.

13.2 Relations to Planarity Variants and Complexity

The work of Schaefer [43] shows that a large part of the known variants for graph planarity for which efficient algorithms exist can (in most cases quite easily) be reduced to SEFE for two graphs. As mentioned before, this puts SEFE in the position that an efficient algorithm would not only solve a long-standing question in graph drawing, but would also provide a unified planarity testing algorithm for almost all planarity variants. The position of SEFE in this setting has been further emphasized by the connections to the well-known problem of *clustered planarity* (*c-planarity* for short), which was uncovered by Schaefer [43] and by Angelini and Da Lozzo [2]. Finally, while the complexity of SEFE for two input graphs remains open, there has been success in showing NP-hardness for some quite restricted generalizations of the problem. Together with the reductions mentioned before, over the last few years we have seen a border of tractability emerge, with several planarity variants that are efficiently testable on the one side and NP-hard problems on the other side. In the following, we will outline this border of tractability in more detail by explaining the connections to several planarity variants, discussing various recent complexity results related to SEFE, and finally sketching the reductions that establish the connection to the c-planarity problem.

13.2.1 Planarity Variants

In the following we show examples of reductions from different planarity variants to SEFE. The fact that most of these reductions are not difficult highlights the expressive power of the SEFE problem and underlines the importance of determining its complexity. Most of the following reductions are due to Schaefer [43].

Partially Embedded Planarity

Given a graph G , and a subgraph $H \subseteq G$ with a fixed planar embedding \mathcal{E}_H , the problem PARTIALLY EMBEDDED PLANARITY (PEP) is to determine whether there exists a planar embedding \mathcal{E}_G of G whose restriction to H coincides with \mathcal{E}_H . The embedding \mathcal{E}_G is then called an *embedding extension*. Figure 13.1a, b show an instance of PEP and an embedding extension.

An instance (G, H, \mathcal{E}_H) of this problem can be transformed into an equivalent instance of SEFE as follows [43]. Choose $G_1 := G$, and choose G_2 as a triangulation

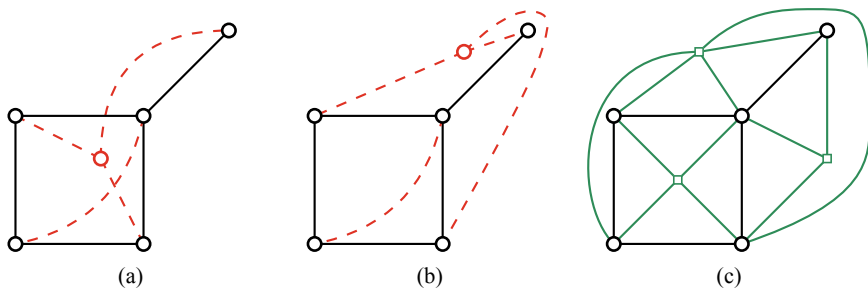


Fig. 13.1 Example instance (G, H, \mathcal{E}_H) of PARTIALLY EMBEDDED PLANARITY **(a)** and an embedding extension \mathcal{E}_G **(b)**. The graph H is drawn black, the vertices of $V(G) - V(H)$ are red, and the edges of $E(G) - E(H)$ are red and dashed. **c** Shows the triangulation G_2 constructed from H by adding vertices (green squares) and edges (green)

of H with its embedding \mathcal{E}_H that is obtained by adding vertices into the faces of H . Since we do not add any edges between vertices of H , it follows that $G_1 \cap G_2 = H$. Moreover, since G_2 is a triangulated planar graph, its combinatorial embedding on the sphere is uniquely determined up to reflection, and therefore the combinatorial embedding of H induced by a planar embedding of G_2 is either \mathcal{E}_H or its reflection. It follows that G admits an embedding that extends \mathcal{E}_H if and only if G_1 and G_2 admit a SEFE. Figure 13.1c shows the graph G_2 for the instance from Fig. 13.1a.

Theorem 1 ([43, Theorem 6.2]) *There is a linear-time reduction from PARTIALLY EMBEDDED PLANARITY to SEFE.*

Similar constructions can be made for more general types of planarity constraints. A *PQ-constraint* for a vertex v is a PQ-tree T_v that constrains the circular order of a subset of the edges incident to v to be one of the orders represented by v . Such a constraint is *total* if it constrains all the edges incident to v . The problem PARTIALLY PQ-CONSTRAINED PLANARITY asks whether a given graph G with a PQ-constraint T_v for each vertex v of G admits a planar embedding that satisfies all the constraints. Gutwenger et al. (who call such constraints *ec-constraints*) show that the problem PQ-CONSTRAINED PLANARITY, which requires that all constraints are total, can be solved in linear time [36]. Using gadgets similar to the ones by Gutwenger et al. [36] and the triangulation construction above, Schaefer [43] shows the following.

Theorem 2 ([43, Theorem 6.16]) *There is a linear-time reduction from PARTIALLY PQ-CONSTRAINED PLANARITY to SEFE.*

Bläsius and Rutter showed that PARTIALLY PQ-CONSTRAINED PLANARITY can be solved in linear time if the input graph is biconnected [16]. The general case remains open.

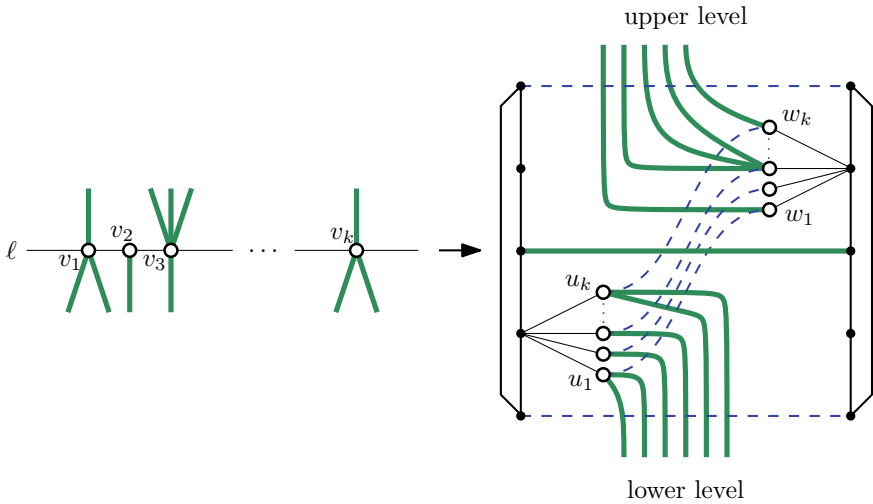


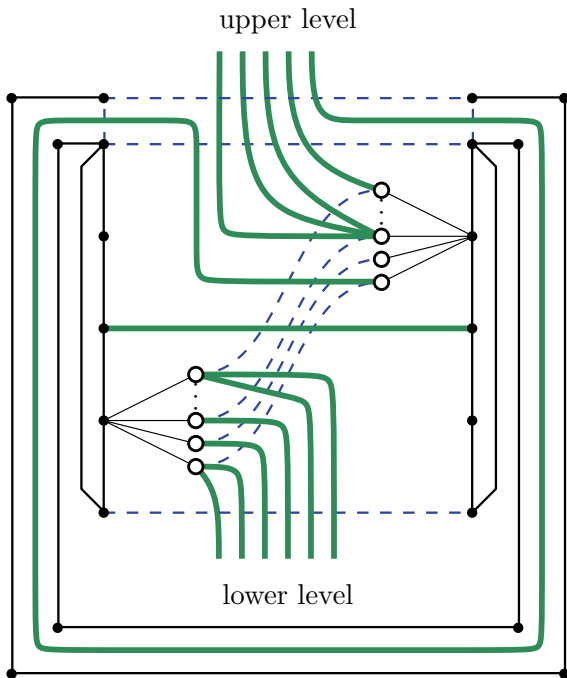
Fig. 13.2 Illustration for the reduction of level planarity to SEFE. A level ℓ of a level graph with vertices v_1, \dots, v_k and edges coming from the previous and going to the next level (left) together with the corresponding gadget (right). Edges of the shared graph are solid black, exclusive edges are dashed blue and bold green

(Radial) Level Planarity

It is well known that level planarity reduces to radial level planarity (e.g., by adding a single edge that connects a new vertex on the first level to a new vertex on the last level). Schaefer [43] describes transformations that reduce level planarity and radial level planarity to SEFE. The key idea is to model each level ℓ of the input graph that contains vertices v_1, \dots, v_k as a box-shaped gadget as in Fig. 13.2, where for each vertex v_i there are two copies u_i and w_i that are used for attaching the incoming and outgoing edges of v_i , respectively. The fact that the dashed blue edges $u_i w_i$ may not cross each other, enforces that the order of the vertices u_1, \dots, u_k in the lower half of the box is transferred to the order of the vertices w_1, \dots, w_k in the upper half of the box. As mentioned before, the incoming edges of v_i are attached to u_i and the outgoing edges to w_i . The reduction from level planarity to SEFE is formed by creating one gadget per level of the input graph and identifying the upper horizontal dashed blue edge of each gadget with the lower horizontal dashed blue edge of the gadget representing the next level. Note that the fact that the thick green edges may not cross each other, models exactly the fact that the drawing should be level planar.

Given a SEFE of the resulting instance, a level planar drawing of the original graph is obtained by taking for each level the order of the vertices u_1, \dots, u_k from top to bottom. By extending each gadget with an additional ring structure as in Fig. 13.3, one can also allow the edges leaving the gadget to “wrap around” the gadget, which models the behavior of radial levels.

Fig. 13.3 Modified gadget for the reduction of radial level planarity to SEFE



Theorem 3 ([43, Lemma 6.12, Corollary 6.13]) *There are linear-time reductions from level planarity and radial level planarity to SEFE, respectively.*

By replacing the star used to attach the vertices u_1, \dots, u_k to the rest of the frame of the gadget by a suitable construction, it is also possible to further constrain the vertex ordering on each level by an arbitrary PQ-tree.

Note that in the reduction from radial level planarity, for each level ℓ the gadgets representing the levels above and below ℓ lie in different faces of the gadget representing the level ℓ . It is thus not possible to link the last level to the first one to model level planarity on the torus. Note that this problem can be solved in polynomial time [4]. It is not entirely clear whether this problem can also be reduced to SEFE, though it seems that a construction similar to [3, arXiv version 1] can be used to modify the gadgets so that the ordering on the last level is propagated through all gadgets without interference.

13.2.2 Relation to Clustered Planarity

Similarly as above, Schaefer also gives a reduction from c-planarity to SEFE. Due to interesting and recent developments in this area, we devote its own section to the relation between SEFE and c-planarity.

An instance of c-planarity consists of a pair (G, \mathcal{T}) , where $G = (V, E)$ is a planar graph and \mathcal{T} is a rooted tree with leaf set V . The tree \mathcal{T} defines a hierarchical clustering of G by considering for each node v of \mathcal{T} the leaves of the subtree rooted at v as a cluster in G . A c-planar drawing of (G, \mathcal{T}) consists of a planar drawing of G together with a closed curve for each cluster C defined by \mathcal{T} such that (i) the curves are pairwise non-crossing, (ii) each curve encloses exactly the vertices contained in the corresponding cluster, and (iii) each edge crosses each curve at most once. Note that the curves enclosing the clusters may only cross edges but are not allowed to contain vertices. The problem c-planarity asks whether a given instance of c-planarity admits a c-planar drawing. The problem was first studied by Lengauer [40]; the name c-planarity was coined by Feng et al. [24]. The computational complexity of the c-planarity problem is open both if the combinatorial embedding of G is fixed and if it is variable. Though recently, Cortese and Patrignani [21] have shown that the general case with hierarchical clusters can be reduced to the case where \mathcal{T} has height 2, i.e., besides the trivial clusters containing only one vertex or the whole graph, there is only one partition into non-trivial clusters. Such a clustering is called *flat clustering*.

Theorem 4 [43] *There is a polynomial-time reduction from c-planarity to SEFE.*

Proof Sketch

Let (G, \mathcal{T}) be an instance of c-planarity. The key idea is to replace each cluster by a gadget as shown in Fig. 13.4. Each such gadget has two special *attachment vertices* I and O . The gadget for a cluster C consists of a *frame*, which is composed from the shared black edges and the bold exclusive edges (dashed blue and solid green). This frame is a subdivision of a 3-connected graph and therefore has a unique planar embedding. The gadget consists of three regions, the triangular *inner region* on the left, the shaded *transmission structure*, and the *outer region*, which lies outside the gadget.

The interior of the cluster is modeled by the content of the inner region. Each edge that leaves the cluster C is split into an *inner edge*, which starts in the inner region and ends in the transmission structure, and an *outer edge*, which starts in the transmission structure and ends in the outer region. In Fig. 13.4, these edges are bold solid green. The transmission structure, which is similar to the gadget used for the reduction of level planarity, synchronizes the order in which the inner and outer edges leave their regions. If the cluster consists of a single vertex, we place a single vertex inside the triangular region, connect all the inner edges to it and attach it to the inner attachment vertex I via a dashed blue edge. Otherwise, the cluster consists of several child clusters (observe that according to our definition each vertex also forms a cluster). We then position the corresponding gadgets inside the inner region and connect their outer attachments to the inner attachment I of cluster C . We further identify the inner edges of C with the outer edges of the children of C ; see Fig. 13.4, where the gadgets of the child clusters are illustrated by a scaled outline of the gadget for C . Note that the dashed blue attachment edges ensure that the gadgets

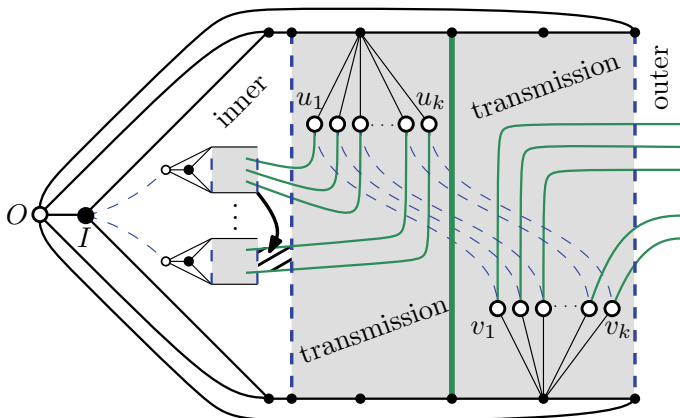


Fig. 13.4 Reduction of c-planarity to SEFE

representing the child clusters remain inside the inner region. Observe further that the order in which the inner edges enter the transmission structure is highly flexible; for example, in Fig. 13.4 by moving the upper child cluster into the tiled region as indicated by the arrow, the edges leaving these clusters can be nested. Since the inner edges may not cross each other, it is, however, not possible that the edges of two child clusters alternate.

Thus, in a SEFE of the resulting instance, we can essentially use the inner region of each cluster gadget as the cluster boundary and draw the corresponding clusters in the inside. Since the inner and outer edges are ordered consistently, they can then be joined back to form the original edges. Conversely, it is not difficult to see that, given a c-planar drawing, the gadgets can be drawn in such a way that a SEFE is obtained. \square

Interestingly, a variant of the converse also holds, though slightly stronger requirements are needed; namely the shared graph needs to be connected. We call the variant of SEFE with this input restriction **CONNECTED SEFE**. The following reduction is due to Angelini and Da Lozzo [2].

Theorem 5 *There is a polynomial-time reduction from **CONNECTED SEFE** to c-planarity.*

Proof Sketch

The construction works in several steps. First, the instance of **CONNECTED SEFE** is modified such that (1) the shared graph is a tree T and (2) only the leaves of T are incident to exclusive edges. To achieve property (1), an edge of the shared graph that lies on a cycle of the shared graph is subdivided twice, and afterward the middle segment is replaced by two parallel paths of length 3, of which the outer two

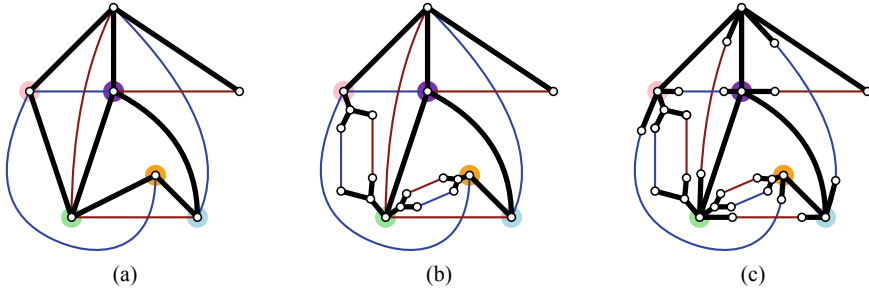


Fig. 13.5 Illustration of the first step of the reduction from `CONNECTED SEFE` to `c-planarity`

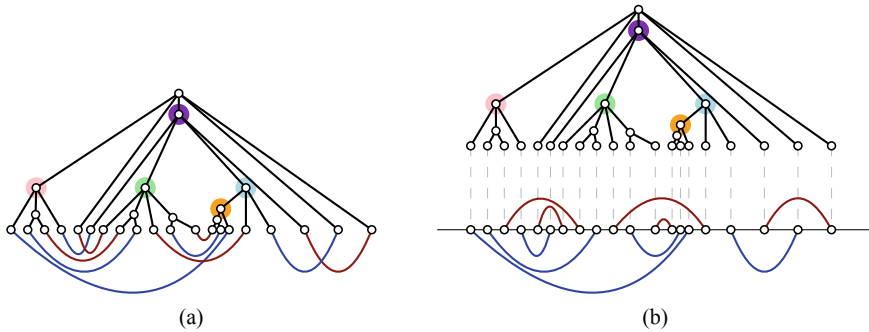


Fig. 13.6 Illustration of the second step of the reduction

edges are shared and the middle edges are exclusive and belong to different graphs; Fig. 13.5b shows the result of iteratively applying this transformation to the instance from Fig. 13.5a. Afterward, as long as there still is an exclusive edge uv such that v is not a leaf of T , subdivide uv into two edges uw, vw and declare uw a shared edge; see Fig. 13.5c. The fact that this transformation preserves the existence of a SEFE is due to Angelini et al. [8].

Observe that any SEFE of the resulting instance can be modified such that the leaves of T lie on the x -axis, the tree is drawn crossing-free in the halfplane above the x -axis, and the exclusive edges are drawn in the lower halfplane; see Fig. 13.6a. By detaching the tree from the remaining edges and flipping the red exclusive edges above the x -axis, we find that this is equivalent to a planar 2-page book embedding of the graph consisting only of the exclusive edges such that (1) the edge partition given by the book embedding coincides with the partition of the exclusive edges to the two graphs and (2) the vertex ordering along the spine is *coherent* with T , i.e., for each node v of T the vertices that correspond to the leaves of the subtree rooted at v are (circularly) consecutive along the spine. This problem is called `T-COHERENT PARTITIONED 2-PAGE BOOK EMBEDDING`. Again this transformation works both ways [8], and thus shows that this is equivalent to `CONNECTED SEFE`.

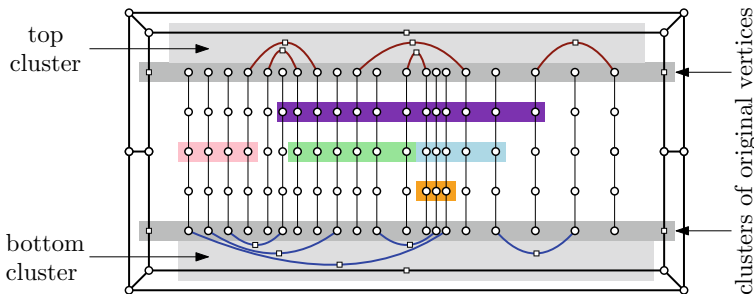


Fig. 13.7 Illustration of the third step of the reduction

Finally, and this is the novel step by Angelini and Da Lozzo [2], an instance of *T*-COHERENT PARTITIONED 2-PAGE BOOK EMBEDDING can be transformed into an equivalent instance of *c*-planarity as illustrated in Fig. 13.7.

The construction starts with a frame (thick black edges) that forms a subdivision of a 3-connected planar graph, and hence has a unique (up to reversal) combinatorial embedding on the sphere. Then, for each vertex of the spine, we create an upper and a lower copy and two clusters, one that contains all upper copies and one that contains all lower copies (dark shaded cluster). The inclusion of two subdivision vertices of the frame ensures that both clusters are attached to the frame. We further connect each pair of copies by a path that contains as many subdivision vertices as the tree *T* has levels (with respect to some arbitrarily chosen root). Altogether, this enforces that the clusters and the paths are embedded as shown in Fig. 13.7. In particular, the copies of the spine vertices must be embedded in the inner face of the frame, and the order of the upper and the lower copies given by the order in which their incident edges cross the boundary of their cluster is the same. The consecutivity constraints imposed by the inner nodes of *T* can then be modeled as clusters on the subdivision vertices of the paths between the copies of the spine vertices. Finally, a top and a bottom cluster is added (light gray). For each red exclusive edge, we create a subdivision vertex and assign it to the top cluster, and for each blue exclusive edge, we create a subdivision vertex and assign it to the bottom cluster. Since the top and bottom clusters also contain a subdivision vertex of the top and bottom of the frame, respectively, this ensures that the red and blue edges must be embedded crossing-free above and below the clusters containing the copies, respectively. Again, it is readily seen that the transformation is correct. □

13.2.3 Complexity

The above reductions show that SEFE is one of the most general planarity variants. Despite much research, the complexity question is still open. While Gassner et al. [33] have shown that SEFE is NP-hard for $k \geq 3$ input graphs, their reduction yields graphs

that have little structure. One may thus wonder whether structural assumptions allow to solve SEFE for $k \geq 3$ graphs, e.g., in the sunflower case or if the intersection graphs are particularly simple. The following recent result by Angelini, Da Lozzo and Neuwirth [5] shows hardness of SEFE for $k \geq 3$ even for highly restricted cases.

Theorem 6 ([5]) *Sunflower SEFE for $k \geq 3$ graphs is NP-complete even if the shared graph is a star.*

Proof Sketch

The reduction is from the well-known NP-hard problem BETWEENNESS [41]. Its input consists of a ground set X and a set T of triplets of distinct elements of X . The question is whether X admits a linear ordering $<$ such that for each triplet $(x, y, z) \in T$ the element y is between x and z , that is, it is $x < y < z$ or $z < y < x$.

The construction of the hardness proof is illustrated in Fig. 13.8. It consists of a frame (thick solid black and blue edges), which is a wheel graph with center vertex o and a special subdivision vertex t on one non-spine edge. Observe that the embedding of the frame is unique. For each triplet $(x, y, z) \in T$, the wheel contains three consecutive spines with end vertices a, b, c . The vertices a and c are connected to t by dotted red edges, and the vertex b is connected to t by a dashed green edge. To model the ordering constraint expressed by (x, y, z) , we create for each $w \in X$ a vertex w_ℓ and a vertex w_r , and connect all of them to o by shared edges. Moreover, we pick some fixed element $w' \in X$ and connect w'_ℓ to a and b , as well as to all vertices w_ℓ with $w \in X \setminus \{w'\}$ by blue edges (thin). This enforces that all vertices w_ℓ are embedded in the face of the frame enclosed by the cycle oab . Further, we add the five blue edges $x_r y_r, y_r z_r, c x_r, c y_r, c z_r$, which enforces that x_r, y_r and z_r are embedded inside the face of the frame bounded by obc . Also, their ordering

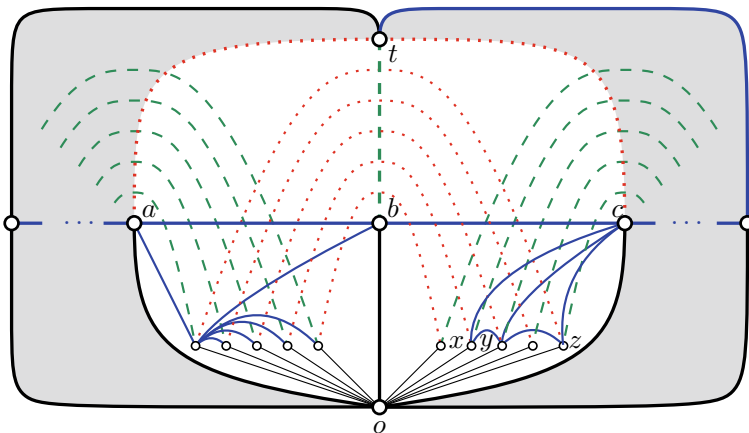


Fig. 13.8 Illustration of the hardness proof of SEFE for three graphs that share a star

around o must be such that y lies between x and z . We connect each pair of vertices w_ℓ, w_r by a red edge (dotted). This ensures that, if the vertices w_r are all embedded in the interior of the face formed by bco , then their linear orderings indeed coincide. Finally, we attach to each vertex w_ℓ a *left connector edge*, and to each vertex w_r a *right connector edge*, which belong to the green graph (green, dashed). If (x, y, z) is the first or the last triplet, we simply attach the left or the right edges to the frame. To place multiple triplets, we simply put copies of the above construction next to each other, identifying them along the boundary and also identifying their connector edges. This not only ensures that indeed all vertices w_r have to reside in the face of the frame bounded by obc , but also ensures that for each face of the frame, the ordering of the elements of X determined by the ordering of the edges ow_ℓ (or ow_r) around o is the same.

Since the construction can be performed in polynomial time, the theorem follows. \square

Angelini et al. [5] show further interesting hardness results. For example, two of the graphs in the construction above can be made 2-connected. They also show that the following natural optimization version MAX SEFE of SEFE is NP-hard, where one asks for planar drawings Γ_1, Γ_2 of G_1, G_2 , respectively, such that as many shared edges as possible are drawn identically in G_1 and in G_2 . Recently, Grilli [34] has shown that testing the existence of a SEFE, where any edge may receive at most k crossings is NP-complete for $k \geq 1$.

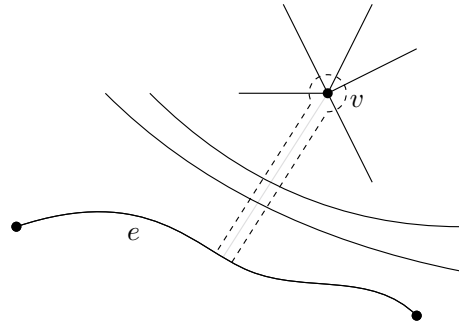
13.3 Algorithmic Approaches to Simultaneous Planarity Testing

In this section, we survey the two main directions that have been pursued for obtaining polynomial-time algorithms for testing simultaneous planarity for restricted inputs. Here, the two main contenders are the Hanani–Tutte approach, pioneered by Schaefer [43], and an approach that is based on modeling the SEFE problem as a planar embedding problem with constraints. The first instances of the latter seem to be the works by Angelini et al. [8] and Jampani et al. [37].

13.3.1 The Hanani–Tutte Approach

The work in this area is based on the Hanani–Tutte characterization of planarity, which states that a graph is planar if and only if it has a drawing where any two edges cross an even number of times [20]. The proof is based on a redrawing procedure that shows how to transform a drawing where any pair of edges crosses an even number of times into a planar drawing by redrawing edges one by one. In this setting, it can even be shown that the rotation system of the initial drawing can be preserved.

Fig. 13.9 Illustration of an (e, v) -move. The curve c used for the redrawing is gray, the redrawn part is dashed



This is known as the *weak Hanani–Tutte theorem*. There is also a strong version of the theorem, which requires only that any pair of independent edges (i.e., edges not sharing an endpoint) crosses evenly. More formally, for a drawing D of a graph G , $\text{iocr}(D)$ denotes the number of pairs of independent edges that cross an odd number of times. The independent odd crossing number $\text{iocr}(G)$ is the minimum of $\text{iocr}(D)$ over all drawings D of G . The strong Hanani–Tutte theorem states that a graph G is planar if and only if $\text{iocr}(G) = 0$. When transforming a drawing D with $\text{iocr}(D) = 0$ into an embedding, it is generally not possible to preserve the rotation system [43]. In fact, Fulek, Kynčl, and Pálvölgyi [28] show a common strengthening of the strong and the weak Hanani–Tutte theorem by proving that a drawing where any pair of independent edges crosses evenly can be made planar such that the rotation system is preserved at vertices whose incident edges cross evenly.

At first sight, this looks like a purely combinatorial tool, and it is unclear how to use it for planarity testing. However, the strong Hanani–Tutte theorem can be used to design an algebraic planarity test. The idea is to start with an arbitrary drawing of the input graph $G = (V, E)$. Now consider an edge $e \in E$ and a vertex v that is not an endpoint of e . An (e, v) -move is a modification of the drawing, which redraws the edge e around v . This is achieved by taking an arbitrary curve c from a point on the curve representing e to v and then rerouting e around c ; see Fig. 13.9. The key insight is that the only additional crossings are with edges incident to v and with edges crossed by the curve c , where the latter crossings come in pairs and do not change the crossing parity. In total, it is exactly the edges incident to v whose crossing parities with e change.

In particular, if $e = \{u, v\}$ and $f = \{x, y\}$ are two edges, then the crossing parity of e and f is changed only by moves of e around endpoints of f and vice versa. We use a variable $x_{e,v}$ over the field \mathbb{F}_2 with two elements to encode whether an (e, v) -move should be performed. The requirement that edges e and f should cross evenly after performing the moves can then be expressed by the equation

$$x_{e,x} + x_{e,y} + x_{f,u} + x_{f,v} = i_{e,f}, \quad (13.1)$$

where $i_{e,f}$ is the number of crossings between e and f modulo 2 in the initial drawing. Taking this equation for each pair of independent edges e, f yields a linear system of equations, which has a solution if and only if there is a set of (e, v) -moves that results in a drawing where any two independent edges cross evenly. By the strong Hanani–Tutte theorem this is equivalent to the existence of a planar drawing.

Schaefer [43] attempts to generalize this approach to simultaneous planar drawings by considering the *simultaneous independent odd crossing number*. For a drawing D of $G_1 \cup G_2$, we denote by $D[G_1]$ and $D[G_2]$ the subdrawings induced by G_1 and G_2 , respectively. For a pair of graphs (G_1, G_2) , Schaefer defines the *simultaneous independent odd crossing number* $\text{siocr}(D) = \text{iocr}(D[G_1]) + \text{iocr}(D[G_2])$. The *simultaneous independent odd crossing number* $\text{siocr}(G_1, G_2)$ is the minimum of $\text{siocr}(D)$ over all drawings D of $G_1 \cup G_2$. Schaefer conjectures the following.

Conjecture 1 Two graphs (G_1, G_2) admit a SEFE if and only if $\text{siocr}(G_1, G_2) = 0$.

If the conjecture holds, then an equation system similar to the one described above yields an efficient algorithm for SEFE: One starts with an arbitrary drawing and then seeks to remove odd crossings between edges of the same graph by (e, v) -moves.

So far a proof has been found only in restricted cases.

Theorem 7 ([43]) *Let G_1, G_2 be two planar graphs such that each connected component of $G_1 \cap G_2$ is biconnected or subcubic. Then G_1 and G_2 admit a SEFE if and only if $\text{siocr}(G_1, G_2) = 0$.*

The advantage of the Hanani–Tutte approach is that it comes with a simple, though somewhat inefficient algorithm (solving a system of $O(n^2)$ linear equations over \mathbb{F}_2), and all the complexity is hidden in the redrawing steps inside the proof. In fact, Schaefer shows how to make the algorithm constructive so that it actually produces a SEFE, provided the conjecture holds. Thus, we do actually have an algorithm that either produces for each input instance a solution, or it gets stuck in a counterexample to the conjecture.

Unfortunately, even though the Hanani–Tutte approach has been extended to various drawing styles [29–31], there has not been further progress on the simultaneous version. This may partially be due to a counterexample of Fulek et al. [27]; it disproves a Hanani–Tutte variant for clustered planarity, which by means of the reduction from Sect. 13.2 from clustered planarity to simultaneous planarity might yield a counterexample to Conjecture 1.

13.3.2 Constraint-Based Simultaneous Planarity Testing

The second quite successful approach to solving restricted cases of SEFE is the constraint-based approach, where one considers the restrictions that the input graphs G_1 and G_2 impose on the planar embedding of the shared graph G as constraints. The question is then, does G admit a planar embedding that satisfies all the constraints? The key ingredient is the following theorem due to Jünger and Schulz [39].

Theorem 8 ([39]) *Let G_1 and G_2 be two planar graphs with shared graph G . Then G_1 and G_2 admit a SEFE if and only if they admit compatible embeddings.*

One might argue that this is but a small step akin to passing from planar drawings to embeddings, which are equivalence classes of drawings. To illustrate the power of this theorem, we show a simple derivation of a known result.

Theorem 9 ([25]) *A planar graph and a tree always admit a SEFE.*

Proof Let $G_1 = (V_1, E_1)$ be a planar graph, let $G_2 = (V_2, E_2)$ be a tree, and let $G = (V_1 \cap V_2, E_1 \cap E_2)$ be their shared graph. Let \mathcal{E}_1 be a planar embedding of G_1 , which exists since G_1 is planar. Let \mathcal{E} denote the restriction of \mathcal{E}_1 to G , i.e., it specifies for each vertex $v \in V$ the circular ordering of the edges in $E_1 \cap E_2$ around it. We arbitrarily extend this information to a rotation system of G_2 . Since every rotation system of a tree is an embedding, this defines a planar embedding \mathcal{E}_2 of G_2 whose restriction to G is \mathcal{E} . That is, \mathcal{E}_1 and \mathcal{E}_2 are compatible embeddings of G_1 and G_2 , and therefore, a SEFE exists by Theorem 8. \square

We note that this proof is significantly shorter than the original one [25], which highlights the power of the underlying Theorem 8 that essentially does all of the heavy lifting. On the other hand, in contrast to the original construction, it is only existential and does not actually provide a SEFE in the form of a drawing. For the purpose of testing the existence of a SEFE, the increased simplicity seems key, and throughout the remainder of this section we will concern ourselves only with the (equivalent) problem of determining the existence of compatible embeddings.

The outline of a generic SEFE algorithm for (G_1, G_2) with shared graph G then becomes as follows.

1. Compute the set Ω_i of all planar embeddings of G that are *compatible with G_i* , for $i = 1, 2$, in the sense that they can be extended to a planar embedding of G_i .
2. Check whether $\Omega_1 \cap \Omega_2 \neq \emptyset$.

Indeed the same algorithm works also for $k \geq 2$ input graphs in the sunflower case. The crux is that, usually, the size of Ω_i is not polynomially bounded in the size of the input, and it is thus not feasible to compute these sets by listing all their elements. To implement the algorithm efficiently, it is thus necessary to design a data structure that can implicitly represent all the planar embeddings of G that are compatible with some planar supergraph G_i . The requirements on such a data structure are twofold. First, it needs to be computable for input graphs (G_1, G_2) in polynomial time and, second, given the data structures representing Ω_1 and Ω_2 , it must be efficiently testable whether $\Omega_1 \cap \Omega_2 \neq \emptyset$. Note that the second property is essential; the pair (G_i, G) can be seen as an implicit representation of all the planar embeddings of G that are compatible with G_i . But then the intersection test is equivalent to the original SEFE problem making this a somewhat useless choice. To date, the existence of a general data structure for answering these questions is open; however, there have been fruitful developments over the past years, of which we will sketch the most

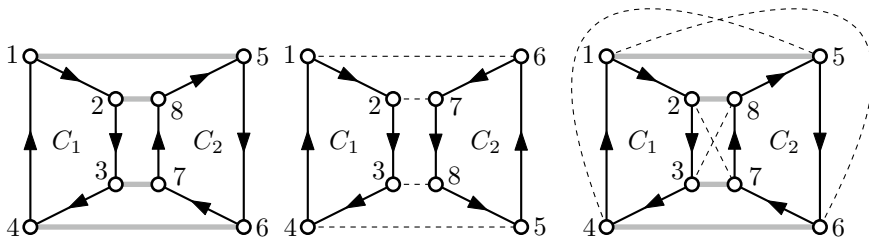


Fig. 13.10 Two graphs G_1 and G_2 with embeddings that induce the same rotation scheme on the shared graph G . The right figure shows that G_1 and G_2 do not admit a SEFE. The reason for this is that the embeddings of G_1 and G_2 induce different relative positions on the connected components of G . For G_1 , cycle C_2 lies to the left of C_1 and cycle C_1 lies to the left of C_2 , whereas for G_2 , cycle C_2 also lies to the left of C_1 , but C_1 lies to the right of C_2

important ones, as they will also nicely highlight the general idea of developing a constraint-based SEFE algorithm.

The first algorithms following this type of paradigm addressed the restriction of SEFE to inputs where the shared graph is biconnected. Interestingly, the simultaneous and independent papers [8, 37] followed the same basic approach but differed in their choice of representing all the possible planar embeddings of the shared graph. While Hauptler, Jampani and Lubiw [37] based their algorithm on PQ-trees, a well-known data structure used in, e.g., planarity testing, Angelini et al. [8] used SPQR-trees, a data structure specifically designed for representing all planar embeddings of a biconnected planar graph [22]. However, the way these algorithms are constructed inherently limit them to the respective class of instances. In the following, we will survey some further algorithmic approaches, which together overcome this limitation and form the state of the art of SEFE testing algorithms to this date.

13.3.3 Relative Positions

Recall that two embeddings of a planar graph are the same if they have both the same rotation system and the same relative positions. Up to 2012, all SEFE testing algorithms focused entirely on the rotation system of the shared graph, and were thus unable to solve instances of SEFE with a disconnected shared graph; see Fig. 13.10 for a pair of graphs that do not admit a SEFE but do admit embeddings that induce the same rotation scheme on the shared graph.

In the following, we sketch the work of Bläsius and Rutter [15], who first studied this problem in isolation by designing a constraint-based algorithm for testing whether two graphs whose shared graph is a collection of disjoint cycles admit a SEFE. Observe that, in this case, the shared graph $G = (V, E)$ has a unique rotation system, so the only relevant information about a planar embedding of G are the relative positions of the connected components of G with respect to each other.

The key is of course the representation of the planar embeddings of G . To this end, they adopt the following simple idea. Fix an arbitrary orientation for each connected component of G as a directed cycle. For each ordered pair of cycles $C, C', C \neq C'$ create a Boolean variable $\text{pos}_C(C')$ that is true if and only if C' lies on the right side of C in the planar embedding of G .

Note that an arbitrary assignment of true/false values to these variables does not necessarily correspond to a planar embedding of G . As an example consider three directed cycles C_0, C_1 , and C_2 . Then the assignment $\text{pos}_{C_i}(C_{i-1}) = \text{true}$ and $\text{pos}_{C_i}(C_{i+1}) = \text{false}$ for $i \in \{0, 1, 2\}$, where indices are taken modulo 3, does not correspond to a planar embedding. Namely, by the values of $\text{pos}_{C_2}(\cdot)$, it follows that C_1 is drawn as a cycle in the plane that has C_0 and C_2 on different sides. But then, for each of the remaining cycles C_0 and C_2 the other two cycles have to lie on the same side, which is not the case in the given assignment.

Bläsius and Rutter [15] characterize the truth assignments of the variables $\text{pos}_C(\cdot)$ that correspond to a planar embedding of a connected planar graph G_i containing the cycles and show that they can be expressed as the satisfying assignments of a set of 2-SAT clauses on these variables. Therefore, Ω_i can be formulated as the set of satisfying assignments of a 2-SAT formula φ_i for $i = 1, 2$. Hence $\Omega_1 \cap \Omega_2$ is described by the satisfying assignments of the formula $\varphi = \varphi_1 \wedge \varphi_2$, which is again an instance of 2-SAT. Thus it can easily be checked whether $\Omega_1 \cap \Omega_2 \neq \emptyset$. The above naive construction yields a quadratic-size formula (as it has a quadratic number of variables). Bläsius and Rutter proceed to show that the number of variables and constraints can be reduced to a linear number. Overall, they obtain the following result.

Theorem 10 ([15]) *Given two planar graphs G_1 and G_2 whose shared graph is a collection of disjoint cycles, it can be tested in linear time whether G_1 and G_2 admit a SEFE.*

Moreover, the result can be generalized to the case where the shared graph consists of several connected components C_1, \dots, C_k , each of which has a fixed rotation system, and it only remains to choose consistent relative positions. In this case, the relative position $\text{pos}_{C_i}(C_j)$ is described by giving a face of C_i whose interior contains C_j . The key insight is that, in most cases, this is a binary choice, and therefore, can be encoded as a Boolean variable. In the other cases, a linear number of different choices is possible, but these choices do not affect each other or the final outcome of the algorithm.

Theorem 11 ([15]) *SEFE can be solved in quadratic time, if the embedding of each connected component of the common graph is fixed.*

13.3.4 Simultaneous PQ-Ordering

We sketch a second constraint-based approach to the SEFE problem. The key here is a novel type of embedding representation that describes all planar embeddings of a

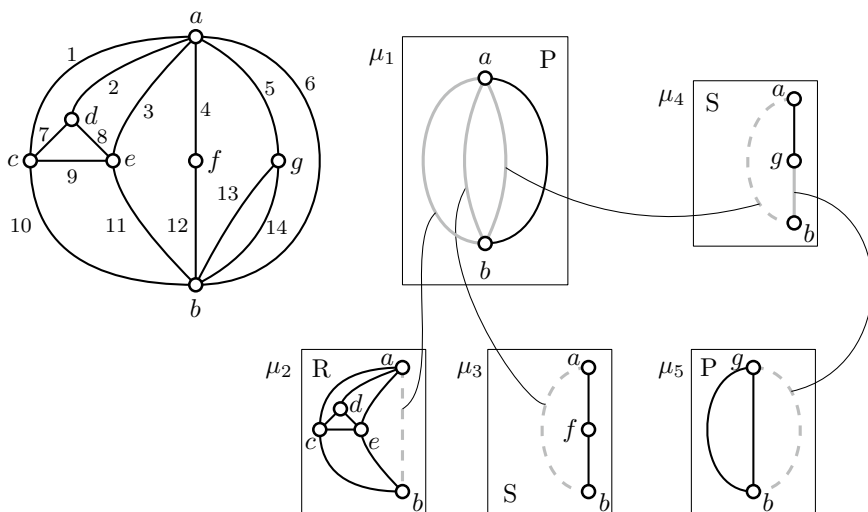


Fig. 13.11 A graph G and its SPQR-tree, where each node of the tree is annotated with its type (P for parallel, S for series, or R for rigid) and its skeleton graph. For clarity, Q-nodes are omitted. Instead, skeletons contain real edges (black) and virtual edges (gray). Observe that the embedding choice for G amounts precisely to choosing the permutations of the edges of μ_1 and μ_5 as well as choosing the flip of the 3-connected skeleton of μ_5

connected graph, the shared graph G , that can be induced by a planar embedding of a biconnected supergraph of G , e.g., the input graph G_i . Since two of these representations can be intersected efficiently, we obtain a constraint-based algorithm for SEFE when the input graphs are biconnected and the shared graph is connected [16]. The algorithm can be further generalized to admit cutvertices with at most two non-trivial blocks. In particular, this includes all cases where the input graphs have maximum degree 5 and the shared graph is connected.

The idea for the construction of this embedding representation is the following. Consider a biconnected graph G . One possibility to describe all planar embeddings of G is to consider the SPQR-tree of G and the embedding choices presented there; see Fig. 13.11.

Another possibility is to consider the embedding from the perspective of a single vertex v by making use of PQ-trees [17]. A PQ-tree T is a tree with a fixed rotation system whose internal vertices are either P- or Q-nodes. We consider two PQ-trees as *equivalent* if one is obtained from the other by arbitrarily changing the rotations at P-nodes and by possibly reversing the rotations at Q-nodes. Traversing a PQ-tree along an Euler tour that respects the rotation system defines a circular ordering of its leaves. A PQ-tree T represents a set of circular orders $\mathcal{O}(T)$ of its leaves $L(T)$, namely the circular orders defined by all PQ-trees that are equivalent to T .

It is well known that, for a biconnected graph G , the possible rotations around a vertex v over all planar embeddings of G can be represented by a PQ-tree $T(v)$ (see e.g., [16]), which we call the *embedding tree* of v . In fact, the embedding tree for

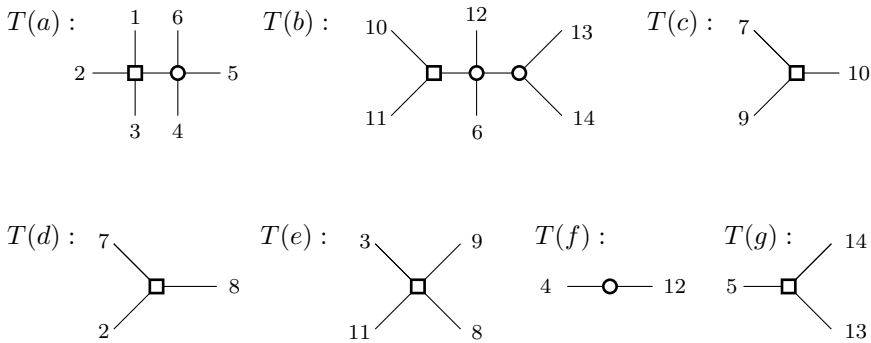


Fig. 13.12 Embedding trees of the graph G from Fig. 13.11. P-nodes are represented by disks, their neighbors may be permuted arbitrarily. Q-nodes are represented as squares and the ordering of their neighbors may be either kept or fully reversed

a vertex v can easily be read from the SPQR-tree of G . Essentially, P-nodes of the SPQR-tree become P-nodes in the PQ-tree, whereas R-nodes of the SPQR-tree become Q-nodes of the PQ-tree. Figure 13.12 shows the embedding trees of the vertices of the graph G from Fig. 13.11.

The advantage of this consideration is the following. The SPQR-tree gives a very global perspective on the embedding, where determining the ordering around v for a specific set of embedding choices requires checking the various embedding choices of the SPQR-tree that together determine the rotation of v . The PQ-tree perspective, on the other hand, is far more local as $T(v)$ directly represents the ordering of the edges around v . Since in the setting of a connected shared graph all that matters are the rotations of vertices, it seems that a more local embedding representation should be advantageous as it allows to express constraints in a more local fashion.

The main difficulty is, however, that the local embedding choices, where for each vertex $v \in V$, we have to choose one of the orders represented by the embedding tree $T(v)$, are not independent. Not every choice of orientations of the trees in Fig. 13.12 actually yields an embedding for the graph G from Fig. 13.11. Namely, the edge orderings of P- and Q-nodes of the PQ-trees that stem from the same P- and R-nodes of the SPQR-tree, respectively, need to be oriented consistently. We thus need to be able to express constraints on these choices. To model these constraints, Bläsius and Rutter [16] introduce networks of PQ-trees and the corresponding problem SIMULTANEOUS PQ-ORDERING.

Recall that each PQ-tree T with leaf set $L(T)$ represents a set $\mathcal{O}(T)$ of circular orders of $L(T)$. Let now T and T' be two PQ-trees. An arc e from T to T' is a triplet $e = (T, T', \varphi)$ such that φ is an injective mapping $\varphi: L(T') \rightarrow L(T)$. A choice of orderings $O \in \mathcal{O}(T)$ and $O' \in \mathcal{O}(T')$ satisfies the arc e if $\varphi(O')$ coincides with the restriction of O to $\varphi(L(T'))$. That is, arc e expresses the condition that the ordering for T has to be chosen in such a way that the subordering of the elements whose ordering is restricted by T' is compatible with T' . Similarly, Bläsius and Rutter

also allow *reversing arcs*, where one insists that the ordering $\varphi(O')$ is the reverse of the restriction of O to $\varphi(L(T'))$.

An instance of the SIMULTANEOUS PQ-ORDERING problem is a DAG N where each node v is equipped with a PQ-tree T_v , and moreover, each arc (u, v) comes with an injective mapping $\varphi_{u,v}: L(T_v) \rightarrow L(T_u)$ and is either marked as a normal or as a reversing arc. The problem SIMULTANEOUS PQ-ORDERING asks whether it is possible to choose for each node $v \in V(N)$ an ordering $O_v \in \mathcal{O}(T_v)$ such that all arcs of N are satisfied. This is a very powerful problem, and in fact it is easily shown that this problem is NP-complete. To build some familiarity with the problem, we repeat the proof from [16].

Theorem 12 SIMULTANEOUS PQ-ORDERING is NP-complete.

Proof The problem is clearly in NP, since we can guess a circular ordering for each node of an instance of SIMULTANEOUS PQ-ORDERING and then check in polynomial time, whether it satisfies all arcs.

We show NP-hardness by giving a reduction from the NP-hard problem CYCLIC ORDERING [32], which given a ground set X and a set of triplets $C = \{(x_1^i, x_2^i, x_3^i)\}_{i=1}^k$ asks whether there exists a circular ordering O of X such that for each triplet $(x_1^i, x_2^i, x_3^i) \in C$ the circular order in O is (x_1^i, x_2^i, x_3^i) .

The network N we construct has $k + 2$ nodes. A source s whose tree T_s consists of a *universal* PQ-tree with leaf set X , i.e., it consists of a single P-node whose neighbors are the elements of X . In this way, T_s represents all circular orderings of X . For each triplet (x_1^i, x_2^i, x_3^i) of C we create a corresponding node i whose tree is a single Q-node whose leaves are x_1^i, x_2^i, x_3^i , and we create an arc (s, i, φ_i) where $\varphi_i: \{x_1^i, x_2^i, x_3^i\} \rightarrow X$ is the identity. Finally, we create a sink node t whose PQ-tree consists of a single Q-node with leaves 1, 2, 3. For $i = 1, \dots, k$, we add the arc (i, t, φ'_i) with $\varphi'_i: \{1, 2, 3\} \rightarrow \{x_1^i, x_2^i, x_3^i\}$ given by $\varphi'_i(j) = x_j^i$ for $j = 1, 2, 3$. We claim that the constructed instance of SIMULTANEOUS PQ-ORDERING is satisfiable if and only if the original instance of CYCLIC ORDERING admits a valid solution.

Given an ordering O of X that is compatible with all triplets in C , we choose the ordering of s as O , and we choose the circular ordering for node i as (x_1^i, x_2^i, x_3^i) and the ordering for node t as $(1, 2, 3)$. This choice clearly satisfies all arcs of the instance of SIMULTANEOUS PQ-ORDERING.

Conversely, assume that we find an ordering for each node of N such that all arcs are satisfied. Without loss of generality, we may assume that the order chosen for t is $(1, 2, 3)$; otherwise we reverse all chosen orders. Let O be the order chosen for node s . The fact that the arc (i, t) is satisfied implies that the order chosen for node i is (x_1^i, x_2^i, x_3^i) . And since the arcs (s, i) are all satisfied it follows that O is compatible with all triplets (x_1^i, x_2^i, x_3^i) for $i = 1, \dots, k$, i.e., it forms a solution of the instance of CYCLIC ORDERING. \square

We now return to the construction of an embedding representation of a biconnected planar graph $G = (V, E)$ by means of an instance of SIMULTANEOUS PQ-ORDERING. As mentioned before, the task of deciding an embedding is equivalent to choosing an ordering for each of the embedding trees $T(v)$, $v \in V$, such that

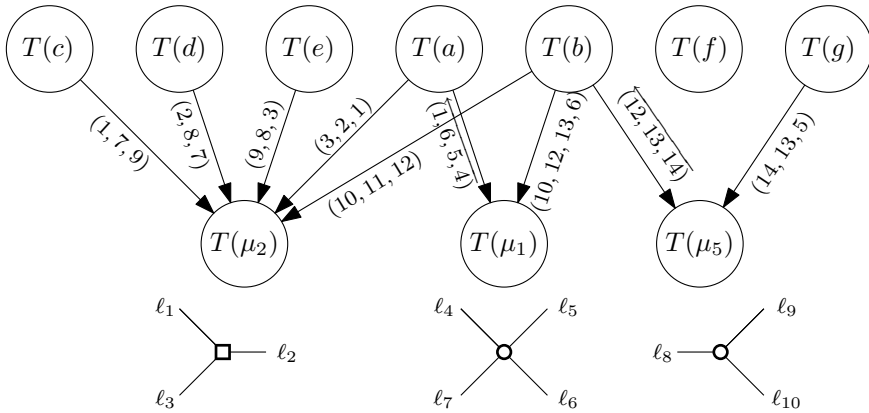


Fig. 13.13 SIMULTANEOUS PQ-ORDERING embedding representation of the graph G from Fig. 13.11. Each arc is annotated with a vector that encodes the mapping from the leaves of the child tree to the leaves of the parent by listing the images of the leaves ℓ_i in the order of their subscripts. For example, the vector $(1, 7, 9)$ for the arc from $T(c)$ to $T(\mu_2)$ encodes that ℓ_1 maps to 1, ℓ_2 to 7, and ℓ_3 to 9. Moreover, the vectors of reversing arcs are marked with a backward arrow

the edge orderings of P- and Q-nodes of the PQ-trees that stem from the same P- and R-nodes of the SPQR-tree, respectively, are oriented consistently. We ensure this by creating additional constraint trees; one for each P- and R-node of the SPQR-tree. The resulting instance, which is called the SIMULTANEOUS PQ-ORDERING embedding representation, for the graph from Fig. 13.11 is shown in Fig. 13.13.

For each R-node μ of the SPQR-tree of G , we create a PQ-tree $T(\mu)$, called the constraint tree of μ , that consists of one Q-node and three leaves, say ℓ_1, ℓ_2, ℓ_3 . For each Q-node in an embedding tree $T(v)$ that stems from μ , we create an arc from $T(v)$ to $T(\mu)$, where ℓ_1, ℓ_2, ℓ_3 are mapped to edges incident to v that are in different virtual edges of the skeleton of μ and that are all ordered clockwise in one planar embedding of that skeleton. As an example, consider the tree $T(\mu_2)$ in Fig. 13.13.

For each P-node μ of the SPQR-tree, we create a constraint tree $T(\mu)$ that consists of a single P-node and whose leaves correspond to the parallel edges in the skeleton of μ . For each P-node in an embedding tree $T(v)$ that stems from μ , we create an arc from $T(v)$ to $T(\mu)$, where each leaf ℓ of $T(\mu)$ is mapped to an edge of G that is contained in the virtual edge of the skeleton of μ the leaf ℓ corresponds to. Note that for each P-node μ there are precisely two trees $T(v)$ for which we create such arcs (the two poles), and we mark exactly one of the arcs as reversing. This models precisely that the clockwise orders of the virtual edges around each of the poles of a P-node skeleton must be the reverse of each other. Examples of this are the trees $T(\mu_1)$ and $T(\mu_5)$ in Fig. 13.13.

Bläsius and Rutter [16] show that the solutions to the SIMULTANEOUS PQ-ORDERING embedding representation of a biconnected graph G are exactly the rotation systems of planar embeddings of G . Moreover, if $H = (V', E')$ is a subgraph of G , then the rotation system of H induced by restricting the rotation system

of G can be obtained by creating for each $v \in V'$ a *subgraph tree* $T'(v)$ consisting of a single P-node whose leaves are the edges of H incident to v and creating an arc from the embedding tree $T(v)$ to $T'(v)$, whose associated mapping is the identity.

Now, given two biconnected planar graphs (G_1, G_2) with shared graph G , we can decide the existence of a SEFE as follows. We form an instance of SIMULTANEOUS PQ-ORDERING that is obtained by taking the SIMULTANEOUS PQ-ORDERING embedding representations I_1 and I_2 of G_1 and G_2 , respectively, including the subgraph trees for the graph G . The instance for (G_1, G_2) is obtained by joining I_1 and I_2 into a single instance, by identifying the subgraph trees that correspond to the same vertex v of G . We denote the resulting instance by $I(G_1, G_2)$. Observe that the solutions for the individual instances I_1 and I_2 correspond bijectively to planar rotation systems of G_1 and G_2 . Hence, by construction, the solutions of instance I bijectively correspond to pairs of planar rotation systems of G_1 and G_2 that induce the same rotation system on the shared graph. If the shared graph is connected, then the relative positions do not matter, and the existence of a solution is equivalent to the existence of a SEFE. By showing that the instance $I(G)$ can be solved in quadratic time, the following result is obtained.

Theorem 13 ([16]) *SEFE can be decided in $O(n^2)$ time for two biconnected input graphs whose shared graph is connected.*

Moreover, the result can be slightly generalized. For example, if each cutvertex of the input graphs is incident to at most *two non-trivial block*, i.e., blocks that do not consist of a single edge. This includes, for example, all graphs with maximum degree 5. The reason is that, in this case, the rotations also at cutvertices can be represented by a PQ-tree. Moreover, the result also applies if the shared graph is disconnected but acyclic, since also in this case the relative positions do not matter. The last result has found recent applications in clustered planarity [9]. The following theorem summarizes these generalizations. To the best of our knowledge, these general forms can currently not be handled by the Hanani–Tutte approach.

Theorem 14 *SEFE can be decided in $O(n^2)$ time for two input graphs G_1 and G_2 with shared graph G if both the following conditions hold:*

- (i) *each cutvertex of the input graphs is incident to at most one non-trivial block,*
- (ii) *the shared graph G is connected or acyclic.*

We note that the results from this section crucially depend on the fact that there are only two input graphs. They do not apply to three or more input graphs even in the sunflower case.

13.3.5 *Combination of Rotation System and Relative Positions*

The results of Sects. 13.3.3 and 13.3.4 are in a sense complementary. The former only deals with relative positions and ignores all information concerning the rotation

system, whereas the latter ignores relative positions entirely and deals only with the rotation system. It is a natural question whether it is possible to combine these two results to obtain the best of both worlds.

Note that the Hanani–Tutte approach does not make this distinction and thus can handle both aspects of embeddings at the same time. Indeed, the work of Schaefer [43], which appeared almost simultaneously with the results from the previous two subsections, was the first algorithmic approach that could handle both the rotation system and relative positions together. It was only a year later when it was found out how to combine the two approaches from the previous two subsections.

Bläsius et al. [13] proceed in two ways. First, they give a reduction rule that allows to replace in a SEFE instance each connected component of the shared graph that is biconnected by a cycle [13, Lemma 10]. If all connected components of G are biconnected, then iteratively applying this procedure yields an instance whose shared graph is a collection of cycles, which can hence be solved using Theorem 10.

To deal with non-biconnected components, Bläsius et al. restrict their attention to graphs of maximum degree 3 (they can also handle a slightly more general case that allows higher degrees within biconnected components, as long as they do not produce large parallel structures). A good starting point is of course the result from Theorem 10. However, the approach there encodes the relative positions of two distinct connected components C_1 and C_2 of the shared graph by a variable $\text{pos}_{C_1}(C_2)$, which can take as value any of the faces of C_1 . But if the embedding of C_1 is no longer fixed, then its faces can change and the possible values one may choose for $\text{pos}_{C_1}(C_2)$ depend on the embedding of C_1 , making it difficult to implement this idea.

To enable this approach, Bläsius et al. [13] choose a cycle cover of the connected component C_1 of G and express the relative positions of C_2 with respect to each cycle in the cycle cover of C_1 . They show that these relative positions uniquely determine the face f of C_1 that contains C_2 , if it exists. However, such a cycle basis usually contains a linear number of different cycles, and only a small fraction of the possible relative position assignments corresponds to a face in any embedding. Here Bläsius et al. [13] make use of the fact that, due to the low degrees of the shared graph, the embedding choice of G can essentially be described by independent Boolean decisions, and the relations between these embedding choices and which relative positions correspond to faces of the embedding can be encoded in an equation system over \mathbb{F}_2 .

Theorem 15 ([13, Corollary 1]) *SEFE can be solved in $O(n^3)$ time if each connected component of the shared graph is biconnected or has maximum degree 3.*

We note that the instances covered by this result coincide with the one from the Hanani–Tutte approach from Theorem 7. The approach, here, can be generalized to also handle cases where the shared graph is not biconnected but has a degree more than 3, provided that the interaction with the two input graphs is sufficiently well behaved. Bläsius et al. [13] formalize this with the notion of the common P-node degree. However, the details are somewhat technical, and we omit them here.

13.3.6 Obstacles to Further Progress

As of today, it remains open how to choose consistent relative positions in the presence of rotation system choices of the shared graph, where more than three edges may be permuted arbitrarily, which appears both for (high-degree) cutvertices of the shared graph, and for large parallel structures of the shared graph that are not grouped into binary choices by the individual graphs.

It seems that to overcome the current stagnation, a better understanding of the interplay of the rotation at cutvertices on the one hand and the relative positions of connected components on the other hand is necessary. In fact, understanding the cutvertices alone is a formidable task. Even the case where the shared graph G is a tree (i.e., all non-leaves are cutvertices) is still open and, by Theorem 5, intimately related to the complexity of the infamous c -planarity problem.

13.4 Realizability Problems

As mentioned in the introduction to Sect. 13.3.2, the characterization of SEFE in terms of the existence of compatible embeddings from Theorem 8 has provided a new perspective on the development of SEFE algorithms, which now focus entirely on the existence of compatible embeddings, rather than on actual simultaneous planar drawings. While this has certainly been advantageous from the algorithmic point of view, it comes with the disadvantage that these algorithms only test the existence of a drawing but do not compute one. By contrast, many previous results have indeed given quality guarantees on the obtained drawings. This opens the field for the study of so-called *realizability problems*, which ask, for a given pair of compatible embeddings, how drawings realizing these embeddings can look like. We note that, though the combinatorial embeddings of G_1 and G_2 are both fixed, the topology of $G_1 \cup G_2$ is not, as we may still choose which of the exclusive edges of G_1 and G_2 cross each other. It thus makes sense to try and minimize the number of crossings or to minimize the number of bends in the drawing.

From the perspective of crossing minimization, Chimani et al. [19] have shown that in a SEFE it may be necessary that a pair of edges crosses multiple times, even linearly often in non-sunflower instances.

The question of minimizing the bends in a SEFE that realizes a given pair of compatible embedding was first introduced by Haeupler et al. [37]. They showed that if the shared graph is connected, then one can draw G_1 straight-line with the given embedding, and one can then draw the graph G_2 with at most $O(|V(G)|)$ bends per edge. Later it was shown that this can in fact be made to work even if the shared graph is disconnected.

Theorem 16 ([18]) *Let G be a graph with a fixed embedding and let H be a subgraph of G with a fixed straight-line drawing. Then there exists a planar drawing of G that extends the drawing of H and has at most $72|V(H)|$ bends per edge.*

It is an open problem whether the factor of 72 can be significantly decreased. It is not hard to see that drawing one of the graphs straight-line in an arbitrary fashion may impose a linear number of bends per edge on the other one. It is thus natural to ask whether it is possible to draw all edges with a constant number of bends. Grilli et al. [35] define a (k_1, k_2) -drawing as one where the shared edges have at most k_1 bends and the exclusive edges have at most k_2 bends. They answer the above question in the affirmative by showing that every pair of graphs with given compatible embeddings admits a $(0, 9)$ drawing, i.e., the shared graph is drawn with straight-line segments, and the exclusive edges have at most nine bends each. Grilli et al. give better results if the shared graph is connected or even biconnected. In particular, if the shared graph is biconnected and an induced subgraph of both input instances, then there exists a $(0, 0)$ -drawing, i.e., a straight-line simultaneous drawing.

Theorem 17 ([35, Theorem 1, Corollaries 4,5]) *Let G_1, \dots, G_k be a sunflower instance of SEFE with shared graph G that admits compatible embeddings.*

- (i) *If G is biconnected and an induced subgraph of each G_i , then there exists a $(0, 0)$ -drawing.*
- (ii) *If G is biconnected, then there exists a $(0, 1)$ -drawing.*
- (iii) *If G is connected and an induced subgraph of each G_i , then there exists a $(0, 1)$ -drawing.*
- (iv) *If G is connected, then there exists a $(0, 3)$ -drawing.*
- (v) *If G is an induced subgraph of each G_i , then there exists a $(0, 4)$ -drawing.*
- (vi) *In all cases there exists a $(0, 9)$ -drawing.*

Frati et al. [26] consider the same problem and give drawing algorithms for compatible embeddings of two trees, a planar graph and a tree, and two planar graphs, respectively. For two trees they use only one bend per edge, for the other cases they use six bends per edge, while shared edges are drawn without bends.

Theorem 18 *Any pair of planar graphs that admits compatible embeddings has a $(0, 6)$ -drawing such that any pair of exclusive edges crosses at most 16 times.*

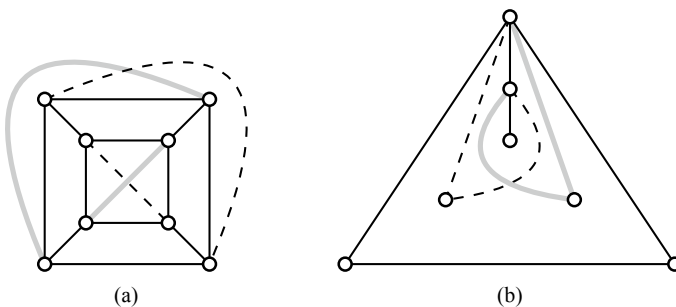


Fig. 13.14 Compatible embeddings that do not admit a straight-line simultaneous drawing even when the shared graph is **a** biconnected or **b** connected and an induced subgraph of each instance

It remains an open question whether these results can be improved. Grilli et al. [35] show that there exist graphs with compatible embeddings that do not admit a $(0, 0)$ -drawing; see Fig. 13.14. But it is unclear whether a $(0, 1)$ -drawing is always possible. This, however, seems quite unexpected and rather indicates a distinctive lack of lower bounds for such drawings. For SEFE of k graphs in the sunflower model, Grilli et al. prove the following asymptotic lower bound.

Theorem 19 *There exist compatible embeddings of k graphs with sunflower intersection such that any $(0, c)$ -drawing has $c \in \Omega((\sqrt{2})^k/k)$.*

13.5 Conclusion

We have surveyed the state of the art in the area of simultaneous embeddings with a focus on the findings that appeared after the previous survey [14] from 2012. Since then, it has turned out that simultaneous embedding with fixed edges is indeed one of the most general variants of planarity. It encompasses almost all planarity variants for which efficient testing algorithms are known and also some whose complexity is unknown, such as the problem of clustered planarity.

We have further described the current algorithmic approaches and their limitations. In particular, it seems that further progress in the constrained-embedding approach hinges on a better understanding of the interplay between the relative positions of the connected components of the shared graph and the embedding choices offered by cutvertices of the input graphs.

Finally, we have discussed realization problems, which ask to further optimize aesthetic criteria of simultaneous planar drawings, whose existence is certified by compatible embeddings, such as the number of crossings per edge pair and the number of bends per edge.

References

1. Angelini, P., Chaplick, S., Cornelsen, S., Da Lozzo, G., Di Battista, G., Eades, P., Kindermann, P., Kratochvíl, J., Lipp, F., Rutter, I.: Simultaneous orthogonal planarity. In: 24th International Symposium on Graph Drawing and Network Visualization (GD'16), vol. 9801 of Lecture Notes in Computer Science, pp. 532–545. Springer (2016)
2. Angelini, P., Da Lozzo, G.: SEFE = c -planarity? *Comput. J.* **59**(12), 1831–1838 (2016)
3. Angelini, P., Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M., Rutter, I.: Testing cyclic level and simultaneous level planarity (2015). CoRR, [arXiv:1510.08274](https://arxiv.org/abs/1510.08274)
4. Angelini, P., Da Lozzo, G., Di Battista, G., Frati, F., Patrignani, M., Rutter, I.: Beyond level planarity. In: Hu, Y., Nöllenburg, M. (eds.) Proceedings of the 24th International Symposium on Graph Drawing and Network Visualization (GD'16), vol. 9801 of Lecture Notes in Computer Science, pp. 482–495. Springer (2016)
5. Angelini, P., Da Lozzo, G., Neuwirth, D.: Advancements on SEFE and partitioned book embedding problems. *Theor. Comput. Sci.* **575**, 71–89 (2015)

6. Angelini, P., Di Battista, G., Frati, F.: Simultaneous embedding of embedded planar graphs. *Int. J. Comput. Geometry Appl.* **23**(2), 93–126 (2013)
7. Angelini, P., Di Battista, G., Frati, F., Jelínek, V., Kratochvíl, J., Patrignani, M., Rutter, I.: Testing planarity of partially embedded graphs. *ACM Trans. Algorithms* **11**(4), 32:1–32:42 (2015)
8. Angelini, P., Di Battista, G., Frati, F., Patrignani, M., Rutter, I.: Testing the simultaneous embeddability of two graphs whose intersection is a biconnected or a connected graph. *J. Discrete Algorithms* **14**, 150–172 (2012)
9. Angelini, P., Lozzo, G.D.: Clustered planarity with pipes. *Algorithmica* **81**(6), 2484–2526 (2019)
10. Argyriou, E.N., Bekos, M.A., Kaufmann, M., Symvonis, A.: Geometric RAC simultaneous drawings of graphs. *J. Graph Algorithms Appl.* **17**(1), 11–34 (2013)
11. Bachmaier, C., Brandenburg, F.J., Forster, M.: Radial level planarity testing and embedding in linear time. *J. Graph Algorithms Appl.* **9**(1), 53–97 (2005)
12. Bekos, M.A., van Dijk, T.C., Kindermann, P., Wolff, A.: Simultaneous drawing of planar graphs with right-angle crossings and few bends. *J. Graph Algorithms Appl.* **20**(1), 133–158 (2016)
13. Bläsius, T., Karrer, A., Rutter, I.: Simultaneous embedding: edge orderings, relative positions, cutvertices. *Algorithmica* **80**(4), 1214–1277 (2018)
14. Bläsius, T., Kobourov, S.G., Rutter, I.: Simultaneous embedding of planar graphs. In: Tamassia, R. (ed.) *Handbook of Graph Drawing and Visualization, Discrete Mathematics and its Applications*, pp. 349–373. CRC Press (2014)
15. Bläsius, T., Rutter, I.: Disconnectivity and relative positions in simultaneous embeddings. *Comput. Geom.* **48**(6), 459–478 (2015)
16. Bläsius, T., Rutter, I.: Simultaneous PQ-ordering with applications to constrained embedding problems. *ACM Trans. Algorithms* **12**(2), 16:1–16:46 (2016)
17. Booth, K.S., Lueker, G.S.: Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.* **13**(3), 335–379 (1976)
18. Chan, T.M., Frati, F., Gutwenger, C., Lubiw, A., Mutzel, P., Schaefer, M.: Drawing partially embedded and simultaneously planar graphs. *J. Graph Algorithms Appl.* **19**(2), 681–706 (2015)
19. Chimani, M., Jünger, M., Schulz, M.: Crossing minimization meets simultaneous drawing. In: *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis '08)*, pp. 33–40. IEEE (2008)
20. Chojnacki, C.: Über wesentlich unplättbare Kurven im dreidimensionalen Raume. *Fundamenta Mathematicae* **23**(1), 135–142 (1934)
21. Cortese, P.F., Patrignani, M.: Clustered planarity = flat clustered planarity (2018). [CoRR arXiv:1808.07437](https://arxiv.org/abs/1808.07437)
22. Di Battista, G., Tamassia, R.: On-line maintenance of triconnected components with SPQR-trees. *Algorithmica* **15**(4), 302–318 (1996)
23. Estrella-Balderrama, A., Gassner, E., Jünger, M., Percan, M., Schaefer, M., Schulz, M.: Simultaneous geometric graph embeddings. In: *Proceedings of the 15th International Symposium on Graph Drawing, (GD'07)*, pp. 280–290 (2007)
24. Feng, Q., Cohen, R.F., Eades, P.: Planarity for clustered graphs. In: Spirakis, P.G. (ed.) *Proceedings of the 3rd Annual European Symposium on Algorithms (ESA'95)*, vol. 979 of *Lecture Notes in Computer Science*, pp. 213–226. Springer (1995)
25. Frati, F.: Embedding graphs simultaneously with fixed edges. In: Kaufmann, M., Wagner, D. (eds.) *Proceedings of the 14th International Symposium on Graph Drawing (GD'06)*, vol. 4372 of *Lecture Notes in Computer Science*, pp. 108–113. Springer (2006)
26. Frati, F., Hoffmann, M., Kusters, V.: Simultaneous embeddings with few bends and crossings. In: Giacomo, E.D., Lubiw, A. (eds.) *23rd International Symposium on Graph Drawing and Network Visualization (GD'15)*, vol. 9411 of *Lecture Notes in Computer Science*, pp. 166–179. Springer (2015)
27. Fulek, R., Kyncl, J., Malinovic, I., Pálvölgyi, D.: Clustered planarity testing revisited. *Electr. J. Comb.* **22**(4), P4.24 (2015)

28. Fulek, R., Kyncl, J., Pálvölgyi, D.: Unified Hanani-Tutte theorem. *Electr. J. Comb.* **24**(3), P3.18 (2017)
29. Fulek, R., Pelsmajer, M., Schaefer, M.: Hanani-Tutte for radial planarity. In: Di Giacomo, E., Lubiw, A. (eds.) *Graph Drawing and Network Visualization*, pp. 99–110. Springer International Publishing (2015)
30. Fulek, R., Pelsmajer, M., Schaefer, M.: Hanani-Tutte for radial planarity II. In: Hu, Y., Nöllenburg, M. (eds.) *Graph Drawing and Network Visualization*, pp. 468–481. Springer International Publishing (2016)
31. Fulek, R., Pelsmajer, M.J., Schaefer, M., Štefankovič, D.: Hanani-Tutte, monotone drawings, and level-planarity. In: Pach, J. (ed.) *Thirty Essays on Geometric Graph Theory*, pp. 263–287. Springer, New York (2013)
32. Galil, Z., Megiddo, N.: Cyclic ordering is NP-complete. *Theor. Comput. Sci.* **5**, (1977)
33. Gassner, E., Jünger, M., Percan, M., Schaefer, M., Schulz, M.: Simultaneous graph embeddings with fixed edges. In: *Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG'06)*, pp. 325–335 (2006)
34. Grilli, L.: On the NP-hardness of GRacSim drawing and k-SEFE problems. *J. Graph Algorithms Appl.* **22**(1), 101–116 (2018)
35. Grilli, L., Hong, S., Kratochvíl, J., Rutter, I.: Drawing simultaneously embedded graphs with few bends. In: *Proceedings of the 22nd International Symposium on Graph Drawing (GD'04)*
36. Gutwenger, C., Mutzel, P., Weiskircher, R.: Inserting an edge into a planar graph. *Algorithmica* **41**(4), 289–308 (2005)
37. Haeupler, B., Jampani, K.R., Lubiw, A.: Testing simultaneous planarity when the common graph is 2-connected. *J. Graph Algorithms Appl.* **17**(3), 147–171 (2013)
38. Jünger, M., Leipert, S.: Level planar embedding in linear time. In: Kratochvíl, J. (ed.) *Graph Drawing and Network Visualization*, pp. 72–81. Springer, Berlin Heidelberg (1999)
39. Jünger, M., Schulz, M.: Intersection graphs in simultaneous embedding with fixed edges. *J. Graph Algorithms Appl.* **13**(2), 205–218 (2009)
40. Lengauer, T.: Hierarchical planarity testing algorithms. *J. ACM* **36**(3), 474–509 (1989)
41. Opatrný, J.: Total ordering problem. *SIAM J. Comput.* **8**(1), 111–114 (1979)
42. Pach, J., Wenger, R.: Embedding planar graphs at fixed vertex locations. *Graphs Comb.* **17**(4), 717–728 (2001)
43. Schaefer, M.: Toward a theory of planarity: Hanani-tutte and planarity variants. *J. Graph Algorithms Appl.* **17**(4), 367–440 (2013)