Simultaneous Embedding of Planar Graphs with Few Bends*

Cesim Erten and Stephen G. Kobourov

Department of Computer Science,
University of Arizona
{cesim,kobourov}@cs.arizona.edu

Abstract. We present an $O(n)$ time algorithm for simultaneous embedding of pairs of planar graphs on the $O(n^2) \times O(n^2)$ grid, with at most three bends per edge, where n is the number of vertices. For the case when the input graphs are both trees, only one bend per edge is required. We also describe an $O(n)$ time algorithm for simultaneous embedding with fixed-edges for tree-path pairs on the $O(n) \times O(n^2)$ grid with at most one bend per tree-edge and no bends along path edges.

1 Introduction

Traditional problems in graph drawing involve the layout of a single graph. Problems in simultaneous graph drawing, involve the layout of multiple related graphs. Visualization of multiple related graphs, that is, graphs that are defined on the same set of vertices, arise in many applications. Software engineering, databases, and social network analysis, are all examples of areas where multiple relationships on the same set of objects are often studied.

Consider the case where a pair of related graphs is given and the goal is to visualize them so as to compare the two. If drawings for the two graphs are obtained independently, there would be little correspondence between the two layouts, since the viewer has no "mental map" between the two graphs. When examining a graph the user constructs a mental view of it, for example, using the positions of the vertices relative to each other. When viewing multiple graphs the user has to reconstruct this mental view after examining each graph and our goal should be to aid the user in this reconstruction while providing a readable drawing for each graph individually.

In simultaneous graph embedding, the vertices are placed in the exact same locations in all the graphs. Fixing the vertex positions in all the graphs preserves the mental map, but at the expense of readability of the individual drawings, if edges are to be drawn with straight-line segments. With this in mind, in this paper we consider the problem of drawing planar graphs on the same point-set using few bends. We describe efficient algorithms for simultaneous drawing of pairs of general planar graphs on small integer grids. We also describe better results for pairs of trees or tree-path pairs.

* This work is partially supported by the NSF under grant ACR-0222920 and by ITCDI under grant 003297.

J. Pach (Ed.): GD 2004, LNCS 3383, pp. 195–205, 2004.

1.1 Previous Work

The existence of simultaneous geometric embeddings for pairs of paths, cycles, and caterpillars is shown in [2]. Counter-examples for pairs of general planar graphs, pairs of outer-planar graphs, and triples of paths are also presented there. Modified force-directed algorithms are used in [1, 8] to simultaneously visualize general graphs, while attempting to preserve the user's mental map and obtaining readable individual drawings.

A related notion is that of *graph thickness* [12], defined as the minimum number of planar subgraphs whose union yields the given graph. If a graph has thickness two then it can be drawn on two layers such that each layer is crossing-free and the corresponding vertices of different layers are placed in the same locations. *Geometric thickness* is a version of the thickness problem where the edges are required to be straight-line segments [6]. Thus, if two graphs have a simultaneous geometric embedding, then their union has geometric thickness at most two. Similarly, the union of any two planar graphs has graph thickness at most two. Simultaneous geometric embedding techniques are used in [7] to show that degree-four graphs have geometric thickness two.

The existence of straight-line, crossing-free drawings for planar graphs is well known [9, 15, 17]. It is also known that every 3-connected planar graph has a convex drawing [16]. These techniques, however, do not guarantee anything about the resolution of the drawing and thus are not well-suited for automated graph drawing. The vertex resolution problem is addressed in [5, 14] where it is shown that any planar graph can be drawn with straight-lines and no crossings on a grid of size $O(n) \times O(n)$.

Simultaneous drawing of multiple graphs is also related to the problem of embedding planar graphs on a fixed set of points in the plane. Several variations of this problem have been studied. If the mapping between the vertices V and the points P is not fixed, then the graph can be drawn without crossings using two bends per edge in polynomial time [11]. However, if the mapping between V and P is fixed, then $O(n)$ bends per edge are necessary to guarantee planarity, where n is the number of vertices in the graph [13].

1.2 Our Results

Formally, the drawing $\mathcal{D}$ of a graph $G = (V, E)$ is a function that maps each vertex $u \in V$ to a distinct point $\mathcal{D}(u)$ in the plane, and each edge $(u, v) \in E$ to a simple Jordan curve $\mathcal{D}(u, v)$ with endpoints $\mathcal{D}(u)$ and $\mathcal{D}(v)$. The problem of simultaneously embedding two planar graphs G_1, G_2 is the problem of finding drawings D_1, D_2 with corresponding vertices of G_1 and G_2 mapped to the same points in the plane. The following are three variations of the simultaneous embedding problem for pairs of planar graphs:

Definition 1. Given two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ *simultaneous geometric embedding of* G_1 *and* G_2 is the problem of finding plane straight-line drawings D_1 and D_2 of G_1 and G_2, respectively, such that every vertex is mapped to the same point in the plane in both D_1 and D_2.

Definition 2. Given two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ *simultaneous embedding of* G_1 *and* G_2 *with fixed edges* is the problem of finding plane drawings D_1 and D_2 of G_1 and G_2, respectively, such that every vertex is mapped to the same point in the plane in both D_1 and D_2 and every shared edge $e \in G_1 \cap G_2$ is represented with the same simple open Jordan curve in D_1 and D_2.

Definition 3. Given two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ *simultaneous embedding of* G_1 *and* G_2 is the problem of finding plane drawings D_1 and D_2 of G_1 and G_2, respectively, such that every vertex is mapped to the same point in the plane in both D_1 and D_2.

The definitions are inclusive in the given order: simultaneous geometric embedding is a special case of simultaneous embedding with fixed edges, which is in turn a special case of simultaneous embedding.

In Section 2 we present a simple non-existence proof for simultaneous geometric embedding of a pair of graphs. Next, we present an $O(n)$ time algorithm for simultaneous embedding of pairs of planar graphs on the $O(n^2) \times O(n^2)$ grid, with at most three bends per edge, where n is the number of vertices. For the case when the input graphs are both trees, we only need one bend per edge. We also describe an $O(n)$ time algorithm for simultaneous embedding with fixed-edges for tree-path pairs on the $O(n) \times O(n^2)$ grid with at most one bend per tree-edge and no bends along the path edges. In Section 3 we briefly describe the implementation of these algorithms, show some of the resulting layouts, and conclude with several open problems.

2 Simultaneous Embedding

Simultaneous geometric embeddings are easy to find on small integer grids for pairs of paths, pairs of cycles, pairs of caterpillars, and others [2]. For pairs of general planar graphs, and even for pairs of outer-planar graphs, simultaneous geometric embeddings do not always exist. This is the main motivation for relaxing the conditions of simultaneous geometric embeddings, to just simultaneous embedding, by dropping the straight-line edge constraint. Under these weaker constraints, we can obtain simultaneous drawings with few bends per edge. Such drawings are also useful for pairs of trees, as it is not known whether simultaneous geometric embedding of pairs of trees is always possible.

2.1 Simultaneous Geometric Embedding

Here we briefly describe a simple case of a pair of planar graphs that do not admit simultaneous geometric embedding.

Theorem 1. *There exists a planar graph G and a path P such that there is no simultaneous geometric embedding of G and P.*

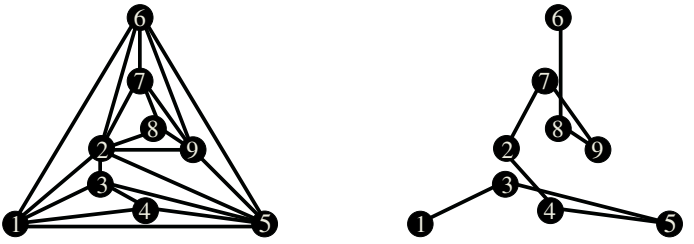

Fig. 1. Planar graph G and path P that do not allow a simultaneous geometric embedding.

Proof Sketch: Consider graph G and path P as shown in Fig. 1. Let G' be the subgraph of G induced on vertices $\{1, 2, 3, 4, 5\}$, and G'' be the subgraph of G induced on vertices $\{2, 6, 7, 8, 9\}$. Since G is 3-connected fixing the outer-face fixes an embedding for G. With the given outer-face of G, the path P contains two crossings: one involving $(2, 4)$, and the other one involving $(6, 8)$. Graph G' has six faces and unless we change the outer-face of G' such that it contains the edge $(1, 3)$ or $(3, 5)$, the edge $(2, 4)$ is involved in a crossing in the path. Similarly for G'', unless we change its outer-face such that it contains $(2, 7)$ or $(7, 9)$, the edge $(6, 8)$ is involved in a crossing in the path. However G' and G'' do not share any faces and removing both crossings depends on taking two different outer-faces, which is impossible. Thus, regardless of the choice for the outer-face of G, path P contains a crossing. □

2.2 Relaxing the Constraints

While some classes of planar graphs allow simultaneous geometric embedding, there are other classes that do not, and still others for which it is not known whether simultaneous geometric embeddings exist. Since the latter two categories contain a large number of planar graph classes (trees, outer-planar graphs, general planar graphs), it is natural to look for simultaneous drawings with weaker constraints. One possible solution for larger classes of graphs is to relax the constraints on the edges. Instead of restricting the edges to be straight-line segments we allow each edge to be drawn as a sequence of straight-line segments. Recall that such embeddings are called *simultaneous embeddings* (rather than simultaneous geometric embeddings).

Note that it is trivial to find a simultaneous embedding of any two planar graphs, if we are willing to accept a large number of bends per edge. Given a point-set P of size n in the plane and a planar graph G with n vertices, together with a one-to-one mapping between the vertices of G and the points in P, we can find a crossing-free drawing of G on P using edges with bends [13]. This allows us to embed any number of planar graphs simultaneously. However, the resulting drawings contain $O(n)$ bends per edge. Next, we describe methods to simultaneously embed any two planar graphs so that each edge has at most three bends.

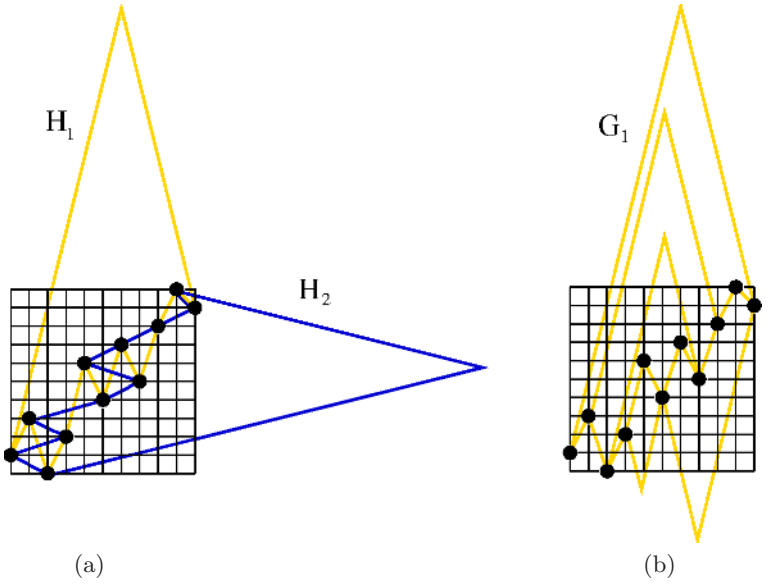

Fig. 2. (a) H_1 and H_2 drawn simultaneously. (b) Only the edges of G_1 are shown. Edges not in the hamiltonian cycle have the same slopes as the outermost edge.

2.3 Simultaneous Embedding with Few Bends

Since in this version of the problem we no longer insist on straight-line edges, the problem of simultaneously embedding two graphs boils down to finding a point-set in the plane and a mapping between the vertices of graphs and the points, with as few bends per edge as possible. The following theorem summarizes our results for pairs of general planar graphs.

Theorem 2. *Given two planar graphs G_1 and G_2 and a mapping between their vertices, we can simultaneously embed G_1 and G_2 using at most three bends per edge. The resulting drawing requires an integer grid of size $O(n^2) \times O(n^2)$ such that each vertex is placed on a grid point, and the algorithm requires $O(n)$ time, where n is the number of vertices.*

Proof Sketch: Vertex Placement: We make use of two techniques described in [2, 11]. Initially, we assume the graphs are 4-connected. We show how to remove this assumption later in the proof. First we find a hamiltonian cycle H_1 of G_1 and a hamiltonian cycle H_2 of G_2. We can do this in linear time using the algorithm of [4]. Starting at a random vertex in H_1 we traverse its vertices, assigning increasing x–coordinates to each vertex visited. Starting at a random vertex in H_2 we traverse its vertices, assigning increasing y–coordinates to each vertex visited. Not considering the final edges enclosing the cycles, this gives us an x–monotone path for H_1 and a y–monotone path for H_2; see Fig. 2(a).

Since both paths are monotone the edges of the paths are crossing-free. Let δ be the largest slope of the edges on the path defined by H_1. We complete the

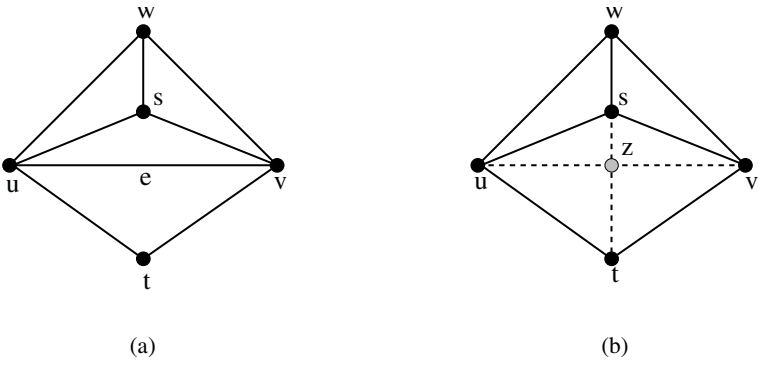

Fig. 3. (a) Removing separating triangles. (a) Edge e is part of the separating triangle (u, v, w). The two faces containing e are (u, v, s) and (u, v, t). (b) The separating triangle is removed by deleting e, introducing z and connecting it to u, v, s, and t.

drawing of the cycle H_1 by drawing the final edge between the leftmost vertex and the rightmost vertex. It is drawn with two segments such that the slope of the initial segment starting at the leftmost vertex is δ' and the slope of the second segment ending at the rightmost vertex is $-\delta'$, where δ' is slightly larger than δ. Since G_1 is hamiltonian, the cycle H_1 divides the edges into two groups: inside and outside edges (with respect to H_1). Then each of the inside edges is drawn with two line segments with slopes δ' and $-\delta'$ on the inside of H_1. Similarly, the outside edges are drawn with the same slopes on the outside of H_1. Note that some edges will overlap but postprocessing rotation can be used to remove the overlaps; see Fig. 2(b).

The edges of G_2 are handled in the same way with respect to H_2. It is easy to see that the vertex set requires grid size $n \times n$. The overall area of the drawing is larger, as the bend points lie outside the original grid. It is easy to show, however, that the entire drawing fits inside an $O(n^2) \times O(n^2)$ grid.

Making the Graphs 4-Connected: For the case when the input graphs are not 4-connected, we use techniques introduced in [11] to augment them. Given a 3-connected planar graph G we create a 4-connected planar graph by introducing new vertices. This is done by removing all the separating triangles in G. A separating triangle is a cycle of length 3 such that the removal of the vertices of the cycle disconnects G. Separating triangles of G can be easily found by the algorithm of [3]. Let $e = (u, v)$ be an edge of a separating triangle in G such that e is adjacent to the faces (u, v, s) and (u, v, t); see Fig. 3. We remove the separating triangle by inserting a dummy vertex z on e, deleting the edge e, and introducing four new edges: $(u, z), (v, z), (s, z), (t, z)$. The newly introduced vertex z is not part of any separating triangle, so each time we introduce such a vertex we decrease the number of separating triangles. Doing the same operation on all the separating triangles gives us a 4-connected planar graph.

Once G_1 and G_2 have been augmented to 4-connected graphs, we obtain the hamiltonian cycles H_1 and H_2 of G_1 and G_2. We augment the edges of H_2 with the extra vertices of G_1 and augment the edges of H_1 with the extra

vertices of G_2. The placement of the hamiltonian cycles and the drawing of the remaining edges is done as before. After finishing the placement we treat the dummy vertices as bend points and ignore the edges inserted in the augmentation phase. As a result, an edge $e = (u, v)$ that got split with a dummy vertex z ends up having three bend points: one between u and z, one at the location of z, and finally one between v and z. As there are $O(n)$ dummy vertices, the bounds for the integer grid remain unchanged.

Running Time: The two non-trivial operations are finding the separating triangles and finding the hamiltonian cycles. Finding the separating triangles and making the graphs 4-connected takes linear time [3]. A Hamiltonian cycle in a 4-connected planar graphs can be found in linear time [4]. □

The corollary below follows from the above theorem by fixing the slopes of all the edges and refining the grid.

Corollary 1. *Given two planar graphs G_1 and G_2 and a mapping between their vertices, we can simultaneously embed G_1 and G_2 using at most three bends per edge on an integer grid of size $O(n^3) \times O(n^3)$, with all the vertices and bend-points at grid-points.*

Proof Sketch: Consider the original $n \times n$ grid where H_1 and H_2 are placed. Let the slope $\delta = n$, where δ and $-\delta$ are the slopes of all edge segments among edges drawn with bends. Let $e = (u, v) \in G_1$ such that u is placed to the left of v and e is drawn with a bend point p. Let x_{dist}, y_{dist} be the x-coordinate and y-coordinate distances between u and v. The x-coordinate distance between u and the point p is $(n \times x_{dist} - y_{dist})/2n$. If we place a $2n \times 2n$ grid inside each unit square of the original grid, then the x-coordinate distance between u and p is an integer. Since the slope of the segment $\overline{up}$ is n, the y-coordinate distance between u and p is also an integer, and p is on a grid point. Similar argument applies to the edges of G_2 as well. The final grid area is $O(n^3) \times O(n^3)$. □

If both input graphs are trees then it is easy to reduce the number of bends required to only one per edge. The Theorem below follows from Theorem 2 and the above corollary.

Theorem 3. *Given two trees T_1 and T_2 and a mapping between their vertices, they can be simultaneously embedded in linear time, using at most one bend per edge, on an integer grid of size $O(n^2) \times O(n^2)$ (or $O(n^3) \times O(n^3)$, if both the vertices and bend-points are on grid points).*

2.4 Simultaneous Embedding with Fixed Edges

The algorithm from the previous section simultaneously embeds two planar graphs with the corresponding vertices mapped on the same positions and thus preserves the mental map for the vertex set. There is a significant drawback with respect to preserving the mental map for the edge set. In particular, edges common to both graphs are drawn differently in the two drawings unless they happen to be on the paths defined by the hamiltonian cycles.

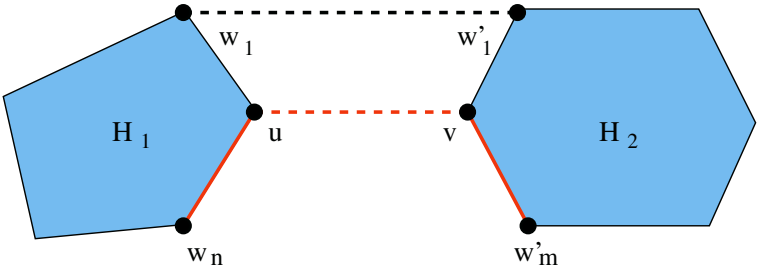

Fig. 4. Constructing the hamiltonian cycle H_T from H_1 and H_2. The common edges are shown in red.

Simultaneous embedding with fixed edges, requires that shared edges be represented the same way in both drawings. We describe an algorithm for simultaneous embedding with fixed edges for a tree and a path below.

Theorem 4. *Given a tree T, a path P, and a mapping between their vertices they can be simultaneously embedded with fixed edges in linear time, using at most one bend per edge, on an integer grid of size $O(n) \times O(n^2)$ (or $O(n^2) \times O(n^3)$, if both the vertices and bend-points are on the grid).*

Proof Sketch: The main idea is the same as that in Theorem 2, except that we ensure that the edges common to both T and P belong to the hamiltonian cycle for the tree. Then the path and the hamiltonian cycle have a simultaneous geometric embedding. The rest of the tree edges are routed as before using one bend per edge, thus yielding a simultaneous embedding with fixed edges for T and P.

Let $E_{T,P}$ be the set of edges common to both T and P. In order to obtain a hamiltonian cycle for the tree T we augment it with edges until the resulting graph T' has a hamiltonian cycle H_T that contains all edges that are in common with the path. We use a recursive divide-and-conquer procedure to construct H_T: the input to the recursive call is a subtree T and the output is the hamiltonian cycle H_T and the modified graph T'.

The base case for the recursion is a tree with just one node, $T = \{u\}$. In this case, let $H_T = (u, u)$, and $T' = T$. For all other cases, we take an edge $e = (u, v) \in E_{T,P}$ from T if such an edge exists. If not we take an arbitrary edge $e = (u, v) \in T$. Let T_1, T_2 be the two trees obtained after the removal of e from T. Assume we can construct hamiltonian cycles H_1, and H_2 of T_1 and T_2, respectively. Let T_1' and T_2' be the graphs that we get after these constructions, corresponding to T_1 and T_2, respectively. We merge the two subgraphs into the new graph $T' = T_1' \cup T_2'$ by adding e to T'.

In order to combine the hamiltonian cycles of the two subgraphs into a hamiltonian cycle for union, we need to add one more edge between the two subgraphs (as the edge e is a bridge). We add an edge between a neighbor u_{new} of u to a neighbor v_{new} of v and combine the two cycles by dropping the edges (u, u_{new}) and (v, v_{new}).

Let $H_1 = (u, w_1, w_2, \ldots, w_n, u)$ and $H_2 = (v, w_1', w_2', \ldots, w_m', v)$. If T_1' has only one vertex u we assign $u_{new} = u$, and if it has two vertices u and u' we

assign $u_{new} = u'$. We do similar assignments for v_{new} if T_2' has one or two vertices. In order to find u_{new}, v_{new} for all other cases, we check the first and the last edges of the hamiltonian cycles.

Since P is a path, either $(u, w_1) \notin E_{T,P}$, or $(u, w_n) \notin E_{T,P}$ (otherwise, vertex u must have degree greater than 2 in the path). Without loss of generality, assume $(u, w_1) \notin E_{T,P}$. We assign $u_{new} = w_1$. The same holds for H_2, that is, either $(v, w_1') \notin E_{T,P}$ or $(v, w_m') \notin E_{T,P}$. Without loss of generality, assume $(v, w_1') \notin E_{T,P}$. We assign $v_{new} = w_1'$. We insert edge (u_{new}, v_{new}) in T', if $e \neq (u_{new}, v_{new})$. As a result of this insertion the new hamiltonian cycle becomes, $H_T = (u, v, w_m', w_{m-1}', \ldots, w_1', w_1, w_2, \ldots, w_n, u)$; see Fig. 4.

Planarity: The above recursive procedure augments the tree T to a graph T' that has a hamiltonian cycle which contains all the edges that T has in common with the path P. We still need to show that the resulting graph T' is planar. Recall the recursive procedure above and let us assume that T_1' and T_2' are planar. Then there exists a planar embedding for T_1' so that the edge (u, w_1) is on the outer-face and a planar embedding for T_2' so that the edge (v, w_1') is on the outer-face. Since all the vertices u, w_1, v, w_1' are on the outer-faces of their graphs, the inserted edges (u, v) and (w_1, w_1') do not have any crossings with the edges of T_1' and T_2'. The resulting graph T' is planar, and the resulting embedding is a planar embedding.

Running Time: We only need to show that the hamiltonian cycle construction takes linear time, since the rest of the algorithm is the same as the one described in the previous section. Note that we do not have to explicitly find planar embeddings of T_1' and T_2' at each level of the recursion. The planar embedding of the final graph T' suffices and we can find it in linear time [10]. The merging of the two hamiltonian cycles requires constant number of operations at each recursive step and thus the overall running time of the algorithm is $O(n)$. □

3 Conclusion and Future Work

We implemented the algorithms described above using the LEDA library in `C++`. Fig. 5 shows the layouts obtained for a path and tree. All of the algorithms in this paper rely on the approach of augmenting planar graphs to hamiltonian planar graphs, so as to obtain simultaneous embeddings and simultaneous embeddings with fixed edges, using one or three bends. However, for simultaneous embedding with fixed edges, this technique cannot be extended from the path and tree case to pairs of trees (and hence cannot be extended to larger classes of planar graphs). We do not know of an algorithm for fixed-edge simultaneous embedding of pairs of trees. Neither do we have a counter-example. Similarly, the problem of simultaneous geometric embedding of pairs of trees is still open.

Acknowledgments

We would like to thank Michael Kaufmann and David Eppstein for discussions about these problems, and Petr Moravsky for helping with the implementation.

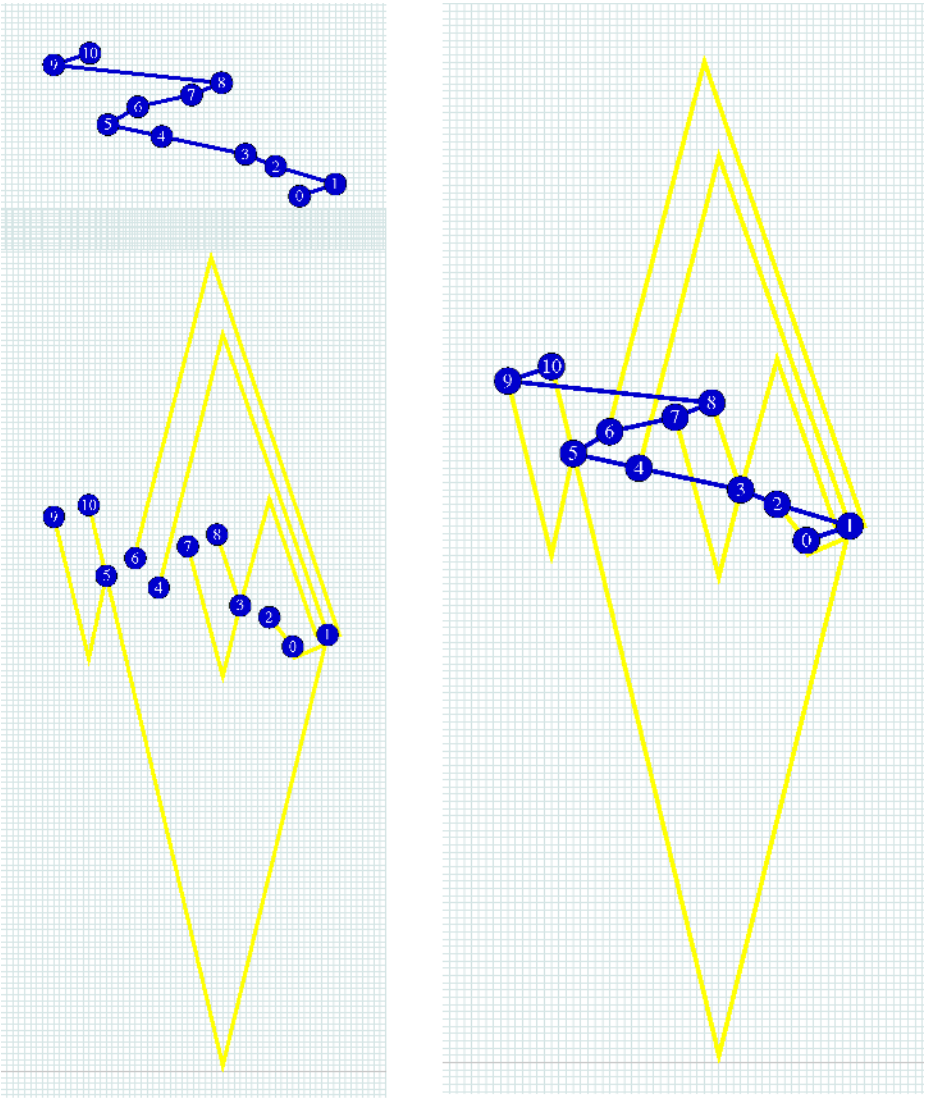

Fig. 5. A simultaneous embedding with fixed edges for a tree and a path. The path $(0, 1, \ldots, 10)$ is shown on the top left. The tree is shown on the bottom left. Note that the path and the tree share the edge (0,1). The combined view of the tree and the path is shown on the right.

References

1. U. Brandes and S. R. Corman. Visual unrolling of network evolution and the analysis of dynamic discourse. In *IEEE Symposium on Information Visualization (INFOVIS '02)*, pages 145–151, 2002.
2. P. Brass, E. Cenek, C. A. Duncan, A. Efrat, C. Erten, D. Ismailescu, S. G. Kobourov, A. Lubiw, and J. S. B. Mitchell. On simultaneous graph embedding. In *8th Workshop on Algorithms and Data Structures*, pages 243–255, 2003.
3. N. Chiba and T. Nishizeki. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14:210–223, 1985.
4. N. Chiba and T. Nishizeki. The hamiltonian cycle problem is linear-time solvable for 4-connected planar graphs. *Journal of Algorithms*, 10(2):187–211, 1989.
5. H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.
6. M. B. Dillencourt, D. Eppstein, and D. S. Hirschberg. Geometric thickness of complete graphs. *Journal of Graph Algorithms and Applications*, 4(3):5–17, 2000.
7. C. A. Duncan, D. Eppstein, and S. G. Kobourov. The geometric thickness of low degree graphs. In *20th Annual ACM-SIAM Symposium on Computational Geometry (SCG)*, pages 340–346, 2004.
8. C. Erten, S. G. Kobourov, A. Navabia, and V. Le. Simultaneous graph drawing: Layout algorithms and visualization schemes. In *11th Symposium on Graph Drawing (GD)*, pages 437–449, 2003.
9. I. Fáry. On straight lines representation of planar graphs. *Acta Scientiarum Mathematicarum*, 11:229–233, 1948.
10. J. Hopcroft and R. E. Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, 1974.
11. M. Kaufmann and R. Wiese. Embedding vertices at points: Few bends suffice for planar graphs. *Journal of Graph Algorithms and Applications*, 6(1):115–129, 2002.
12. P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of graphs: a survey. *Graphs Combin.*, 14(1):59–73, 1998.
13. J. Pach and R. Wenger. Embedding planar graphs at fixed vertex locations. *Graphs and Combinatorics*, 17:717–728, 2001.
14. W. Schnyder. Embedding planar graphs on the grid. In *Proceedings of the 1st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 138–148, 1990.
15. S. K. Stein. Convex maps. *Proceedings of the American Mathematical Society*, 2(3):464–466, 1951.
16. W. T. Tutte. How to draw a graph. *Proc. London Math. Society*, 13(52):743–768, 1963.
17. K. Wagner. Bemerkungen zum vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–32, 1936.