On Simultaneous Planar Graph Embeddings

P. Brass[1], E. Cenek[2], C. A. Duncan[3], A. Efrat[*4], C. Erten[*4], D. Ismailescu[5], S. G. Kobourov[⋆4], A. Lubiw[2], and J. S. B. Mitchell[⋆⋆6]

[1] Dept. of Computer Science, City College of New York, peter@cs.ccny.cuny.edu
[2] Dept. of Computer Science, University of Waterloo, {acenek,alubiw}@uwaterloo.edu
[3] Dept. of Computer Science, Univ. of Miami, duncan@cs.miami.edu
[4] Dept. of Computer Science, Univ. of Arizona, {alon,cesim,kobourov}@cs.arizona.edu
[5] Dept. of Mathematics, Hofstra University, Dan.P.Ismailescu@hofstra.edu
[6] Dept. of Applied Mathematics and Statistics, Stony Brook University, jsbm@ams.sunysb.edu

Abstract. We consider the problem of simultaneous embedding of planar graphs. There are two variants of this problem, one in which the mapping between the vertices of the two graphs is given and another in which the mapping is not given. In particular, given a mapping, we show how to embed two paths on an $n \times n$ grid, and two caterpillar graphs on a $3n \times 3n$ grid. We show that it is not always possible to simultaneously embed three paths. If the mapping is not given, we show that any number of outerplanar graphs can be embedded simultaneously on an $O(n) \times O(n)$ grid, and an outerplanar and general planar graph can be embedded simultaneously on an $O(n^2) \times O(n^2)$ grid.

1 Introduction

The areas of graph drawing and information visualization have seen significant growth in recent years [10, 15]. Often the visualization problems involve taking information in the form of graphs and displaying them in a manner that both is aesthetically pleasing and conveys some meaning. The aesthetic criteria alone are the topic of much debate and research, but some generally accepted and tested standards include preferences for straight-line edges or those with only a few bends, a limited number of crossings, good separation of vertices and edges, as well as a small overall area. Some graphs change over the course of time and in such cases it is often important to preserve the "mental map".

Consider a system that visualizes the evolution of software, information can be extracted about the program stored within a CVS version control system [8]. Inheritance graphs, program call-graphs, and control-flow graphs can be visualized as they evolve in time; see Fig. 1. Such tools allow programmers to understand the evolution of a legacy program: Why is the program structured the way it is? Which programmers were responsible for which parts of the program during which time periods? Which parts of the program appear unstable over long periods of time and may need to be rewritten? For such a visualization tool, it is essential to preserve the mental map for the graph under scrutiny. That is, slight changes in the graph structure should not yield large changes in the actual drawing of the

⋆ Partially supported by NSF Grant ACR-0222920.
⋆⋆ Partially supported by Metron Aviation, Inc., NASA Ames Research (NAG2-1325), NSF (CCR-0098172), and the U.S.-Israel Binational Science Foundation.

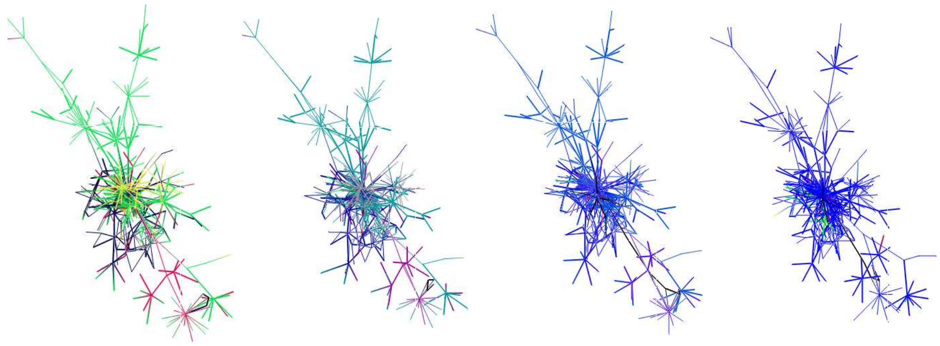

Fig. 1. The inheritance graph of a large Java program as it evolves through time. Different colors indicate different authors. For every time-step that a node does not change, its color fades to blue.

graph. Vertices should remain roughly near their previous locations and edges should be routed in roughly the same manner as before [10, 15].

While graphs that evolve through time are not necessarily planar, solving the planar case can provide intuition and ideas for the more general case. Thus, the focus of the this paper is on the problem of simultaneous embedding of planar graphs. This problem is related to the thickness of graphs; see [18] for a survey. The thickness of a graph is the minimum number of planar subgraphs into which the edges of the graph can be partitioned. Thickness is an important concept in VLSI design, since a graph of thickness k can be embedded in k *layers*, with any two edges drawn in the same layer intersecting only at a common vertex and vertices placed in the same location in all layers. A related graph property is *geometric thickness*, defined to be the minimum number of layers for which a drawing of G exists having all edges drawn as straight-line segments [11]. Finally, the *book thickness* of a graph G is the minimum number of layers for which a drawing of G exists, in which edges are drawn as straight-line segments and vertices are in convex position [2]. It has been shown that the book thickness of planar graphs is no greater than four [21].

As initiated by Cenek and Lubiw [5], we look at the problem almost in reverse. Assume we are given the layered subgraphs and now wish to simultaneously embed the various layers so that the vertices coincide and no two edges of the same layer cross. Take, for example, two graphs from the 1998 Worldcup; see Fig. 2. One of the graphs is a tree illustrating the games played. The other is a graph showing the major exporters and importers of players on the club level. In displaying the information, one could certainly look at the two graphs separately, but then there would be little correspondence between the two layouts if they were created independently, since the viewer has no "mental map" between the two graphs. Using a simultaneous embedding, the vertices can be placed in the exact same locations for both graphs, making the relationships more clear. This is different than simply merging the two graphs together and displaying the information as one large graph.

In simultaneous embeddings, we are concerned with crossings but not between edges belonging to different layers (and thus different graphs). Typical graph drawing algorithms lose all information about the separation of the two graphs and so must also avoid such non-essential crossings. Techniques for displaying simultaneous embeddings can be quite

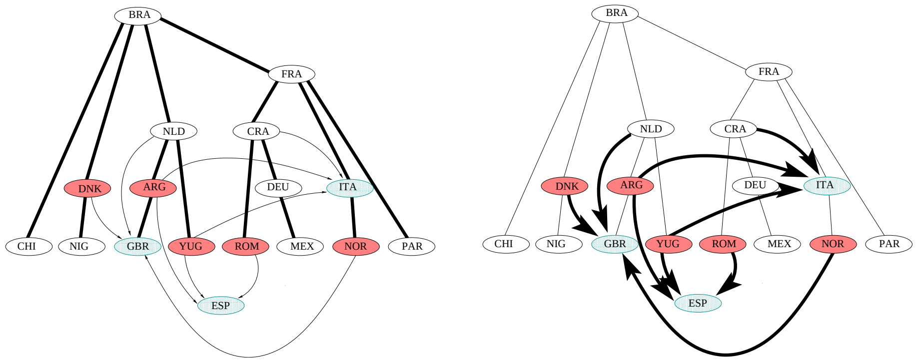

Fig. 2. The vertices of this graph represent the round of 16 teams from Worldcup 1998 (plus Spain). The 8 teams eliminated in the round of 16 are on the bottom; next are the 4 teams eliminated in the quarter-finals, etc.Thick edges in the left drawing indicate matches played. Thick edges in the right drawing indicate export of players on the club level. The light (dark) shaded vertices indicate importers (exporters) of players.

varied. One may choose to draw all graphs simultaneously, employing different edge styles, colors, and thickness for each edge set. One may choose a more three-dimensional approach in order to differentiate between layers. One may also choose to show only one graph at a time and allow the users to choose which graph they wish to see by changing the edge set (without moving the vertices). Finally, one may highlight one set of edges over another, giving the effect of "bolding" certain subgraphs, as in Fig. 2.

The subject of simultaneous embeddings has many different variants, several of which we address here. The two main classifications we consider are embeddings with and without predefined vertex mappings.

Definition 1. Given k planar graphs $G_i = (V, E_i)$ for $1 \leq i \leq k$, *simultaneous (geometric) embedding of* G_i *with mapping* is the problem of finding plane straight-line drawings D_i of G_i such that for every $u \in V$ and any two drawings D_i and D_j, u is mapped to the same point on the plane in all k drawings.

Definition 2. Given k planar graphs $G_i = (V_i, E_i)$ for $1 \leq i \leq k$, *simultaneous (geometric) embedding of* G_i *without mapping* is the problem of finding plane straight-line drawings D_i of G_i such that given any two drawings D_i and D_j there exists a bijective mapping $f : V_i \to V_j$. such that $u \in V_i$ and $v \in V_j$ are mapped to the same point in the plane in both drawings.

Note that in the final drawing a crossing between two edges a and b is allowed *only if* there does not exist an edge set E_i such that $a, b \in E_i$.

In both versions of the problem, we are interested in embeddings that map the vertices to a small cardinality set of candidate vertex locations. Throughout this paper, we make the standard assumption that candidate vertex locations are at integer grid points, so our objective is to bound the size of the integer grids required.

The following table summarizes our current results regarding the two versions under various constraints on the type of graphs given; entries in the table indicate the size of the integer grid required.

Graphs	With Mapping	Without Mapping
G_1: Planar, G_2: Outerplanar	not always possible	$O(n^2) \times O(n^2)$
G_1, G_2: Outerplanar	not always possible	$O(n) \times O(n)$
C_1, C_2: Caterpillar	$3n \times 3n$	$O(n) \times O(n)$ (outerplanar)
C_1: Caterpillar, P_2: Path	$n \times 2n$	$O(n) \times O(n)$ (outerplanar)
P_1, P_2: Path	$n \times n$	$\sqrt{n} \times \sqrt{n}$
C_1, C_2: Cycle	$4n \times 4n$	$\sqrt{n} \times \sqrt{n}$
P_1, P_2, P_3: Path	not always possible	$\sqrt{n} \times \sqrt{n}$

2 Previous Work

Computing straight-line embeddings of planar graphs on the integer grid is a well-studied graph drawing problem. The first solutions to this problem are given by de Fraysseix, Pach and Pollack [9], using a canonical labeling of the vertices in an algorithm that embeds a planar graph on n vertices on the $(2n-4) \times (n-2)$ integer grid and, independently, by Schnyder [19] using the barycentric coordinates method. The algorithm of Chrobak and Kant [7] embeds a 3-connected planar graph on an $(n-2) \times (n-2)$ grid so that each face is convex. Miura, Nakano, and Nishizeki [17] further restrict the graphs under consideration to 4-connected planar graphs with at least four vertices on the outer face and present an algorithm for straight-line embeddings of such graphs on an $(\lceil n/2 \rceil - 1) \times (\lfloor n/2 \rfloor)$ grid.

Another related problem is that of simultaneously embedding more than one planar graph, not necessarily on the same point set. This problem dates back to the circle-packing problem of Koebe [16]. Tutte [20] shows that there exists a simultaneous straight-line representation of a planar graph and its dual in which the only intersections are between corresponding primal-dual edges. Brightwell and Scheinerman [4] show that every 3-connected planar graph and its dual can be embedded simultaneously in the plane with straight-line edges so that the primal edges cross the dual edges at right angles. Erten and Kobourov [13] present an algorithm for simultaneously embedding a 3-connected planar graph and its dual on an $O(n) \times O(n)$ grid.

Bern and Gilbert [1] address a variation of the problem: given a straight-line planar embedding of a planar graph, find suitable locations for dual vertices so that the edges of the dual graph are also straight-line segments and cross only their corresponding primal edges. They present a linear-time algorithm for the problem in the case of convex 4-sided faces and show that the problem is NP-hard for the case of convex 5-sided faces.

3 Simultaneous Embedding With Mapping

We first address the simplest version of the problem: embedding paths.

Theorem 1. *Let P_1 and P_2 be 2 paths on the same vertex set, V, of size n. Then a simultaneous geometric embedding of P_1 and P_2 with mapping can be found in linear time and on an $n \times n$ grid.*

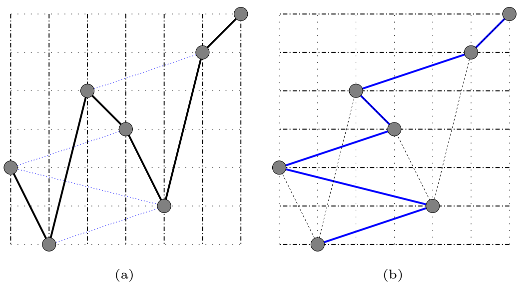

Fig. 3. An example of embedding two paths on an $n \times n$ grid. The two paths are respectively $v_1, v_2, v_3, v_4, v_5, v_6, v_7$ and $v_2, v_5, v_1, v_4, v_3, v_6, v_7$. They are drawn using (a) increasing x-order and (b) increasing y-order.

Proof: For each vertex $u \in V$, we embed u at the integer grid point (p_1, p_2), where $p_i \in \{1, 2, \ldots, n\}$ is the vertex's position in the path P_i, $i \in \{1, 2\}$. Then, P_1 is embedded as an x-monotone polygonal chain, and P_2 is embedded as a y-monotone chain; thus, neither path is self-intersecting. See Fig. 3. $\square$

This method can be extended to handle two cycles, but does not extend to more than two paths. We present these results in turn.

Theorem 2. *Let C_1 and C_2 be 2 cycles on the same vertex set of size n, each with the edges oriented clockwise around an interior face. Then a simultaneous geometric embedding (with mapping) of C_1 and C_2 that respects the orientations can be found in linear time on a $4n \times 4n$ grid, unless the two cycles are the same cycle oppositely oriented. In the latter case no such embedding exists.*

Proof:

Assume that C_1 and C_2 are not the same cycle oppositely oriented. Then there must exist a vertex v such that the predecessor of v in C_1, say a, is different from the successor of v in C_2, say b. Place v at the point $(0, 0)$, and use the simultaneous path drawing algorithm from Theorem 1 to draw the path in C_1 from v to a as an x-monotone path, and the backward path in C_2 from v back to b as a y-monotone path. Then a will be drawn as the point of maximum x coordinate, and b as the point of maximum y coordinate.

Without destroying the simultaneous embedding, we can pull v diagonally to the grid point $(-n, -n)$ and a horizontally out to the right until the line segment av lies completely below the other points. Let c be the predecessor of v in C_2. The line segment cv has slope at least $1/2$. The y-coordinate distance between v and a is at most $2n$. If the x-coordinate distance between v and a is greater than $4n$ then the slope of the segment av becomes less than $1/2$ and and it is below the other points. The same idea applies to b (this time shifting b up vertically) also and we get a grid of total size $4n \times 4n$. $\square$

Theorem 3. *There exist three paths $\mathcal{P} = \bigcup_{1 \leq i \leq 3} P_i$ on the same vertex set V such that at least one of the layers* must *have a crossing.*

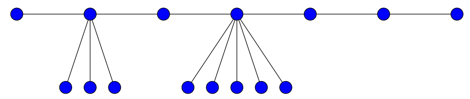

Fig. 4. A caterpillar graph C is drawn with solid edges. The vertices on the top row and the edges between them form the spine. The vertices on the bottom row form the legs of the caterpillar.

Proof: A path of n vertices is simply an ordered sequence of n numbers. The three paths we consider are: 714269358, 824357169 and 758261439. For example, the sequence 714269358 represents the path $(v_7, v_1, v_4, v_2, v_6, v_9, v_3, v_5, v_8)$. We will write ij for the edge connecting v_i to v_j. There are twelve edges in the union of these paths

$$E = \{14, 16, 17, 24, 26, 28, 34, 35, 39, 57, 58, 69\}.$$

It is easy to see that the graph G consisting of these edges is a subdivision of $K_{3,3}$ and therefore non-planar: collapsing 1 and 7, 2 and 8, 3 and 9 yields the classes {1,2,3} and {4,5,6}.

It follows that there are two nonadjacent edges of G that cross each other. It is easy to check that *every pair of nonadjacent edges* from E appears in *at least one* of the paths given above. Therefore, at least one path will cross itself which completes the proof. □

3.1 Caterpillars

A simple class of graphs similar to paths is the class of caterpillar graphs. Let us first define the specific notion of a caterpillar graph.

Definition 3. *A caterpillar graph $C = (V, E)$ is a tree such that the graph obtained by deleting the leaves, which we call the* **legs** *of C, is a path, which we call the* **spine** *of C; see Fig. 4.*

We describe an algorithm to simultaneously embed two caterpillars on a $3n \times 3n$ grid. As a first step in this direction we argue that a path and a caterpillar can be embedded in a smaller area, as the following theorem shows.

Theorem 4. *Given a path P and a caterpillar graph C, we can simultaneously embed them, with mapping, on an $n \times 2n$ grid.*

Proof: We use much the same method as embedding two paths, with one exception: we allow some vertices to share the same x-coordinate. Let S and L, respectively, denote the spine and the legs of C. For a vertex v let $o_p(v)$ denote v's position in P. If v is in S, then let $o_c(v)$ be its position in S and place v initially at the location $(2o_c(v), o_p(v))$. Otherwise, if $v \in L$, let $o_c(v) = o_c(p(v))$ be its parent's position and initially place v at the location $(2o_c(v) + 1, o_p(v))$.

We now proceed to attach the edges. By preserving the y-ordering of the points, we guarantee that the path has no crossings. In our embedding, we may need to shift, but we shall only perform right shifts. That is, we shall push points to the right of a vertex v by one unit right, in essence inserting one extra grid location when necessary. Note that this step still preserves the y-ordering.

To attach the caterpillar edges, we march along the spine. Let $L(u)$ denote the legs of a vertex u in the spine S. If we do not consider any edges of S then all the legs can be drawn with straight-line edges and no crossings by the initial placement. Now when we attach an edge from u to v on the spine, where $u, v \in S$, it is not planar if and only if there exists $w \in L(u)$ that is collinear with u and v. In this case, we simply shift v and all succeeding points by one unit to the right. We continue the right shift until none of the legs is collinear with u and v. Now, the edge from u to v on the spine is no longer collinear with other vertices. This right shift does not affect the planarity of the legs since the relative x-coordinates of the vertices are still preserved. The number of shifts we made is bounded by $|L(u)|$.

We continue in this manner until we have attached all edges. Let k be the total number of legs of the caterpillar. Then the total number of shifts made is k. Since we initially start with $2 \times (n-k)$ columns in our grid, the total number of columns necessary is $2n-k$. Thus, in the worst case the grid size needed is less than $2n \times n$. □

The algorithm for embedding two caterpillars is also similar but before we can prove our main result for caterpillars, we need an intermediary theorem. In order to embed two caterpillars, we allow shifts in two directions. Let $C_1 = (V, E_1)$ and $C_2 = (V, E_2)$ be two caterpillars. Denote the vertices on the spine of C_1 (C_2) with S_1 (S_2). Let $L_1(u)$ ($L_2(u)$) denote the legs of $u \in S_1$ (S_2). Let $\mathcal{T}_1$ ($\mathcal{T}_2$) be a fixed traversal order of vertices on S_1 (S_2). Let $u(X)$ and $u(Y)$ denote the x-coordinate and y-coordinate of the vertex u, respectively. We will place the vertices such that the following *initial placement invariants* hold:

1. For any $u, v \in V$, $u(X) \neq v(X)$ and $u(Y) \neq v(Y)$.
2. If $u \in S_1$ appears before $v \in S_1$ in $\mathcal{T}_1$ then $u(X) < w(X) < v(X)$ where $w \in L_1(u)$. If $u \in S_2$ appears before $v \in S_2$ in $\mathcal{T}_2$ then $u(Y) < w(Y) < v(Y)$ where $w \in L_2(u)$.
3. The set of vertices belonging to $L_1(u)$ that are above (below) $u \in S_1$ are monotonically increasing in the x-coordinate, and monotonically decreasing (increasing) in the y-coordinate. Similarly for C_2, the set of vertices belonging to $L_2(u)$ that are to the left (right) of $u \in S_2$ are monotonically increasing in the x-coordinate, and monotonically increasing (decreasing) in the y-coordinate.

Theorem 5. *The initial placement can be done on an $n \times n$ grid.*

Proof. We start by assigning x-coordinates of the vertices in S_1 by following the order in $\mathcal{T}_1$. The first vertex is assigned 1. We assign $v(X) = u(X) + |L_1(u)| + 1$ where $v \in S_1$ follows $u \in S_1$ in $\mathcal{T}_1$. Similarly we assign y-coordinates of the vertices in S_2, i.e., the first vertex is assigned 1 and $v(Y) = u(Y) + |L_2(u)| + 1$ where $v \in S_2$ follows $u \in S_2$ in $\mathcal{T}_2$.

Next we assign the x-coordinates of the vertices in $L_1(u)$ for each $u \in S_1$. We sort the vertices in $L_1(u)$ based on their y-coordinate distance from u in descending order. For each $w \in L_1(u) \cup \{u\}$, if $w \in S_2$, we use $w(Y)$ for comparison while sorting otherwise $w \in L_2(w')$ for some $w' \in S_2$ and we use $w'(Y) + 1$. Following this sorted order we assign $u(X) + 1$, $u(X) + 2$, ... to each vertex in $L_1(u)$. While sorting we use the same y-coordinate for two vertices $r, r' \in L_1(u)$ only if $r, r' \in L_2(v)$. In this case their x-coordinates get randomly

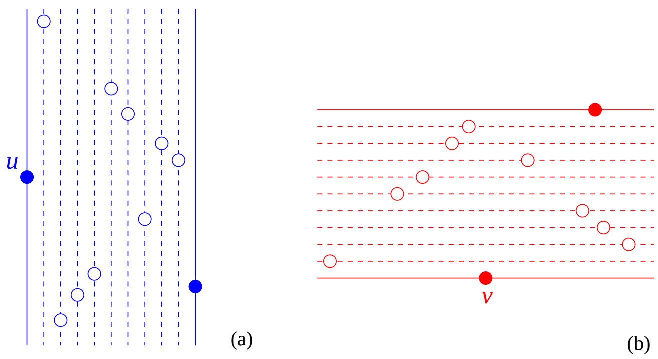

Fig. 5. a) Arrangement of $u \in S_1$ and $L_1(u)$. The legs of u are shown with empty circles. The x-coordinate of each vertex in $L_1(u)$ is determined by its vertical distance from u. b) Arrangement of $v \in S_2$ and $L_2(v)$. The legs of v are shown with empty circles. The y-coordinate of each vertex in $L_2(v)$ is determined by its horizontal distance from v.

assigned. However, this is not a problem, since the y-coordinate calculation of the legs in C_2 takes into account the x-coordinates we just calculated, and both the coordinates will then be compatible in terms of the initial placement invariants above. We similarly calculate the y-coordinates of the vertices in $L_2(v)$, but this time considering the exact x-coordinates we just calculated for comparison in sorting.

After the initial placement we get the arrangement in Fig. 5. It is easy to see that with the initial placement invariants satisfied, for any $u \in S_1$ (S_2), any leg $w \in L_1(u)$ ($L_2(u)$) is visible from u and if we do not consider the edges on the spine, C_1 (C_2) is drawn without crossings.

Theorem 6. *Let C_1 and C_2 be 2 caterpillars on the same vertex set of size n. Then a simultaneous geometric embedding of C_1 and C_2 with mapping can be found on a $3n \times 3n$ grid.*

Proof: In the initial placement, a spine edge between $u, v \in S_1$ is not planar if and only if a vertex $w \in L_1(u)$ is collinear with u and v. We can avoid such collinearities while ensuring that no legs are crossing by shifting some vertices up/right. The idea is to grow a rectangle starting from the bottom-left corner of the grid, and to make sure that parts of C_1 and C_2 that are inside the rectangle are always non-crossing. This is achieved through additional shifting of the vertices up/right.

First we make the following observation regarding the shifting:

Observation: Given a point set arrangement that satisfies the initial placement invariants, shifting any vertex $u \in V$ and all the vertices that lie above (to the right of) u up (right) by one unit preserves the invariants.

Since shifting a set of points up, starting at a certain y-coordinate, does not change the relative positions of the points, the invariants are still preserved.

Fig. 6. Given the above mapping between the vertices; the outerplanar graphs O_1 and O_2 can not be embedded simultaneously.

We start out with the rectangle $\mathcal{R}_1$ such that the bottom-left corner of $\mathcal{R}_1$ is the bottom-left corner of the grid and the upper-right corner is the location of the closest vertex u, where $u \in S_1$ or $u \in S_2$. Since no other vertices lie in $\mathcal{R}_1$, the parts of C_1, C_2 inside $\mathcal{R}_1$ are non-crossing.

Now assume that after the k^{th} step of the algorithm, the parts of the caterpillars lying inside $\mathcal{R}_k$ are planar. We find the closest vertex v, to $\mathcal{R}_k$, where $v \in S_1$ or $v \in S_2$. There are two cases.

- *Case 1:* v is above $\mathcal{R}_k$, i.e., $x(v)$ is between the x-coordinate of the left edge and right edge of the rectangle. Enlarge $\mathcal{R}_k$ in the *y-direction* so that v lies on the top edge of the rectangle, and call the new rectangle $\mathcal{R}_{k+1}$. Let u (u') be the spine vertex before (after) v in $\mathcal{T}_1$. Let w (w') be the spine vertex before (after) v in $\mathcal{T}_2$. If any one of u, u', w, or w' lies inside $\mathcal{R}_{k+1}$ we check if v is visible from that vertex. If not, we shift v one unit up and enlarge $\mathcal{R}_{k+1}$ accordingly.
- *Case 2:* v is not above $\mathcal{R}_k$. If v is to the right of $\mathcal{R}_k$ we enlarge it in the *x-direction* so that v lies on the right edge of the rectangle, otherwise we enlarge it in both x and y directions so that v lies on the top-right corner. We call the new rectangle $\mathcal{R}_{k+1}$. As in *Case 1*, we check for the visibility of the neighboring vertices along the spines, but in this case we perform a right shift and enlarge $\mathcal{R}_{k+1}$ in the *x-direction* accordingly, if we encounter any collinearities.

When we perform an up/right shift, we do not make any changes inside the rectangle, so the edges drawn inside the rectangle remain non-crossing. Each time we perform a shift we eliminate a collinearity between the newly added vertex v and the vertices lying inside the rectangle. Hence, after a number of shifts all the collinearities involving v and such vertices inside the rectangle will be resolved, and all the edges inside our new rectangle, including the edges involving the new vertex v are non-crossing.

¿From the above observation shifting the vertices does not violate the initial placement invariants and so the legs of the caterpillars remain non-crossing throughout the algorithm.

Since each leg (in C_1 or C_2) contributes to at most one shifting, the size of the grid required is $(n + k_1) \times (n + k_2)$, where $(k_1 + k_2) < 2n$, thus yielding the desired result. □

3.2 Outerplanar Graphs

Simultaneous embedding of outerplanar graphs is not always possible.

Theorem 7. *There exist two outerplanar graphs which, given a mapping between the vertices of the graphs, cannot be simultaneously embedded.*

Proof: The two outerplanar graphs O_1, O_2 are as shown in Figure 6. The union of O_1, and O_2 contains $K_{3,3}$ as a subgraph, which means that when embedded simultaneously the edges of the two graphs contain at least one intersection. Assume O_1 and O_2 can be simultaneously embedded. Then the crossing in the union of the two graphs must be between an edge of O_1 and an edge of O_2. The edges belonging to O_1 only are 12 and 36. The edges belonging to O_2 only are 23 and 16. However, we can not pick a crossing pair out of these, since each such pairing consists of incident edges which can not cross. Thus there must be another pair (either in O_1 or in O_2 which intersects. $\square$

4 Simultaneous Embedding Without Mapping

In this section we present methods to embed different classes of planar graphs simultaneously when no mapping between the vertices are provided. For the remainder of this section, when we say simultaneous embeddings we always mean without vertex mappings. This additional freedom to choose the vertex mapping does make a great difference. For example, any number of paths or cycles can be simultaneously embedded. Indeed, in this setting of simultaneous embedding without vertex mappings we do not have any non-embeddability result; it is perhaps the most interesting open question whether any two planar graphs can be simultaneously embedded. We do have a positive answer if all but one of the planar graphs are outerplanar.

Theorem 8. *A planar graph G_1 and any number of outerplanar graphs $G_2, \ldots, G_r$, each with n vertices, can be simultaneously embedded (without mapping) on an $O(n^2) \times O(n^2)$ grid.*

Theorem 9. *Any number of outerplanar graphs can be simultaneously embedded (without mapping) on an $O(n) \times O(n)$ grid.*

Key to the proof of both theorems is the construction of grid subsets in general position, since it is known that any outerplanar graph can be embedded on any point set in general position (no three points collinear):

Theorem 10. *[3, 14] Given a set P of n points in the plane, no three of which are collinear, an outerplanar graph H with n vertices can be straight-line embedded on P.*

These embeddings can even be found efficiently. Gritzmann *et al* [14] provide an embedding algorithm for such graphs that runs in $O(n^2)$ time, and Bose [3] further reduces the running time to $O(n \lg^3 n)$.

Theorem 9 then follows from the existence of sets of n points in general position in an $O(n) \times O(n)$ grid. But this is an old result by Erdös [12]: choose the minimum prime number p greater than n (there is a prime between n and $(1+\varepsilon)n$ for $n > n_0(\varepsilon)$), then the points $(t, t^2 \bmod p)$ for $t = 1, \ldots, p$ are a set of $p \geq n$ points in the $p \times p$-grid with no three points collinear. So we can choose the required points in a $(1+\varepsilon)n \times (1+\varepsilon)n$-grid.

The smallest grid size in which one can choose n points in general position is known as the 'no-three-in-line'-problem; the only lower bound is $\frac{1}{2}n \times \frac{1}{2}n$, below that there are already three points in the same row or column.

In order to prove Theorem 8, we must embed an arbitrary planar graph, G_1, in addition to the outerplanar graphs; unlike outerplanar graphs, we cannot embed G_1 on any point set in general position. Thus, we begin by embedding G_1 in an $O(n) \times O(n)$ grid using the algorithm of [6]. The algorithm draws any 3-connected planar graph in an $O(n) \times O(n)$ grid under the *edge resolution rule*, and produces a drawing of that graph with the special property that for each vertex and each edge not incident with this vertex, the distance between the vertex and the edge in the embedding is at least one grid unit. This embedding may still contain many collinear vertices; we resolve this in the next step. We again choose the smallest prime $p \geq n$, and blow up the whole drawing by a factor of $2p$, mapping a previous vertex at (i, j) to the new location $(2pi, 2pj)$. In this blown-up drawing, the distance between a vertex and a non-incident edge is at least $2p$. Now let v_1v_2 be an edge in that drawing, w a vertex not incident to that edge, and let v_1', v_2', w' be arbitrary grid points from the small $p \times p$-grids centered at v_1, v_2, w. Then the distance of v_1', v_2', w' to v_1, v_2, w is at most $\frac{1}{\sqrt{2}}p$, so the distance of w' to the segment $v_1'v_2'$ is at least $(2 - \frac{2}{\sqrt{2}})p > 0$. Thus, any perturbation of the blown-up drawing, in which each vertex v is replaced by some point v' from the $p \times p$-grid centered at v, will still have the same combinatorial structure, and still be a valid plane drawing. We now choose a special such perturbation to obtain a general-position set: If the vertex v_ν was mapped by the algorithm of [6] on the point (i, j), then we map it on the point $(2pi + (\nu \bmod p), 2pj + (\nu^2 \bmod p))$. This new embedding is still a correct embedding for the planar graph, since all vertices have still sufficient distance from all non-incident edges. Further, it is a general-position point set, suitable for the embedding of outerplanar graphs, since by a reduction modulo p the points are mapped on the general-position point set $\{(\nu, \nu^2 \bmod p) : \nu = 1, \dots, n\}$, and collinearity is a property that is preserved by the $\bmod\, p$-reduction of the coordinates. So we have embedded the planar graph in an $O(n^2) \times O(n^2)$ grid, on a point set in general position, on which now all outerplanar graphs can also be embedded. This completes the proof of Theorem 8.

5 Open Problems

- Can 2 lobster graphs[4] or 2 trees be simultaneously embedded with mapping? We have answered affirmatively for the special case of 2 caterpillars.
- Given a general planar graph G, and a path P with two or more vertices, can we always simultaneously embed with mapping G and P ?
- While, in general, it is not always possible to simultaneously embed (with mapping) two arbitrary planar graphs, can we test in polynomial time whether two particular graphs can be embedded for a given mapping?
- Can any two planar graphs be simultaneously embedded without mapping?

[4] A lobster graph is a tree such that the graph obtained by deleting the leaves is a caterpillar.

6 Acknowledgments

We would like to thank Ed Scheinerman for stimulating discussions about different variations of the problem and Esther M. Arkin for her proof of Theorem 3 (independent of our work).

References

1. M. Bern and J. R. Gilbert. Drawing the planar dual. *Information Processing Letters*, 43(1):7–13, Aug. 1992.
2. F. Bernhart and P. C. Kainen. The book thickness of a graph. *J. Combin. Theory*, Ser. B 27:320–331, 1979.
3. P. Bose. On embedding an outer-planar graph in a point set. *CGTA: Computational Geometry: Theory and Applications*, 23(3):303–312, 2002.
4. G. R. Brightwell and E. R. Scheinerman. Representations of planar graphs. *SIAM Journal on Discrete Mathematics*, 6(2):214–229, May 1993.
5. E. Cenek. *Layered and Stratified Graphs*. PhD thesis, University of Waterloo, forthcoming.
6. M. Chrobak, M. T. Goodrich, and R. Tamassia. Convex drawings of graphs in two and three dimensions. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 319–328, 1996.
7. M. Chrobak and G. Kant. Convex grid drawings of 3-connected planar graphs. *Intl. Journal of Computational Geometry and Applications*, 7(3):211–223, 1997.
8. C. Collberg, S. G. Kobourov, J. Nagra, J. Pitts, and K. Wampler. A system for graph-based visualization of the evolution of software. In *1st ACM Symposium on Software Visualization*. To appear in 2003.
9. H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.
10. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, Englewood Cliffs, NJ, 1999.
11. 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.
12. P. Erdös. Appendix. In K. F. Roth. On a problem of Heilbronn. *J. London Math. Soc.*, 26:198–204, 1951.
13. C. Erten and S. G. Kobourov. Simultaneous embedding of a planar graph and its dual on the grid. In *13th Intl. Symp. on Algorithms and Computation (ISAAC)*, pages 575–587, 2002.
14. P. Gritzmann, B. Mohar, J. Pach, and R. Pollack. Embedding a planar triangulation with vertices at specified points. *American Math. Monthly*, 98:165–166, 1991.
15. M. Kaufmann and D. Wagner. *Drawing graphs: methods and models*, volume 2025 of *Lecture Notes in Computer Science*. Springer-Verlag Inc., New York, NY, USA, 2001.
16. P. Koebe. Kontaktprobleme der konformen Abbildung. *Berichte "uber die Verhandlungen der Sächsischen Akademie der Wissenschaften zu Leipzig. Math.-Phys. Klasse*, 88:141–164, 1936.
17. K. Miura, S.-I. Nakano, and T. Nishizeki. Grid drawings of 4-connected plane graphs. *Discrete and Computational Geometry*, 26(1):73–87, 2001.
18. P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of graphs: a survey. *Graphs Combin.*, 14(1):59–73, 1998.
19. W. Schnyder. Planar graphs and poset dimension. *Order*, 5(4):323–343, 1989.
20. W. T. Tutte. How to draw a graph. *Proc. London Math. Society*, 13(52):743–768, 1963.
21. M. Yannakakis. Embedding planar graphs in four pages. *Journal of Computer and System Sciences*, 38(1):36–67, Feb. 1989.