Available online at www.sciencedirect.com
ScienceDirect
Computational Geometry 36 (2007) 117–130
Computational Geometry
Theory and Applications
www.elsevier.com/locate/comgeo

On simultaneous planar graph embeddings ☆

Peter Brass [a], Eowyn Cenek [b], Cristian A. Duncan [c], Alon Efrat [d], Cesim Erten [d,*], Dan P. Ismailescu [e], Stephen G. Kobourov [d,1], Anna Lubiw [b], Joseph S.B. Mitchell [f,2]

[a] *Department of Computer Science, City College of New York, USA*
[b] *Department of Computer Science, University of Waterloo, USA*
[c] *Department of Computer Science, University of Miami, USA*
[d] *Department of Computer Science, University of Arizona, USA*
[e] *Department of Mathematics, Hofstra University, USA*
[f] *Department of Applied Mathematics and Statistics, Stony Brook University, USA*

Received 22 March 2004; received in revised form 15 May 2006; accepted 15 May 2006

Available online 30 June 2006

Communicated by Roberto Tamassia

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. We present positive and negative results for the two versions of the problem. Among the positive results with given mapping, we show that we can embed two paths on an $n \times n$ grid, and two caterpillar graphs on a $3n \times 3n$ grid. Among the negative results with given mapping, we show that it is not always possible to simultaneously embed three paths or two general planar graphs. 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.

Keywords: Graph drawing; Simultaneous visualizations; Planar graphs

1. Introduction

The areas of graph drawing and information visualization have seen significant growth in recent years [10,16]. 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

☆ An extended abstract of this paper appeared in WADS 2003.

* Corresponding author.

E-mail addresses: peter@cs.ccny.cuny.edu (P. Brass), ewcenek@uwaterloo.edu (E. Cenek), duncan@cs.miami.edu (C.A. Duncan), alon@cs.arizona.edu (A. Efrat), cesim@cs.arizona.edu (C. Erten), dan.p.ismailescu@hofstra.edu (D.P. Ismailescu), kobourov@cs.arizona.edu (S.G. Kobourov), alubiw@uwaterloo.edu (A. Lubiw), jsbm@ams.sunysb.edu (J.S.B. Mitchell).

1 Partially supported by NSF Grant ACR-0222920.

2 Partially supported by Metron Aviation, Inc., NASA Ames Research (NAG2-1620), and the National Science Foundation (CCF-0431030, CCF-0528209).

0925-7721/$ – see front matter
doi:10.1016/j.comgeo.2006.05.006

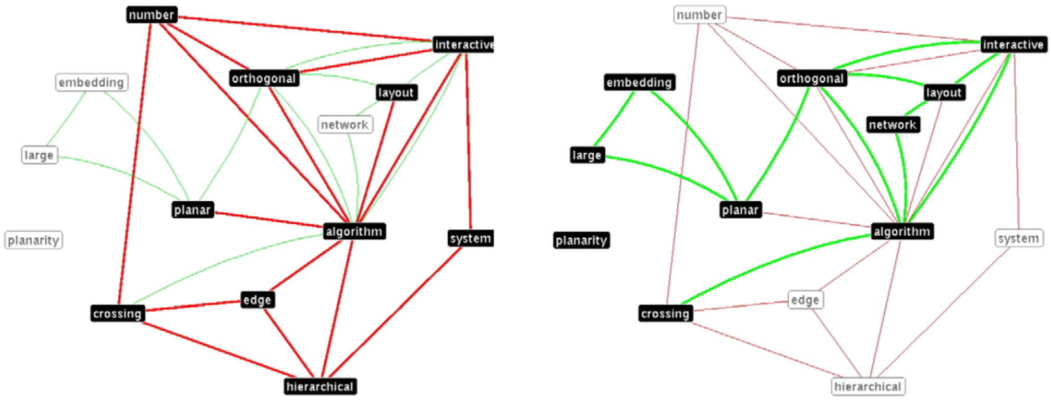


Fig. 1. The vertices of these graphs represent the most popular keywords from the graph drawing literature for two time intervals: 1996–1998 and 1998–2000. Black vertices and thick edges in the left/right drawing are from the first/second period.

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. Often, we have a series of related graphs that we would like to compare visually. These graphs may come from social network analysis, where different relationships among the same set of people are being studied. Similarly, in biology, different algorithms produce different philogenetic trees on the same set of organisms. Finally, we may be studying an evolving relationship, represented by graphs that change over the course of time.

While in some of the above examples the graphs 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 [19] 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 [23]. Recently, Duncan et al. [12] have shown that graphs with maximum degree three have geometric thickness two, using simultaneous embedding techniques.

As initiated by Cenek [6], 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. If we do not require straight-line edges, this can be easily achieved using edges with bends. This follows from a result by Pach and Wenger [20], who show that a planar graph can be realized in the plane with fixed (given) vertex locations and with edges that do not cross but may have bends; however, the number of bends per edge may be linear in the size of the graph, which makes such drawings difficult to read.

Take, for example, two keyword-graphs from the graph drawing literature. A keyword-graph for a given time interval has as vertices the keywords from papers in the time interval. Two vertices in a keyword-graph are connected by an edge if the two keywords appear in the same paper. Fig. 1 shows two pieces of the keyword-graphs for the intervals 1996–1998 and 1998–2000, respectively. 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). Techniques for displaying simultaneous embeddings can be quite 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, as in Fig. 2. One may also choose

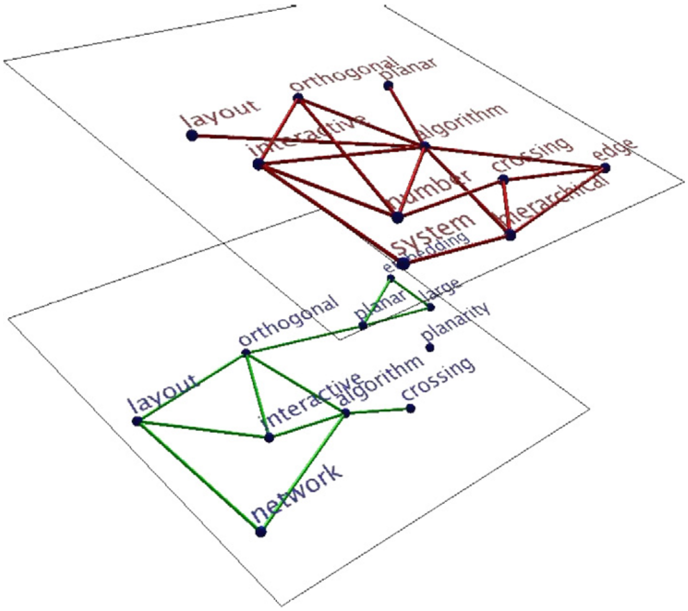


Fig. 2. The keyword-graph from Fig. 1 but this time displayed in 3D with each graph in its own 2D plane. In order to achieve better readability for each individual graph we place the matching vertices in close proximity rather than at the exact same locations.

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. 1.

2. Previous work

Realizing straight-line embeddings of planar graphs on the integer grid is a well-studied 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 [21] using a barycentric coordinates method. The algorithm of Chrobak and Kant [8] embeds a 3-connected planar graph on an $(n-2) \times (n-2)$ grid so that each face is convex. Miura et al. [18] 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 [17]. Tutte [22] 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 [14] 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. Our contributions

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.

Table 1

Graphs	With mapping	Without mapping
G_1: Planar, G_2: Outerplanar	not always possible	$O(n^2) \times O(n^2)$
G_1: Planar, P_2: Path	not always possible	$O(n^2) \times O(n^2)$
$G_1, G_2, \ldots, G_k$: 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)
$S_1, S_2, \ldots, S_k$: Star	$O(c^k\sqrt{n}) \times O(c^k\sqrt{n})$	$O(\sqrt{n}) \times O(\sqrt{n})$
$\mathcal{X}_1$: Extended star, P_2: Path	$O(n^2) \times O(n)$	$O(n) \times O(n)$ (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	$\sqrt{n} \times \sqrt{n}$

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

Definition 2. Given k planar graphs $G_i = (V_i, E_i)$ for $1 \leqslant i \leqslant k$, *simultaneous (geometric) embedding of G_i without mapping* is the problem of finding plane straight-line drawings D_i of G_i such that all drawings use the same set of n points in the plane as the points to which vertices are mapped.

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.

Table 1 summarizes our current results regarding the two versions under various constraints on the type of graphs given; entries in Table 1 indicate the size of the integer grid required.

4. Simultaneous embedding with mapping: Positive results

In this section we provide algorithms to simultaneously embed two graphs given a mapping between the vertices.

4.1. Paths and cycles

We first address the simplest version of the problem: simultaneously embedding two paths. While the main idea and the algorithm are straightforward, they form the basis for several of the algorithms that follow.

Theorem 3. *Let P_1 and P_2 be two 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 on an $n \times n$ grid.*

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. □

The above method can be easily extended to handle two cycles.

Theorem 4. *Let C_1 and C_2 be two 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.*

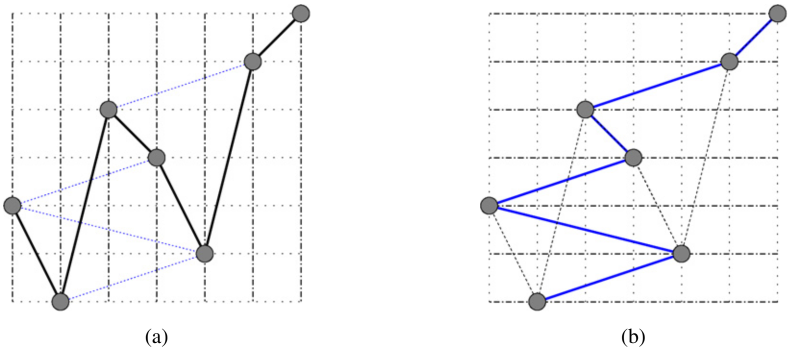


Fig. 3. An example of embedding two paths on an $n \times n$ grid. The two paths are respectively $v_1, v_2, v_3, v_4, v_5, v_6, v_7$ and $v_2, v_5, v_1, v_4, v_3, v_6, v_7$. They are drawn using (a) increasing x-order and (b) increasing y-order.

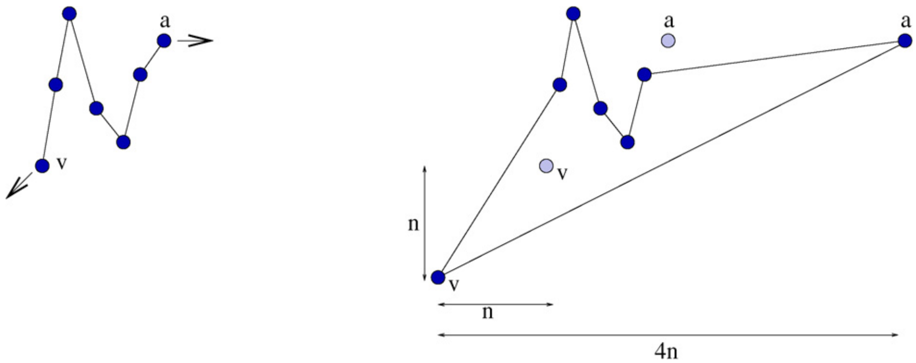


Fig. 4. Vertex v is shifted diagonally and a is shifted horizontally, so that the edge av does not cross any of the existing edges.

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 3 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, see Fig. 4. 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 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 $4n \times 4n$. □

4.2. Stars and extended stars

In this section we extend the results from paths and cycles to stars, brooms and extended stars; see Fig. 5. We begin with simultaneously embedding multiple stars.

Definition 5. A star $S = (V, E)$ is a connected graph in which each vertex except a single root vertex v_r has degree 1.

Theorem 6. *Given k stars $S_1, S_2, \ldots, S_k$, where k is a constant, we can simultaneously embed them with mapping on an $O(c^k\sqrt{n}) \times O(c^k\sqrt{n})$ grid.*

Proof. In an integer lattice (square grid), a point p is *visible* from the origin if no other lattice point lies on the line segment from the origin to p; we then say that p is a *primitive lattice point*. Let $\mathcal{X}$ be the set of primitive lattice points

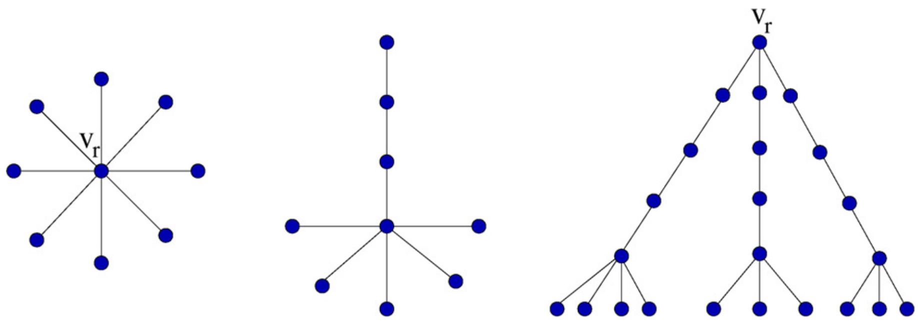


Fig. 5. A star, a broom, and an extended star respectively from left to right.

in the integer lattice. It is known [5] that $\mathcal{X}$ has density $c = \frac{6}{\pi^2}$ in the sense that for each $\epsilon > 0$ there is an n_0 such that for all $n > n_0$ the lattice square $\{-n, \dots, n\}^2$ contains at least $(c - \epsilon)4n^2$ primitive lattice points.

We choose point p (fixed) from the square $\{-a, \dots, a\}^2$ and translation t uniformly from the square $\{-b, \dots, b\}^2$ (finite set, so uniform distribution is no problem). Then the point $p + t$ is in the square $\{-(a + b), \dots, (a + b)\}^2$. Assume b much larger than a, and look at the square $\{-(b - a), \dots, (b - a)\}^2$. For each point q in that square, there is a unique t in $\{-b, \dots, b\}^2$ such that $p + t = q$. Among the $4(b - a)^2$ possible points q, there are at least $(c - \epsilon)4(b - a)^2$ primitive points (if $b - a$ is larger than $n_0(\epsilon)$). So among the $4b^2$ possible choices of t, there are at least $(c - \epsilon)4(b - a)^2$ choices for which $p + t$ is a primitive lattice point. So for uniform distribution of t in that lattice square $\{-b, \dots, b\}^2$ the probability that a fixed point p will give $p + t$ primitive is at least $(c - \epsilon)(1 - (a/b))^2$.

We can repeat the same procedure several times, for $t_1, \dots, t_k$ chosen independently at random from the square. Since the events $p + t_i$ are independent, all events will happen simultaneously with probability $((c - \epsilon)(1 - (a/b))2)^k$, in which case $p + t_i$ is primitive for each t_i. It is, however, possible that several t_i coincide, in which case we reject each point p. The translations $t_1, \dots, t_k$ are chosen independently and uniformly from the $4b2$ elements of $\{-b, \dots, b\}^2$, so the probability of at least two of these choices coinciding is less than $\frac{1}{4b2}k2$. Thus, the probability that for a fixed point p all the $p + t_i$ are primitive and no two t_i coincide is at least $((c - \epsilon)(1 - (a/b))2)^k - \frac{k2}{4b2}$.

Now we can do this for each of the $4a2$ possible choices of p in that square $\{-a, \dots, a\}^2$. The expected number of points p for which this holds is just $4a2$ times the probability for a single p that it holds. So the expected size of the subset of $\{-a, \dots, a\}^2$ of points p for which the translates $p + t_i$, $i = 1, \dots, k$, are all primitive, and no two t_i coinciding, is at least $4a2((c - \epsilon)^k(1 - (a/b))^k - \frac{k2}{4b2})$.

What we need for our construction are the points p and the points $(-t_i)$ (these are the centers of the stars); so the lattice square in which these points are contained is $\{-b, \dots, b\}^2$. If we are not particularly interested in the constants, the simplest choice is $b = 2a$, $a > n_0(\epsilon)$. Then we have among the $4b2 = 16a2$ points of that square a set of at least $4a2((c - \epsilon)^k(1/2)^k - \frac{k2}{16a2}) = 4a2((c - \epsilon)^k(1/2)^k - o(1))$ points p, and k distinct points $(-t_i)$, such that all differences $p + t_i$ are primitive. As this is a positive fraction of all lattice points in the square we are done. □

Definition 7. A broom $B = (V, E)$ is a connected graph that is the union of a path P and a star S.

Definition 8. An extended star $\mathcal{X} = (V, E)$ is a connected graph with a special root vertex v_r such that v_r is connected to the start of each of the path sections of the brooms in a forest of brooms.

We describe an algorithm to simultaneously embed an extended star with a path using an $O(n^2) \times O(n)$ grid.

Theorem 9. *Given a path P and an extended star $\mathcal{X}$, we can simultaneously embed P and $\mathcal{X}$ with mapping on an $O(n^2) \times O(n)$ grid.*

Proof. Let $P = (v_1, v_2, \dots, v_r, v_{r+1}, \dots, v_n)$ be the path where v_r is the root of the extended star $\mathcal{X}$. We first embed the path $P' = (v_{r+1}, v_{r+2}, \dots, v_n, v_1, \dots, v_r)$ with $\mathcal{X}$ simultaneously and then connect v_r to v_{r+1} without causing any intersections in P'.

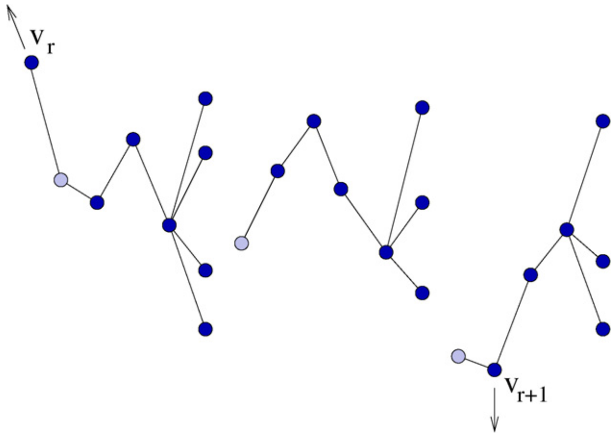


Fig. 6. Extended star of Fig. 5 before shifting: v_r has to see the beginning vertex of each of the path sections of the brooms, drawn with dotted circles, and the vertex v_{r+1}. In order to do this we shift v_{r+1} down and v_r up and to the left.

We embed P' in a y-monotone fashion as usual so that vertex v_{r+1} has y-coordinate 1, v_{r+2} has y-coordinate 2 and so on. Once we have determined the y-coordinates of vertices this way, we embed the forest of brooms by finding out the x-coordinates. We place v_r at the leftmost position, and assign increasing x-coordinates for the vertices of the path section of a broom B and finally the same x-coordinate for the vertices of the star section of B and go on to the next broom. Note that this way the path section of a broom is embedded to the left of the star section. Once we are done, we have a simultaneous embedding of P' and $\mathcal{X}$, except that v_r is not connected to the start of any of the path sections of the brooms but the initial one. In addition we need to complete the embedding of P by connecting v_r to v_{r+1}. In order to achieve these two goals we shift v_r to the left by n, up by $2n^2$, and v_{r+1} down by $3n^2$; see Fig. 6.

Now in the extended star, the slope of the line going through the starting vertex of a path and any leaf of a star section of a previous broom is smaller than n. The x-coordinate distance of v_r to any start vertex of a path is less than $2n$ so that the slope of the edge between v_r and such a vertex is greater than n. The newly drawn edges in the extended star do not cross any of the previous edges of $\mathcal{X}$. To see that P does not have any crossings after adding an edge between v_r and v_{r+1}, let d be the x-coordinate distance between v_{r+1} and the left corner of the original grid before shifting. The slope of the edge between v_{r+1} and the bottom left corner of the original grid is $(3n^2)/d$. The slope of the newly added edge between v_r and v_{r+1} is at most $(5n^2+n)/(n+d)$ which is smaller than $(3n^2)/d$ when $d \leqslant n$. The final area of the grid is $O(n^2) \times O(n)$. $\square$

4.3. Caterpillars

Another 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 10. 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. 7.

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 11. *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

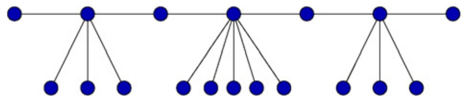


Fig. 7. A caterpillar graph C is drawn with solid edges. The vertices on the top row and the edges between them form the spine. The vertices on the bottom row form the legs of the caterpillar.

$(2o_c(v), o_p(v))$. 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 monotonically increasing y-order of the vertices, we guarantee that the path has no crossings. To ensure that edges can be safely added we may shift vertices to the right when extra space is needed. Note that this step still preserves the y-ordering.

To attach the caterpillar edges, we travel 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 in 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 successive vertices 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 along 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 at most k. Since we initially start with $2(n-k)$ 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 $2n \times n$. $\square$

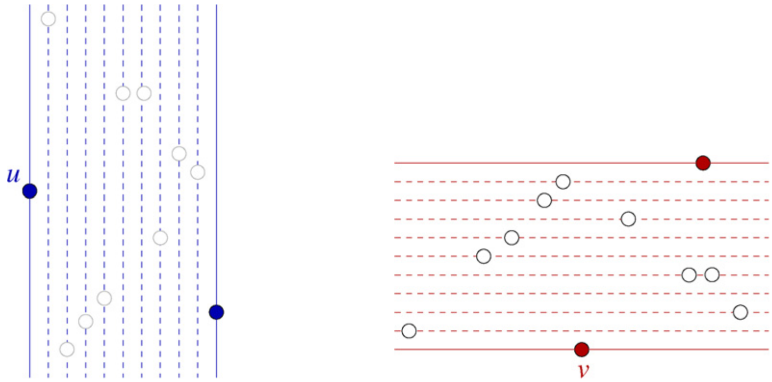
The algorithm for embedding two caterpillars is similar but before we can prove our main result for two caterpillars, we need an intermediate theorem. In order to embed two caterpillars, we allow shifts in two directions. Let $C_1 = (V, E_1)$ and $C_2 = (V, E_2)$ be two caterpillars. As before, let S_1 and S_2 and L_1 and L_2 be the spines and legs of C_1 and C_2. Let $\mathcal{T}_\infty$ and $\mathcal{T}_\in$ be fixed traversal orders of the vertices along the spines S_1 and 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 so 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 non-increasing (non-decreasing) 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 non-decreasing (non-increasing) in the y-coordinate.

Theorem 12. *An initial placement can be obtained 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 x-coordinate 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 y-coordinate 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 $u(X) + 1, u(X) + 2, \ldots$ to the vertices 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 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. For assigning the y-coordinates of the vertices in $L_2(v)$, we first partition its vertices such that $r, r' \in L_2(v)$ are in the same partition if and only if $r, r' \in L_1(u)$ for some $u \in S_1$. We now calculate the y-coordinates of these partitions in $L_2(v)$ similar



(a) (b)

Fig. 8. (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.

to the x-coordinate calculation above (taking the x-coordinate of a random vertex in the partition for comparison in sorting), but this time considering the exact x-coordinates we just calculated. □

After the initial placement we get the arrangement in Fig. 8. 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 along the spine, C_1 (C_2), is drawn without crossings.

Theorem 13. *Let C_1 and C_2 be two 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.

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 kth 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. □

5. Simultaneous embedding with mapping: Negative results

There exist classes of planar graphs that cannot be simultaneously embedded. In this section we provide several examples of two or more graphs that are not simultaneously embeddable. We have developed two different methods for arguing that a given set of graphs cannot be simultaneously embedded. The first method is based on a combinatorial argument and the second method relies on a geometric argument.

The combinatorial technique is used in the proofs of Theorem 15 and Theorem 17. The main idea is as follows: given two or more graphs $G_1, \ldots, G_k$, let G^* be their union. If G^* contains $K_{3,3}$ or K_5 then there must be at least one crossing in any drawing. Now if every pair of edges in G^* is either adjacent or belongs to the same graph G_i for some $1 \leqslant i \leqslant k$, then there does not exist simultaneous embedding of the given graphs.

The geometric technique is used in the proofs of Theorems 14 and 16. The main idea is as follows: given two or more graphs $G_1, \ldots, G_k$, let G_1 be a maximally planar graph. Then there is only one embedding of G_1 up to choosing the outerface. To show that the given graphs cannot be simultaneously embedded it suffices to show that it would require a different embedding of G_1.

5.1. Planar graphs

Theorem 14. *There exist two planar graphs and a mapping between their vertices, such that they cannot be simultaneously embedded.*

Proof. The two planar graphs G_1, G_2 are as shown in Fig. 9. Note that G_1 and G_2 do not have any faces is common. Therefore, any crossings-free drawing of G_1 will induce crossings in G_2 and vice versa.. This implies that with the given mapping G_1 and G_2 cannot be simultaneously embedded. □

5.2. Outerplanar graphs

Theorem 15. *There exist two outerplanar graphs and a mapping between their vertices, such that they cannot be simultaneously embedded.*

Proof. The two outerplanar graphs O_1, O_2 are as shown in Fig. 10. The union graph G^* 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 (that is, none of the edges in that crossings pair can be present in both O_1 and O_2). But this leaves only a few choices to consider: the edges belonging to only O_1 are $(1, 2)$ and $(3, 6)$, while the edges belonging to only O_2 are $(2, 3)$ and $(1, 6)$. However, these edges cannot make a valid crossing pair either, as each such pair consists of incident edges (which cannot cross). Thus there must be another pair either in O_1 or in O_2 which intersects. □

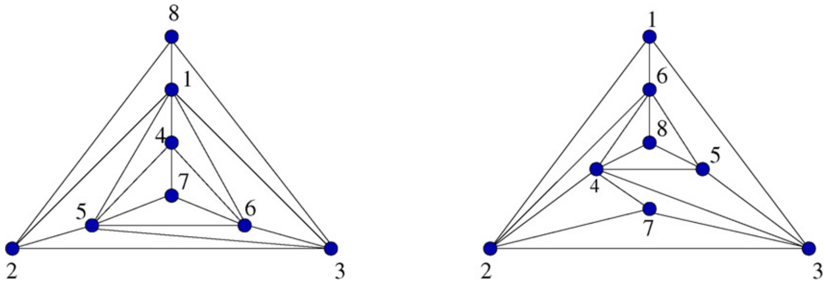


Fig. 9. Given the above mapping between the vertices; the planar graphs G_1 and G_2 cannot be embedded simultaneously.

Fig. 10. Given the above mapping between the vertices; the outerplanar graphs O_1 and O_2 cannot be embedded simultaneously.

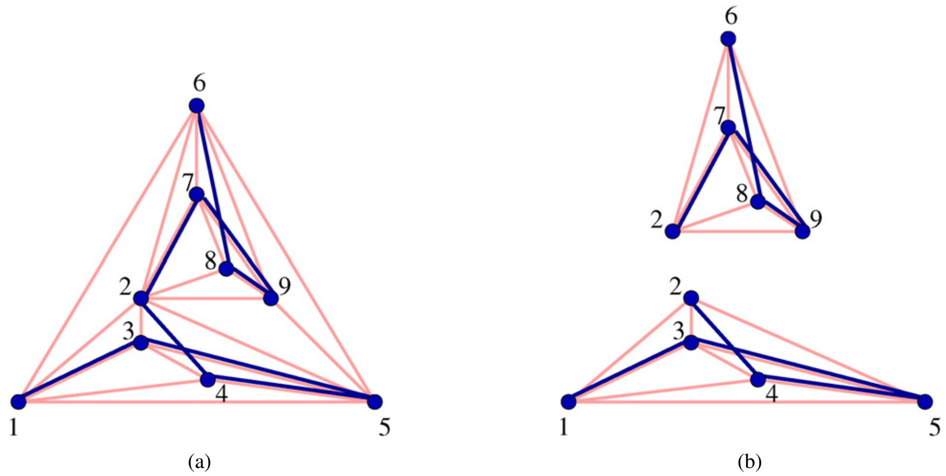


Fig. 11. (a) Graph G drawn with light edges and the path $P = (135427986)$ drawn with dark edges. (b) The two subgraphs of G.

5.3. A planar graph and a path

Theorem 16. *There exist a planar graph G, a path P, and a mapping between their vertices, such that the two graphs cannot be simultaneously embedded.*

Proof. Fig. 11(a) shows the graph G and the path P that cannot be simultaneously drawn without crossings. Let G' and G'' be the two subgraphs of G shown in Fig. 11(b). Note that with the given embedding of G, the path P contains a crossing in both G' and G''. Unless we change the embedding of G', it is impossible to draw it so that the path P does not contain any crossing. The same is true also for G'', However, changing the embedding of one of the subgraphs fixes the embedding of the other, as G is maximally planar. Thus, while we can undo one of P's crossings by choosing a different outerface, we will not be able to undo both of P's crossings at once. □

5.4. Three paths

Theorem 17. *There exist three paths $\mathcal{P} = \bigcup_{1 \leqslant i \leqslant 3} P_i$ and a mapping between their vertices, such that they cannot be simultaneously embedded.*

Proof. A path of n vertices is simply an ordered sequence of n numbers. The three paths we consider are: $P_1 = \{7, 1, 4, 2, 6, 9, 3, 5, 8\}$, $P_2 = \{8, 2, 4, 3, 5, 7, 1, 6, 9\}$ and $P_3 = \{7, 5, 8, 2, 6, 1, 4, 3, 9\}$. There are twelve edges in the union of these paths:

$$E = \big\{(1,4), (1,6), (1,7), (2,4), (2,6), (2,8), (3,4), (3,5), (3,9), (5,7), (5,8), (6,9)\big\}.$$

It is easy to see that the union 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 must be two nonadjacent edges of G that cross each other. However, it is easy to check that *every pair of nonadjacent edges* from E appears in *at least one* of the paths given above. Therefore, in any embedding of these vertices, at least one path will cross itself, which completes the proof. □

6. 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 results; 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 graphs are outerplanar.

Theorem 18. *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 19. *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 point sets in general position from the uniform grid, since it is known that any outerplanar graph can be embedded on any point set in general position (no three points collinear):

Theorem 20. [3,15] *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. [15] 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 19 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 [13]: 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 $p \geqslant n$ 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 always three points in the same row or column.

In order to prove Theorem 18, 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 [7]. The algorithm draws any 3-connected planar graph in an $O(n) \times O(n)$ grid under the *edge resolution rule*, producing 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 $p \geqslant n$, 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 $v_1 v_2$ 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 one such perturbation to obtain a general-position set: If the vertex v_ν was mapped by the algorithm of [7] on the point (i, j), then we map it to 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 $\{(\nu, \nu^2 \bmod p)\colon \nu = 1, \ldots, n\}$, and collinearity is a property that is preserved by the mod 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 18.

7. Open problems

(1) Given an outerplanar graph G, and a path P can we always simultaneously embed G and P with mapping?
(2) 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?
(3) Can any two planar graphs be simultaneously embedded without mapping?
(4) Which of the positive results in this paper are tight (that is, what are the lower bounds on the grid area)?

Acknowledgements

We would like to thank Ed Scheinerman, David Eppstein, and James Abello for stimulating discussions about different variations of the problem and Esther M. Arkin for her proof of Theorem 17 (independent of our work).

References

[1] M. Bern, J.R. Gilbert, Drawing the planar dual, Information Processing Letters 43 (1) (1992) 7–13.
[2] F. Bernhart, P.C. Kainen, The book thickness of a graph, Journal of Combinatorial Theory, Series B 27 (1979) 320–331.
[3] P. Bose, On embedding an outer-planar graph in a point set, Computational Geometry: Theory and Applications 23 (3) (2002) 303–312.
[4] G.R. Brightwell, E.R. Scheinerman, Representations of planar graphs, SIAM Journal on Discrete Mathematics 6 (2) (1993) 214–229.
[5] D. Castellanos, The ubiquitous π, part II, Mathematics Magazine 61 (1988) 148–163.
[6] E. Cenek, Layered and Stratified Graphs, PhD thesis, University of Waterloo, forthcoming.
[7] M. Chrobak, M.T. Goodrich, R. Tamassia, Convex drawings of graphs in two and three dimensions, in: Proc. 12th Annu. ACM Sympos. Comput. Geom., 1996, pp. 319–328.
[8] M. Chrobak, G. Kant, Convex grid drawings of 3-connected planar graphs, International Journal of Computational Geometry and Applications 7 (3) (1997) 211–223.
[9] H. de Fraysseix, J. Pach, R. Pollack, How to draw a planar graph on a grid, Combinatorica 10 (1) (1990) 41–51.
[10] G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis, Graph Drawing: Algorithms for the Visualization of Graphs, Prentice-Hall, Englewood Cliffs, NJ, 1999.
[11] M.B. Dillencourt, D. Eppstein, D.S. Hirschberg, Geometric thickness of complete graphs, Journal of Graph Algorithms and Applications 4 (3) (2000) 5–17.
[12] C.A. Duncan, D. Eppstein, S.G. Kobourov, The geometric thickness of low degree graphs, in: 20th Annual ACM–SIAM Symposium on Computational Geometry (SCG), 2004, pp. 340–346.
[13] P. Erdős, Appendix, in: K.F. Roth, On a problem of Heilbronn, Journal of the London Mathematical Society 26 (1951) 198–204.
[14] C. Erten, S.G. Kobourov, Simultaneous embedding of a planar graph and its dual on the grid, in: 13th Intl. Symp. on Algorithms and Computation (ISAAC), 2002, pp. 575–587.
[15] P. Gritzmann, B. Mohar, J. Pach, R. Pollack, Embedding a planar triangulation with vertices at specified points, American Mathematical Monthly 98 (1991) 165–166.
[16] M. Kaufmann, D. Wagner, Drawing Graphs: Methods and Models, Lecture Notes in Computer Science, vol. 2025, Springer, New York, 2001.
[17] P. Koebe, Kontaktprobleme der konformen Abbildung, Berichte über die Verhandlungen der Sächsischen Akademie der Wissenschaften zu Leipzig. Math.-Phys. Klasse 88 (1936) 141–164.

[18] K. Miura, S.-I. Nakano, T. Nishizeki, Grid drawings of 4-connected plane graphs, Discrete and Computational Geometry 26 (1) (2001) 73–87.
[19] P. Mutzel, T. Odenthal, M. Scharbrodt, The thickness of graphs: A survey, Graphs and Combinatorics 14 (1) (1998) 59–73.
[20] J. Pach, R. Wenger, Embedding planar graphs at fixed vertex locations, In: Graph Drawing, 1998, pp. 263–274.
[21] W. Schnyder, Planar graphs and poset dimension, Order 5 (4) (1989) 323–343.
[22] W.T. Tutte, How to draw a graph, Proceedings of the London Mathematical Society 13 (52) (1963) 743–768.
[23] M. Yannakakis, Embedding planar graphs in four pages, Journal of Computer and System Sciences 38 (1) (1989) 36–67.