Outerplanar Graphs and Trees on Tracks*

Justin Cappos and Stephen G. Kobourov
Department of Computer Science
University of Arizona
{justin,kobourov}@cs.arizona.edu

Abstract

Given a vertex-labeled tree on n vertices we show how to obtain a straight-line, crossings-free drawing of it on a set of n labeled concentric tracks, such that the vertex labels match the track labels. The tracks can be defined by conic sections (such as circles, ellipses, circular arcs) or other smooth convex curves. We show that this type of embedding can be used to simultaneously embed tree-path pairs, such that the tree is drawn without crossings, using one straight-line segment per edge, and the path is drawn without crossings, using one circular arc segment per edge. This result generalizes to outerplanar graphs. We also consider star-track embeddings of trees which we use to obtain simultaneous embeddings of tree-path pairs using piecewise linear edges. In particular, we show how to simultaneously embed tree-path pairs so that the tree is drawn without crossings, using one straight-line segment per edge and the path is drawn without crossings, using at most 2 bends per edge. These results also generalize to outerplanar graphs.

1 Introduction

Embedding trees and other classes of planar graphs on predetermined pointsets, or small integer grids is motivated by graph layout algorithms and applications in the visualization of relational information. Simultaneous embedding of planar graphs is motivated by its relationship with problems of graph thickness, geometric thickness, and contour tree simplification.

We define tracks to be nonintersecting copies of a shape formed by translating the shape in a direction or scaling the shape around a point. As may be seen in Fig. 1, line, sine wave, and staircase tracks may be formed by translating a shape to form parallel copies. Similarly circular and star tracks may be formed by scaling a shape around the origin.

Informally, a graph can be embedded on tracks if we can find a straight-line, crossings-free drawing of the graph on a set of fixed curves in the plane, so that each vertex lies on its corresponding curve; see Fig. 1. Formally, we embed a graph G on a set of tracks L, where G is an n-vertex graph with vertex labels $v_1, v_2, \ldots, v_n$ and L is a set of n tracks (smooth non-intersecting curves in the plane), labeled $l_1, l_2, \ldots l_n$, provided that $v_i \in l_i$, for $1 \leq i \leq n$ and the graph drawing is straight-line and crossings-free.

A *simultaneous geometric embedding* of two vertex-labeled planar graphs on n vertices is possible if there exists a labeled point set of size n such that each of the graphs can be realized on that point set (using the vertex-point mapping defined by the labels) with straight-line edge segments and without crossings. For example, any two paths can be simultaneously embedded, while there exist pairs of outerplanar graphs that do not have a simultaneous embedding. While it may be tempting to say that if the union of the two graphs contains a subdivision of K_5 or $K_{3,3}$ then the two graphs have no simultaneous geometric embedding, this is

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

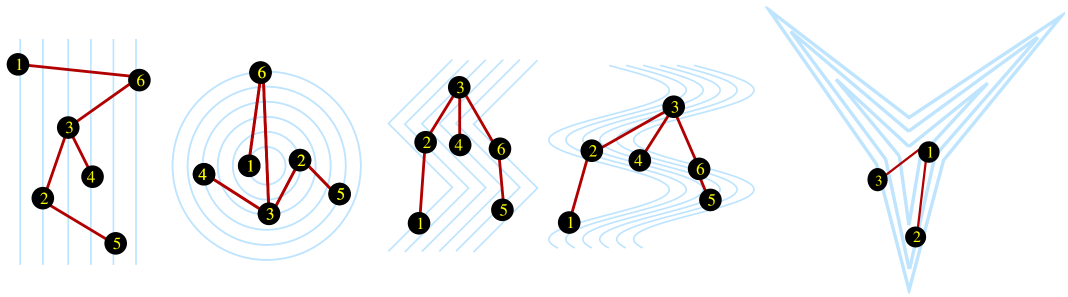


Figure 1: A tree embedded on line tracks, circle tracks, staircase tracks, $\sin x$ tracks, and star tracks.

not the case; see Fig. 2. In fact, while planarity testing for a single graph can be done in linear time [17], the complexity of determining whether a pair of graphs can be simultaneously embedded is not known. In addition to generalizing the notion of planarity, techniques for simultaneous embedding of cycles have been used to show that degree-4 graphs have geometric thickness at most two [12].

Contour trees were proposed by van Kreveld *et al* [24] for computing isolines on terrain maps in geographic information systems. Carr, Snoeyink and van de Panne [7] use contour trees for scientific and medical visualization. Contour tree simplification applies the ideas of topological persistence to trees and is another application for simultaneous drawing of trees, and in particular trees on tracks [6]. Simultaneous embedding techniques are also useful in the visualization of graphs that evolve through time, for example, in the context of visualization of the evolution of software [8].

In this paper we present results about track embeddings of trees and outerplanar graphs, as well as related results on simultaneous embedding of tree-path pairs. In particular, we show that trees cannot be embedded on parallel line tracks, but they can be embedded on tracks defined by conic sections (such as circles, ellipses, circular arcs) or other smooth convex curves. These results generalize to outerplanar graphs as well. We also show that tree-path pairs can be simultaneously embedded, such that the tree is straight-line and crossings-free and the path is crossings-free and each edge is drawn using one circular arc. We also show that tree-path pairs can be simultaneously embedded, such that the tree is straight-line and crossings-free and the path is crossings-free and each edge has at most 2 bends.

1.1 Related Work

The existence of straight-line, crossing-free drawings for a single planar graphs is well known [14, 23, 25]. Moreover, straight-line drawings for n-vertex planar graphs can be found in $O(n)$ time, using $O(n^2)$ area, with vertices placed at integer grid points, as shown by de Fraysseix, Pach and Pollack [9] and Schnyder [22]. If bends on the edges are allowed, Biedl [2] shows that outerplanar graphs can be embedded using $O(n \log n)$ area.

Brass *et al* [5] describe linear time algorithms for *simultaneous geometric embeddings* of pairs of paths, cycles, and caterpillars, using $O(n^2)$ area. If bends on the edges are allowed, Erten and Kobourov [13] show that tree-path pairs can be embedded simultaneously using one bend per tree edge. Moreover, tree-tree pairs can be embedded simultaneously using at most 3 bends per edge.

A related problem is the problem of *graph thickness*, defined as the minimum number of planar subgraphs into which the edges of the graph can be partitioned into; see survey by Mutzel, Odenthal and Scharbrodt [20]. 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. Dillencourt, Eppstein and Hirschberg [10] study *geometric thickness* of graphs, where the edges are required to be straight-line segments. Thus, if two graphs have a simultaneous geometric embedding, then their union has

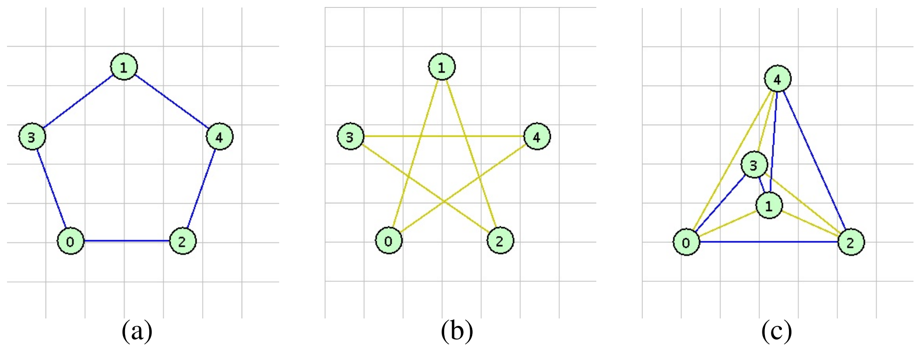


Figure 2: The union of the graphs in (a) and (b) is K_5, but (c) shows a simultaneous geometric embedding.

geometric thickness two. Similarly, the union of any two planar graphs has graph thickness two. Duncan *et al* [12] use simultaneous geometric embedding techniques to show that degree-four graphs have geometric thickness two. Finally, *book thickness* adds the further restriction that the vertices must be in convex position [1].

While the thickness and simultaneous embedding problems are related, results from one do not necessarily translate into the other. Bose, Hurtado, Rivera-Campo and Wood [4] show that the complete convex graph K_{2n} can be partitioned into n plane spanning trees and moreover, characterize all the different partitions. In particular, they show that K_{2n} can be partitioned into n non-crossing paths. However, given n paths it it not possible to always simultaneously embed them for $n \geq 3$, as shown by Brass *et al* [5].

Simultaneous drawing of multiple graphs is also related to the problem of *fixed pointset embedding* of planar graphs. Bose [3] and Gritzman *et al* [16] show that if the mapping between the vertices V and the points P is not fixed, then trees and outerplanar graphs can be drawn without crossings, using straight-line edges. In the same setting general planar graphs cannot be drawn without bends. If bends are allowed, Kaufmann and Wiese [19] show that two bends per edge suffice. However, if the mapping between V and P is predetermined, Pach and Wenger [21] show that $O(n)$ bends per edge are necessary to guarantee planarity, where n is the number of vertices in the graph.

In the context of 3D layout, Dujmovic, Por and Wood [11] study the (k,t)-*track layouts* of graphs, where the graph is vertex t-colored and edge k-colored. They examine the relationship between such layouts and geometric thickness. Felsner, Liotta and Wismath [15] characterize the trees that can be drawn on the $n \times 2$ grid and describe a universal pointset for outerplanar graphs in 3D.

1.2 Our Contribution

We begin with results on track embeddability.[1] There exists a tree with vertices labeled v_1 to v_n such that for any set of labeled parallel lines (i.e., tracks) L_1 to L_n there does not exist a straight-line crossings-free drawing of T, such that v_i is on track L_i. However, if the tracks are concentric circles instead of lines, then for every tree there exists such a drawing on some set of (concentric circular) tracks. We describe a linear time algorithm for obtaining such drawings and show that the algorithm easily generalizes to outerplanar graphs. Moreover, we show that other types of tracks also support such drawings, in particular tracks defined by conic sections, and other smooth convex tracks.

Our motivation for the problem of track embeddings comes from two open problems in simultaneous geometric embedding. Formally, in the problem of simultaneous geometric embedding we are given two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ and we would like to find straight-line crossings-free drawings

[1] Our use of "tracks" differs from earlier use of the word [11, 15], where the mapping between vertices and tracks is not given. In this paper, the mapping between the vertices and the tracks is predetermined by the vertex labels.

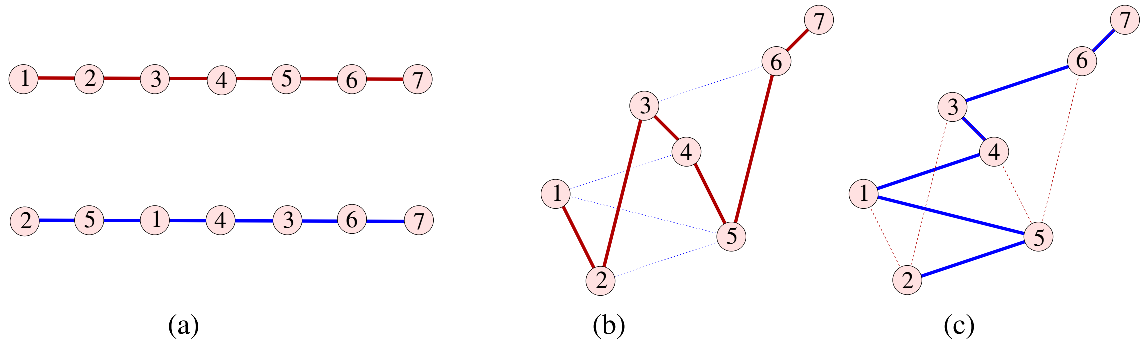


Figure 3: Path-path simultaneous embedding. The vertices of the first (second) path have monotonically increasing x-coordinates (y-coordinates), hence the first (second) path is crossings-free.

D_1 and D_2 such that for all vertices $v \in V$ the location of the corresponding vertices in D_1 and D_2 is the same (i.e., $D_i(v) = D_j(v)$). On one hand, several simple types of pairs, such as path-path, cycle-cycle, and caterpillar-caterpillar can be simultaneously embedded. On the other hand, several types of pairs, such as outerplanar-outerplanar and path-outerplanar, cannot always be simultaneously embedded. These results leave open the status of many other types of pairs. Recently, it was shown that tree-tree pairs cannot always be simultaneously embedded [18].

The circular track layout of trees and outerplanar graphs can be used to obtain simultaneous embeddings of path-outerplanar graph pairs, such that the outerplanar graph is straight-line and crossings-free while the path edges are crossings-free circular-arc segments. Moreover, particular kind of track, which we call a "star track" allows the simultaneous embedding of path-outerplanar graphs pairs so that the outerplanar graph is straight-line and crossings-free and the path edges are crossings-free and have at most 2 bends per edge.

2 Trees and Outerplanar Graphs on Tracks

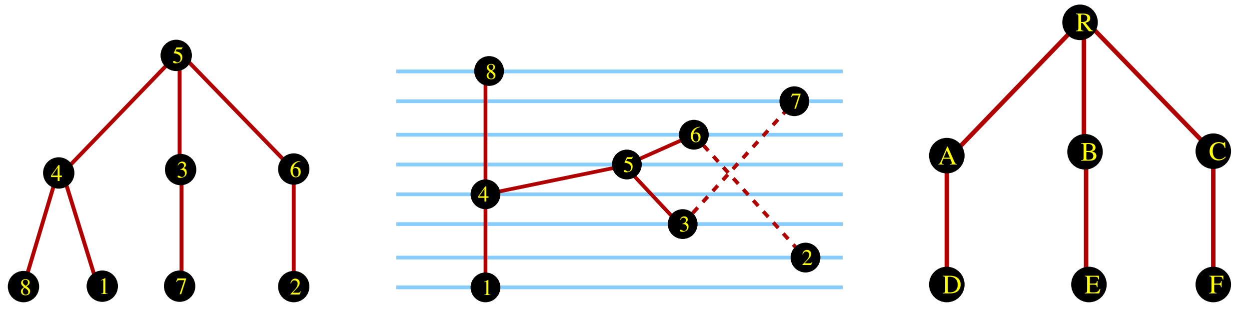
A common method for simultaneous embedding of path-graph pairs, is to place all of the vertices that form the path in order of their appearance on a series of parallel lines [5]. Without loss of generality we can assume that the n-vertex path is labeled $v_1, v_2, \ldots, v_n$. Thus, if we can draw the other graph without crossings while placing the vertices on a set of parallel lines labeled $L_1, L_2, \ldots, L_n$ in order, then the pair can be simultaneously embedded. This method is illustrated for the case when both graphs are paths; see Fig. 3;

In particular, if we can draw a labeled tree on a set of parallel labeled line tracks, then tree-path pairs can be simultaneously embedded. We show that not all trees allow such embeddings. However, we show that trees can be realized on labeled concentric circular tracks, provided that the ratio between the radii of the largest and the smallest circles is small. We generalize the result to outerplanar graphs and show that if the radii of the circular tracks are arbitrary, it is not possible to realize all trees and outerplanar graphs.

2.1 Trees on Parallel Line Tracks

Theorem 2.1 *Labeled n-vertex trees cannot always be embedded on n labeled parallel line tracks.*

Proof Sketch: To show that not all trees can be embedded on a labeled parallel line tracks it suffices to find a counterexample. The 8-vertex tree in Fig. 4 is the smallest such counterexample. Assume that we have already placed all vertices except for 2 and 7, on their corresponding tracks and there are no crossings yet. Then it is easy to show that any placement of 2 on track 2 and 7 on track 7 will result in either a crossing of the edge (3,7) with an edge of the tree or in a crossing of (2,6) with an edge of the tree.



(a) (b) (c)

Figure 4: The tree on the left cannot be drawn on a set of parallel lines with the vertices in increasing order. Regardless of how we place all of the vertices except v_2 and v_7, every placement will cause an intersection in the graph.

We can show that this is the smallest counterexample by observing that every tree which does not have a subgraph of Fig. 4(c) is a caterpillar and caterpillars may always be embedded on tracks[5]. Since Fig. 4(a) has only one more node, for a smaller counterexample to exist it must be Fig. 4(c) with some node labelling. Now observe that regardless of the track labelling of nodes, at least two of the children of R must be either above or below it. WLOG assume that the children are labelled A and B and that they are below R with A is above B. Let us place the R on its track at x=0 and draw nodes C and F at x=-1. Notice this causes no crossings.

Now we have three cases:

- Case 1: D is below A

  If this is the case then we can place A and D at x=0 and place B and E at x=1. This cannot cause a crossing because A and D are drawn directly below R and B and E are both to the right.

- Case 2: D is above A and E is below B

  If this is the case then we can place B and E at x=0 and place A and D at x=1. This cannot cause a crossing because B and E are placed directly below R and A and D are both to the right.

- Case 3: D is above A and E is above B

  Recall that A is above B. Let us place A, B, and D at x=1 and place E at x=2. Since A and B have the same x location the edges from them to R cannot cross. Since A is above B and D is above A similarly (A,D) will not cross another edge. The edge (B,E) cannot cross any others because it is the furthest edge to the right. Thus the drawing has no crossings. ■

It is not surprising that trees cannot be embedded on parallel line tracks, as the restriction that the order of the y-coordinate of every vertex is determined in advance is too strong. What is surprising, however, is that introducing just the slightest curvature to the tracks, is enough to allow us to embed all trees. Next we show how this can be done, starting with circular tracks.

2.2 Trees on Concentric Circular Tracks

The infeasability of embedding trees on a set of labeled parallel lines leads to the question of whether trees can be embedded on other types of tracks. In particular labeled concentric circular tracks are of interest, bearing in mind the simultaneous embedding applications. We show that any vertex-labeled tree can be realized on labeled concentric circular tracks by describing an algorithm for drawing trees on concentric circles and a formula for determining the appropriate radii.

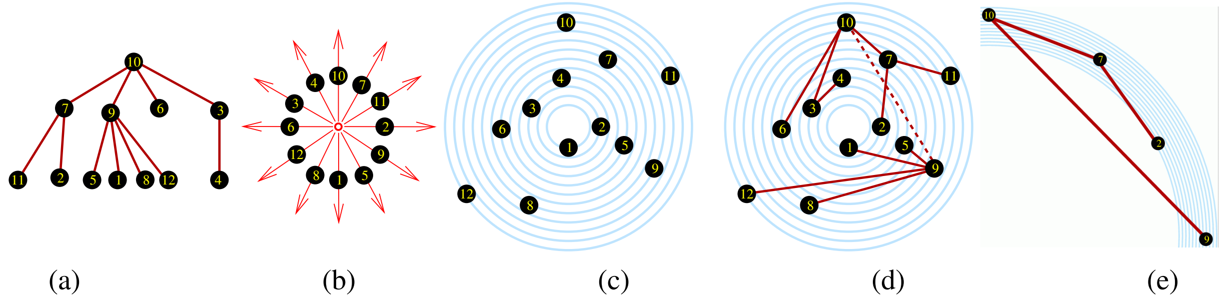


Figure 5: The vertices of the tree in (a) are arranged around a circle in order given by a pre-order traversal (b). The circle is scaled to n concentric circles and each vertex is moved along a ray from the center through its original position, until the appropriate track is reached (c).By moving the vertices to their tracks we may have introduced crossings (d). This problem can be resolved by fixing the ratio between the radii of the circles (e).

Theorem 2.2 *Any n-vertex labeled tree can be drawn without crossings on a set of n labeled concentric circular tracks, using straight-lines segments, in $O(n)$ time.*

Proof: We prove this claim by providing a linear time embedding algorithm. We begin by showing that any tree can be embedded on a single circle with the vertices evenly spaced around it. Next we perturb the n points so that each of them belongs to a unique concentric circle. This step, in effect, corresponds to the construction of a universal circular track set, which can be done by carefully selecting the radii of the n circles. Each of these steps is described below.

Drawing a Tree on a Circle: This is a special case of embedding a tree on a set of points in general position [3, 16]. Since we need the specifics of the placement for the next step, we provide some details. We begin by creating a circle, C, centered at the origin. We place n points $p_1, p_2, \ldots, p_n$ around the circle and evenly spaced. We map the n vertices of the tree to these n points using a mapping obtained from a pre-order traversal of the tree. Recall that in a pre-order traversal we visit the root of the current tree and then recursively explore all of its children. We perform a pre-order traversal of the tree, starting at an arbitrary vertex, and mapping the i-th vertex visited, to the i-th consecutive endpoint, p_i, of the circle; see Fig. 5(a-b).

To see that the resulting straight-line drawing of the tree is crossings-free consider two arbitrary tree edges (a,c) and (b,d). Two vertices labeled i, j by the pre-order traversal, are connected by an edge, only if $\forall k,\ i < k < j, (k,l)$ implies $i \leq l \leq j$. This means that we cannot have crossings by connecting vertices that lie on a circle in this manner because a crossing of (a,c) and (b,d) implies $a < b < c < d$, which contradicts the assumption that the labels came from a pre-order traversal.

Perturbing the Points: In this step we perturb the n points on circle C so that each of them is on its own circular track. We do this by creating n concentric circles starting with C and move each vertex along a ray from the origin through each point of C until it intersects the appropriate circle; see Fig. 5(b-c). However, the resulting tree drawing may not be crossings-free; see Fig. 5(d-e). Fortunately, is it not difficult to remedy this problem, by choosing the radii of the circular tracks more carefully.

Circle Radius Selection: We can draw the tree without any crossings on a set of concentric circles by choosing the radii appropriately. In particular, if a crossing of (a,c) and (b,d) implies $a < b < c < d$ then we can use the same algorithm as above. Notice that with n vertices, it is sufficient to show that any edge between vertices x and y crosses the smallest concentric circle; see Fig. 5(e). In order for each edge to have this property, it is sufficient to show the shortest edge has this property. Given two points on a circle that are $\frac{1}{n} * 2\pi$ radians away from each other, the distance between the midpoint of this line segment and the radius of the circle is $1 - \cos(\frac{\pi}{n})$ in units where 1 is the radius of the circle. This means that if we make n concentric circles have radii $\frac{1-\cos(\frac{\pi}{n})}{n}$ apart, we have the desired property. Note this is not a tight bound as it can be shown that we can relax the circle distance by a factor of 3 by observing that only the edges between vertices at least 2 apart need to cross the smallest concentric circle and that both of the vertices cannot live on the outermost circle. ■

2.3 Outerplanar Graphs on Concentric Circular Tracks

The idea of embedding a tree on a single circle and then refining the circle to n concentric circular tracks can be extended to outerplanar graphs as well, as we show below.

Corollary 2.3 *A n-vertex labeled outerplanar graph can be drawn on a set of n labeled concentric circular tracks without crossings in $O(n)$ time.*

Proof: We assume that the outerplanar graph is maximally outerplanar. If it is not, it can be appropriately augmented, and when the algorithm completes, the extra edges can be removed. A combinatorial embedding of the graph can be found in linear time [17]. Next we place the vertices of the graph onto n points evenly spaced around a given circle, so that the edges can be drawn as straight-line segments and there are no crossings. Once again, the correctness of this step follows from [3, 16]. Similar to the approach in Theorem 2.2, we then perturb the n points by scaling the circle into n distinct circular tracks, $\frac{1-\cos(\frac{\pi}{n})}{n}$ apart. The vertex labels of G determine the appropriate tracks for each vertex. The separation of the tracks was chosen so that every edge on the outerface must intersect the innermost track. Once again, it is straight-forward to verify that if all edges intersect the innermost track, then resulting drawing is still-crossings free. ■

2.4 Trees and Outerplanar Graphs on Refinable Shapes

It is easy to see that the results above extend to circular arc tracks, as well as to conic section tracks, such as parabolas, ellipses, hyperbolas, and more generally, to any shape that may be refined into tracks. We can draw labeled outerplanar graphs on any set of shapes given certain restrictions (all the tracks are obtained by scaling or translating one original shape, tracks do not intersect, and the separation between the tracks is bounded). We summarize the results in the following theorem.

Theorem 2.4 *Given a straight-line, crossings-free drawing of an outerplanar graph G with n vertices on a shape S in the plane where S can be refined into n tracks, G can be drawn on the refinement of S into n tracks.*

Proof Sketch: G is drawn on a some shape S in the plane if its vertices lie on S. Consider a plane drawing of G on S. Let ε be the minimum distance between a pair of non-adjacent edges of G. Consider n copies of S (the n tracks), scaled from the original one, so that the distance between two tracks is at most $\varepsilon/2n$. By perturbing the vertices from their original positions on S to the track determined by their label, in a direction perpendicular to a tangent at S, a vertex moves no more than $\varepsilon/2$ away from its original position. Since the minimum distance between any pair of edges of G was ε, after the vertex positions have been perturbed, no pair of non-adjacent edges intersects. ■

2.5 Trees on Predetermined Concentric Circular Tracks

Circular tracks, tracks determined by conic sections, or tracks determined by smooth convex curves allow the realization of trees and outerplanar graphs only if the separation between neighboring tracks meets certain criteria. If the separation is predetermined, it is not necessarily true that any tree or outerplanar graph can be realized without crossings and using straight-line segments. This idea is captured in the following lemma for the case of circular tracks.

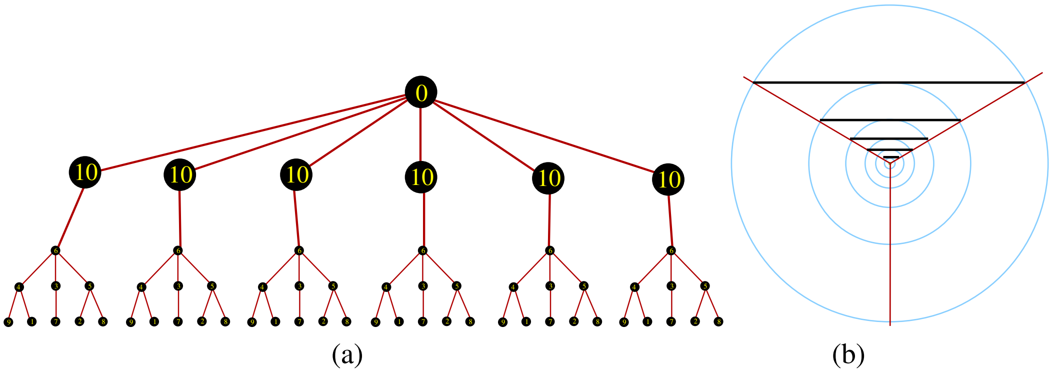


Figure 6: (a) This tree cannot be drawn on circular tracks with predetermined radii if multiple vertices must be placed on the same track; (b) Any vertices placed on the concentric circular arc in the top 1/3 region must have strictly increasing y-coordinates.

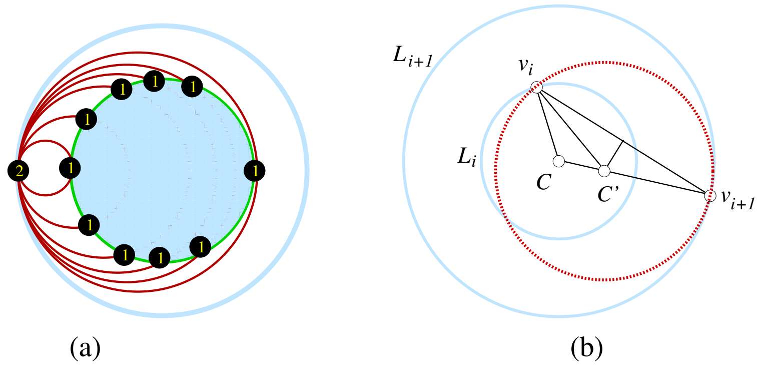


Figure 7: (a) Routing one circular arc per edge that fits inside two consecutive concentric circles; (b) The concentric circles with the center at C. C' is the center of a circle that gives a curve (dashed circle) that must connect v_i and v_{i+1} while staying in the annulus of L_i and L_{i+1}.

Lemma 2.5 *If the radii for the circular tracks are predetermined, trees cannot always be realized without crossings.*

Proof Sketch: We will constrain the problem slightly by placing multiple vertices on the same concentric track. Note that we can do this for an arbitrary tree if we are allowed to choose the radii. The tree in Fig. 6(a) has its root vertex labeled 0, and hence must be placed on the innermost track. The root's 6 children are labeled so that they must be placed on the outermost track. Any drawing of this tree must divide the concentric circular regions into at least three sectors (because we can place two of the edges adjacent to the root next to each other and place pairs of their subtrees on opposite sides). The subtrees hanging off the root's children are copies of the tree that cannot be embedded on parallel tracks; see Fig. 4. Since we have three sectors there must be at least one sector of size 1/3 or less. Without loss of generality let this sector be the top 1/3 of the circular tracks; see Fig. 6(b). Then the radii of the tracks can be chosen progressively larger as we move away from the innermost track, so that any vertices placed on the concentric circular arc in the top 1/3 sector must have strictly increasing y-coordinates. Since we cannot realize the subtree on parallel line tracks, we cannot realize the entire tree. ■

3 Simultaneous Embedding with Curves and Bends

In this section we use the results from the previous section to obtain simultaneous embedding of an outerplanar graph and a path. The fact that we can embed an outerplanar graph on concentric circular tracks, can be used to show that we can simultaneously embed an outerplanar graph and a path, such that the outerplanar graph is straight-line and crossings free, while the path uses one circular arc per edge and is crossings free.

Typically, piecewise-linear edges are used to visualize graph edges, and we can extend the idea to simultaneous embedding with bends. It is straight-forward to simultaneously embed a pair of n-vertex planar graphs such that one is straight-line and crossings-free and the other has $O(n)$ bends per edge and is crossings-free, using the result by Pach and Wenger [21]. Using our track embeddings of outerplanar graphs and trees, we can improve on this result for the case of path-outerplanar graph pairs.

We show that we can simultaneously embed an outerplanar graph and a path, such that the outerplanar graph is straight-line and crossings-free, while the path is crossings-free and has exactly one circular arc per edge.

For the case where we insight on piecewise-linear edges, we can simultaneously embed an outerplanar graph and a path, such that the outerplanar graph is straight-line and crossings-free, while the path is crossings-free and has at most 2 bends per edge. We do this by embedding the tree on "star tracks" without crossings using straight-line edges, while routing the path edges between the star tracks with at most 2 bends.

3.1 Curvilinear Simultaneous Embedding of a Path and Outerplanar Graph

Recall that given n consecutive concentric circular tracks, $L_1, L_2, \ldots, L_n$ we can realize any outerplanar graph on n vertices such that $v_i \in L_i$, $1 \le i \le n$. Also, we assume (without loss of generality) that the path p contains the vertices in order, i.e., $p = v_1, v_2, \ldots, v_n$. We show how to route the edges of the path using exactly one circular arc segment per edge of the path so that no two such circular arcs intersect (other than at incident vertices).

Lemma 3.1 *A crossings-free drawing of a path $p = v_1, v_2, \ldots, v_n$ can be realized on n consecutive concentric circular tracks $L_1, L_2, \ldots, L_n$, such that $v_i \in L_i$, $1 \le i \le n$, using one circular arc per edge.*

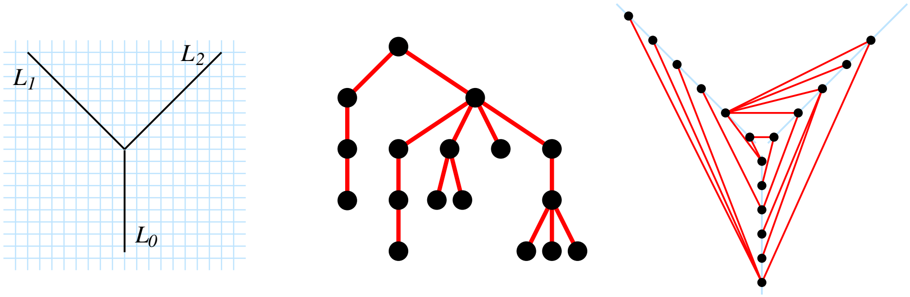
Proof: It suffices to show that one circular arc can be used to connect two consecutive vertices on the path, v_i and v_{i+1}, such that the arc is outside circular track L_i and inside circular track L_{i+1}, regardless of the exact placement of v_i on L_i and v_{i+1} on L_{i+1}; see Fig. 7(a).

Let C' be the circle that forms the needed circular arc connecting v_i and v_{i+1}. We begin by placing C' on the line between C, the center of the concentric circles, and the vertex v_{i+1} on L_{i+1}; see Fig. 7(b). We chose the radius of the circle, so that it intersects L_{i+1} exactly once (and since C' is inside, the circle is completely inside track L_{i+1}). This curve will intersect L_i at most twice and we can place the center of the circle so that one of these intersections is at the vertex v_i on L_i. We can find C' by first drawing a perpendicular line bisecting the line segment between v_i and v_{i+1}. We can then intersect this line with the line from C to v_{i+1} to obtain the location of C'. Since the distance between v_i and C' is equal to the distance between C' and v_{i+1}, the circular arc connecting v_i to v_{i+1} is inside L_{i+1} and outside L_i. ■

Theorem 2.3 together with Lemma 3.1 give us the following theorem.

Theorem 3.2 *An outerplanar graph and a path can be simultaneously embedded, such that the outerplanar graph is straight-line and crossings-free, while the path uses one circular arc per edge and is crossings free.*

Using our algorithm to draw a path with circular arcs and our tree with straight lines, it immediately follows that if we were to view the circular tracks as n-gons instead, we can simultaneously embed the tree with no bends and the path (by following the n-gon) with $O(n)$ bends.



(a) (b) (c)

Figure 8: A star shape (a), a tree (b), and an embedding of the tree on the star shape (c).

3.2 Star Tracks

In this section we show how to reduce the number of bends on the path from $O(n)$ per edge to 2 per edge. In order to do this we will use a different kind of track called a star track. We begin by showing how to draw arbitrary outerplanar graphs on a star shape.

Lemma 3.3 *An unlabeled outerplanar graph G on n vertices can be drawn without crossings using straight-line segments with its vertices placed on integer grid coordinates defined by three line segments* L_0, L_1, L_2*, with endpoints* $(0,n)$ *and* (n,n)*,* $(0,2n)$ *and* (n,n)*,* $(2n,2n)$ *and* (n,n)*, respectively.*

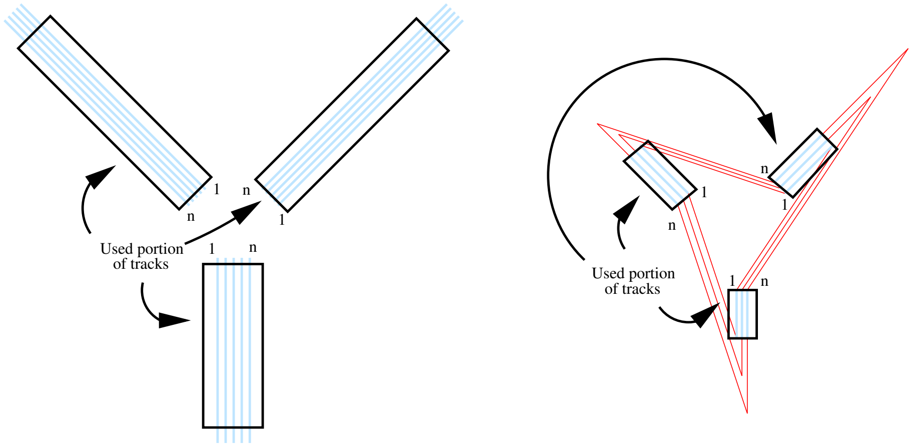
Proof: A slightly weaker result (using 3 longer line-segments) can be obtained as a corollary from Theorem 5 of Felsner *et al* [15], where outerplanar graphs are drawn on a prism in 3D space. The argument below follows along similar lines.

Let us assume we have an unlabeled outerplanar graph G and an outerplanar embeding of this graph. Consider the 3 line segments, L_0, L_1, L_2; see Fig. 8(a). These line segments determine $3n+1$ points on the integer grid: $(n,0),(n,1),\ldots(n,n)$ on L_0; $(0,2n),(1,2n-1),\ldots,(n,n)$ on L_1; and $(2n,2n),(2n-1,2n-1),\ldots(n,n)$ on L_2. Any unlabeled outerplanar graph G on n vertices can be drawn without crossings, using straight-line segments, by placing the vertices on a subset of the $3n+1$ points defined by the 3 line segments as follows. We can take an arbitrary vertex r from the graph and call it the root. Using a counterclockwise breadth first search of G from r we can label the vertices of G with two labels: the order they are visited (ignoring nodes that have already been visited) and their distance from the root. For each vertex, if its distance from the root is k we place it on segment L_i, where $k \bmod 3 \equiv i$. The order of the vertices along the segments L_i is determined by the order they were visited. We begin by placing r at position $(n,n-1)$ on segment L_0. All of its children are placed on L_1, starting with $(n-1,n+1)$, and taking grid points in order. All of r's grandchildren are placed on L_2 starting at $(n+1,n+1)$, and so on in clockwise manner. It is easy to see that this method produces no crossings but we leave the details out of the abstract; see Fig. 8(b-c).

This is an $O(n)$ algorithm that requires a $2n \times 2n$ integer grid.

■

Now, we extend the line segments L_0, L_1, L_2 into a star track as shown in Fig 9(a). First of all, we refine the star shape to have starting and stopping points. We will use the area between the starting and stopping points for drawing our outerplanar graph. In order to build star tracks from our star shape, we connect the end of the used part of track 1 to the beginning of the used part of track 1 on the next clockwise star area. To define track 2, we create a connecting line parallel to that for track 1 and extend track 2 past the used area until they intersect. We draw the other tracks in a similar fashion; see Fig 9(b). Observe that we can draw arbitrarily many tracks in this fashion and none of the tracks intersect.



(a) (b)

Figure 9: A refinement of the star shape (a) and the corresponding star tracks (b).

Since the separation between neighboring tracks is bounded, we can extend the outerplanar graph embedding algorithm for one star shape (determined by L_0, L_1, L_2) to an embedding algorithm on the n star tracks. This leads to the following theorem, the full proof of which is left out of this extended abstract.

Theorem 3.4 *Any n-vertex labeled outerplanar graph can be realized on a set of n concentric star tracks without crossings.*

Proof: We know from Theorem 2.4 that since we can draw an outerplanar graph on this shape, we can draw a outerplanar graph on tracks of this shape (because it may be refined by scaling about the center of the star shape). We can use the algorithm described in Lemma 3.3 to place our nodes on tracks. Note that we only use the indicated regions of the tracks in Fig. 9(b) when we place the nodes. ■

We can use this outerplanar graph embedding on star tracks to yield our final theorem.

Theorem 3.5 *An outerplanar graph and a path can be simultaneously embedded such that the outerplanar graph is straight-line and crossings-free, while the path uses 2 bends per edge and is crossings-free.*

Proof: From the point of view of simultaneous embeddings with bends, Theorem 3.4 provides us with a method for embedding an outerplanar graph on star tracks. Recall that without loss of generality the path is labeled $v_1, v_2, \ldots, v_n$. Given the above star track embedding of the outerplanar graph, we can route edges of the path (v_i, v_{i+1}), $1 \leq i < n$, along the star tracks so that they do not intersect. We have two cases for connecting adjacent path nodes, either they are in the same used region of the star tracks or different regions.

If they are in the same region we can simply connect the nodes and this path line will be fully contained between the two tracks.

If the nodes are in different regions we will connect the lower numbered node by following the higher numbered track either clockwise or counterclockwise (whichever is shorter) until we reach the higher numbered node. To get on the higher numbered track we connect to the higher numbered track at the location of the first bend in the direction of the higher node. Since this is a point on the parallel track we clearly don't move outside of the region between the two tracks (including the higher track). Now we just follow the higher numbered track around to reach the node.

Since each path edge doesn't cross the boundaries between tracks and that tracks themselves do not cross we have no crossings of path edges in the graph. Since it takes 2 bends to go from one region to the next using the tracks, we can draw the path with 2 bends and without crossings. By Theorem 3.4 we can simultaneously draw the outerplanar graph crossings-free with straight lines. ■

4 Conclusions and Open Problems

We presented several results on embedding labeled trees and outerplanar graphs on labeled tracks. We showed how these results can be used to obtain simultaneous embeddings of path-tree and path-outerplanar graph pairs using circular arc edges or a small number of bends. Several simultaneous embedding problems remain open, with two of the most relevant to this work being:

1. Do all tree-path pairs have simultaneous geometric embedding?
2. What is the complexity of determining whether two planar graphs admit a simultaneous geometric embedding?

References

[1] F. Bernhart and P. C. Kainen. The book thickness of a graph. *J. Combin. Theory*, Ser. B 27:320–331, 1979.

[2] T. C. Biedl. Drawing outer-planar graphs in O(n log n) area. In *10th International Symposium on Graph Drawing (GD'02)*, pages 54–65, 2002.

[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] P. Bose, F. Hurtado, E. Rivera-Campo, and D. Wood. Partitions of complete geometric graphs into plane trees. In *12th International Symposium on Graph Drawing (GD '04)*. To appear in 2004.

[5] 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 (WADS)*, pages 243–255, 2003.

[6] H. Carr, J. Snoeyink, and M. van de Panne. Contour tree simplification with local geometric measures. In *14th Annual Fall Workshop on Computational Geometry*, pages 51–52, 2004.

[7] H. Carr, J. Snoeyink, and M. van de Panne. Simplifying flexible isosurfaces using local geometric measures. In *IEEE Visualization*, pages 497–504, 2004.

[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 *ACM Symposium on Software Visualization (SoftVis)*, pages 77–86, 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] 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.

[11] V. Dujmovic, A. Por, and D. R. Wood. Track layouts of graphs. *Discrete Mathematics and Theoretical Computer Science*. To appear.

[12] 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.

[13] C. Erten and S. G. Kobourov. Simultaneous embedding of planar graphs with few bends. In *12th Symposium on Graph Drawing (GD)*. To appear in 2004.

[14] I. Fáry. On straight lines representation of planar graphs. *Acta Scientiarum Mathematicarum*, 11:229–233, 1948.

[15] S. Felsner, G. Liotta, and S. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. *Journal of Graph Algorithms and Applications*, 7(4):363–398, 2003.

[16] 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.

[17] J. Hopcroft and R. E. Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, 1974.

[18] M. Kaufmann and I. Vrto. Two trees without simultaneous embedding. Unpublished manuscript, Dagstuhl 2005.

[19] 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.

[20] P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of graphs: a survey. *Graphs and Combinatorics*, 14(1):59–73, 1998.

[21] J. Pach and R. Wenger. Embedding planar graphs at fixed vertex locations. *Graphs and Combinatorics*, 17:717–728, 2001.

[22] W. Schnyder. Planar graphs and poset dimension. *Order*, 5(4):323–343, 1989.

[23] S. K. Stein. Convex maps. *Proceedings of the American Mathematical Society*, 2(3):464–466, 1951.

[24] M. J. van Kreveld, R. van Oostrum, C. L. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *13th Annual ACM-SIAM Symposium on Computational Geometry (SGG)*, pages 212–220, 1997.

[25] K. Wagner. Bemerkungen zum vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–32, 1936.