Point-Set Embedding of Trees with Edge Constraints[*]

(Extended Abstract)

Emilio Di Giacomo[1], Walter Didimo[1], Giuseppe Liotta[1], Henk Meijer[2], and Stephen Wismath[3]

[1] Dip. di Ingegneria Elettronica e dell'Informazione, Università degli Studi di Perugia
`{digiacomo, didimo, liotta}@diei.unipg.it`
[2] Roosevelt Academy, The Netherlands
`h.meijer@roac.nl`
[3] Department of Mathematics and Computer Science, University of Lethbridge
`wismath@cs.uleth.ca`

Abstract. Given a graph G with n vertices and a set S of n points in the plane, a *point-set embedding* of G on S is a planar drawing such that each vertex of G is mapped to a distinct point of S. A *geometric point-set embedding* is a point-set embedding with no edge bends. This paper studies the following problem: The input is a set S of n points, a planar graph G with n vertices, and a geometric point-set embedding of a subgraph $G' \subset G$ on a subset of S. The desired output is a point-set embedding of G on S that includes the given partial drawing of G'. We concentrate on trees and show how to compute the output in $O(n^2 \log n)$ time and with at most $1 + 2\lceil k/2 \rceil$ bends per edge, where k is the number of vertices of the given subdrawing. We also prove that there are instances of the problem which require at least $k - 3$ bends for some of the edges.

1   Introduction

Let G be a planar graph with n vertices and let S be a set of n points in the plane. A *point-set embedding of G on S* is a crossing-free drawing of G such that each vertex is represented as a distinct point of S and the edges are polygonal chains. The problem of computing a point-set embedding of a graph, also known as the *point-set embeddability problem*, has been extensively studied both when the mapping of the vertices to the points is chosen by the drawing algorithm and when it is partially or completely given as part of the input. A limited list of papers about different versions of the point-set embeddability problem includes, for example, [1,2,3,5,6,7,8,12,15].

This paper studies a natural extension of the point-set embeddability problem. It is assumed to have a mapping of some edges of G to segments defined on S

---

and the goal is to compute a point-set embedding of G that includes the given segments. More precisely, we focus on trees and study the following question: The input is a set S of n points, a tree T with n vertices, and a point-set embedding of a subtree $T' \subset T$ on a subset of S such that all edges of this partial drawing are straight-line segments. The desired output is a *constrained point-set embedding* of T on S, i.e. a point-set embedding of T on S that includes the given partial drawing of T'.

From the application point of view, the point-set embeddability problem is relevant in those contexts where the display of the vertices is constrained to use a set of prescribed locations. Our variant adds the constraint that a portion of the graph is already drawn; this can be important for example to preserve the user's mental map when a certain subgraph of an evolving network does not change over time. Again, representing certain edges as straight-line segments and placing their end-vertices at specific locations can be used to emphasize the importance of these objects with respect to other objects of the graph.

We recall that a recent paper on extending a partial straight-line drawing is [16]. Given a planar graph G and a planar straight-line drawing Γ of a subgraph of G, the author of [16] shows that it is NP-hard deciding whether G admits a planar straight-line drawing including Γ. The main difference between the problem studied in [16] and the one investigated in this paper is that, when extending the partial straight-line drawing, we have fixed locations for the vertices and we allow bends along the edges.

The main contribution of this paper is to provide lower and upper bounds to the maximum number of bends per edge in a constrained point-set embedding of a tree. An outline of the results is as follows.

- We prove that a constrained point-set embedding of a tree on a set of points can require one edge bend even if the partial drawing consists of just a single edge. We recall that every tree with n vertices admits a straight-line point-set embedding onto any set of n points in general position [3,8].
- We extend the above result by showing a lower bound that depends on the number of vertices of the given subdrawing of the tree. Namely we prove there exist trees with $n > 7$ vertices and partial drawings with $k < n$ vertices such that any constrained point-set embedding has at least $n - k$ edges, each having at least $k - 3$ bends.
- We describe a drawing algorithm that computes a constrained point-set embedding of a tree in $O(n^2 \log n)$ time and with at most $1 + 2\lceil k/2 \rceil$ bends per edge, where n is the number of vertices of the tree and k is the number of vertices of the given subdrawing. We remark that the difference between such an upper bound and the lower bound mentioned above is at most 5.

The proof of the upper bound is based on the partial solution of a computational geometry problem that in our opinion is of independent interest. Kaneko and Kano [9,10] studied the problem of computing a point-set embedding with straight-line edges of a forest F of rooted trees such that the location of the root of each tree of F is part of the input. Kaneko and Kano show that the drawing

can always be computed for special types of forests (rooted star forests or forests of trees where the sizes of any two trees differ by at most one) but the problem is still open in the general case.

One of the basic ingredients of our upper bound technique sheds more light on the problem described above. Namely, let $T_0, \ldots, T_{h-1}$ be a forest of trees with n vertices in total. Let $S = \{p_0, \ldots, p_{n-1}\}$ be a set of n points in general position such that $p_0, \ldots, p_{h-1}$ are points of the convex hull of S. We describe an $O(n^2 \log n)$ time procedure to compute a straight-line point-set embedding of the forest such that the root of T_i is on p_i $(i = 1, \ldots, h-1)$.

The remainder of this paper is organized as follows. Preliminary definitions are in Section 2. The study of the constrained point-set embeddability problem for trees is in Section 3: Lower bounds are provided in Subsection 3.1 and an upper bound is given in Subsection 3.2. Conclusions and open problems are in Section 4.

2   Preliminaries

We assume familiarity with basic notions of graph drawing and of computational geometry (see, e.g., [4,11,13,17]).

Let $G = (V, E)$ be a planar graph with n vertices and let S be a set of n points in the plane. A *point-set embedding of G on S*, denoted as $\Gamma(G, S)$, is a planar drawing of G such that each vertex is mapped to a distinct point of S. $\Gamma(G, S)$ is called a *geometric point-set embedding* if each edge is drawn as a straight-line segment.

Let $D(S)$ be a straight-line drawing whose vertices are points of a subset of S. We say that $D(S)$ is a *partial drawing* of G on S if it represents a graph isomorphic to a subgraph of G. A *constrained point-set embedding* $\Gamma(G, D(S))$ is a point-set embedding of G on S such that $D(S)$ is a subdrawing of $\Gamma(G, D(S))$.

For example, Figure 1 shows a graph G, a partial drawing $D(S)$ of G on a set S of points, and a constrained point-set embedding $\Gamma(G, D(S))$.

In the remainder of the paper, we say that the points of S are in *general position* if no three points of S lie on the same line. A corner v of a polygon
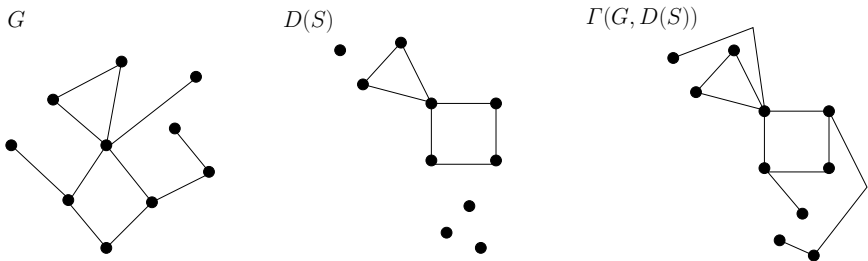

Fig. 1. A planar graph G. A set S of points and a partial drawing $D(S)$ of G. A constrained point-set embedding $\Gamma(G, D(S))$ with at most one bend per edge.

in the plane is said to be a *reflex corner* if the angle at v inside the polygon is greater than 180 degrees.

3   Constrained Point-Set Embeddings of Trees

In this section we investigate the constrained point-set embeddability problem for a tree T on a set S of points. We present lower and upper bounds to the maximum number of bends per edge in a constrained point-set embedding $\Gamma(T, D(S))$. These bounds depend on the number of vertices of the partial drawing $D(S)$.

3.1   Lower Bounds

We first show that there exist a tree and a set of points such that, even for a partial drawing consisting of a single edge, a constrained point-set embedding requires at least one edge bend. A more general lower bound is then provided.

Lemma 1. *There exist a tree T of n vertices, a set S of n points, and a partial drawing $D(S)$ of T on S consisting of a single edge, such that every constrained point-set embedding $\Gamma(T, D(S))$ has an edge with at least 1 bend.*

Sketch of Proof: Consider the tree T and the drawing $D(S)$ in Figure 2. Let s denote the single edge of $D(S)$. Let v_0, v_1, v_2, and v_3 denote four vertices of T as illustrated, i.e., v_0, v_1, v_2, and v_3 form a path from the root of T to a leaf. Assume that we have a drawing $\Gamma(T, D(S))$ without bends. Notice that no point above s can be connected to a point below s without a bend. Because of symmetry we only need consider three cases: either (v_0, v_1), (v_1, v_2), or (v_2, v_3) is mapped to s.

(v_0, v_1) **is mapped to** s: Removal of v_0 and v_1 splits T into three sub-trees, one with 3 vertices and two with 4 vertices each. Each sub-tree either has to be drawn above or below s. This cannot be done since there are 6 points above and 5 points below s and no combination of $\{3, 4, 4\}$ adds up to 5.

(v_1, v_2) **is mapped to** s: Removal of v_1 and v_2 splits T into three sub-trees, two with 1 and one with 9 vertices. No combination of $\{1, 1, 9\}$ adds up to 5.

(v_2, v_3) **is mapped to** s: Removal of v_2 and v_3 splits T into two sub-trees, one with 1 and one with 10 vertices. No combination of $\{1, 10\}$ adds up to 5. □

Lemma 2. *There exist a tree T with $n > 7$ vertices, a set S of n points, and a partial drawing $D(S)$ of a tree with $7 \leq k < n$ vertices, such that every constrained point-set embedding $\Gamma(T, D(S))$ has $n - k$ edges each having at least $k - 3$ bends.*

Sketch of Proof: Consider a tree T consisting of a path $v_0, v_1, \ldots, v_{k-3}$ of $k - 2$ vertices, a vertex u adjacent to v_1, a vertex w adjacent to v_2 and $n - k$ vertices adjacent to v_{k-3} (see Figure 3 for an illustration with $k = 9$ and $n = 14$). Let T' be the subgraph of T containing all vertices of T except the $n - k$ vertices adjacent to v_{k-3}. There is exactly one subgraph in T isomorphic to T', and the
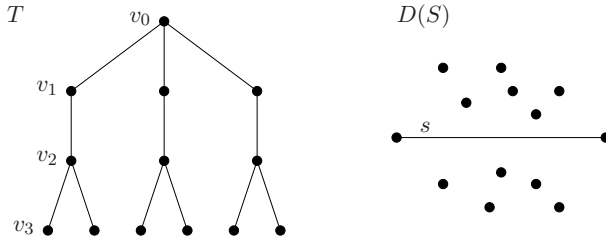
Fig. 2. A tree T, a set S of points, and a partial drawing $D(S)$ with a single edge s

remaining $n - k$ vertices of T are adjacent to v_{k-3} which is the only leaf node of T' with a degree 2 neighbor. Notice that for such a leaf to exist we require $k \geq 7$. Let $D(S)$ be a partial drawing of T on S constructed as shown in Figure 3; the edges of $D(S)$ (the solid edges in the figure) form a tree isomorphic to T'. Since, as already observed, there is only one subgraph in T isomorphic to T', the edges that we must add to $D(S)$ to get a drawing $\Gamma(T, D(S))$ are those adjacent to v_{k-3} (see, e.g., edge e in Figure 3). As also shown in the figure, it is not hard to see that each of these edges requires at least $k - 3$ bends. $\qquad\square$
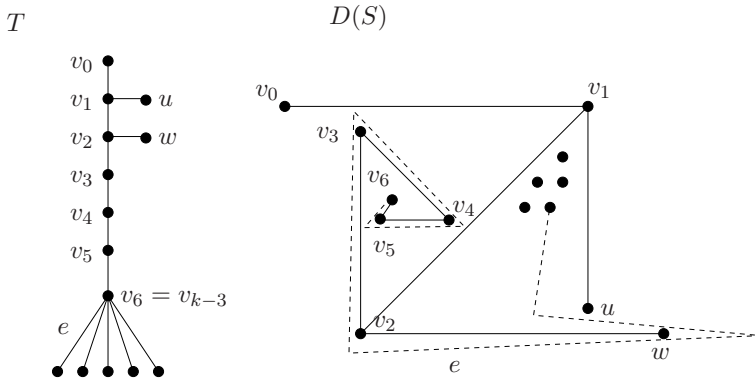

Fig. 3. Illustration of the proof of Lemma 2. A tree T with $n = 14$ vertices, a set S of n points, and a partial drawing $D(S)$ (with solid edges) with $k = 9$ vertices. Every $\Gamma(T, D(S))$ requires $n - k = 5$ edges each having at least $k - 3 = 6$ bends (see for example the dashed edge).

3.2   Upper Bound

Let T be a tree with n vertices and let S be a set of n points. In this section we show that if $D(S)$ is a partial drawing of T on S such that $D(S)$ represents a tree with k vertices, then we can always construct a constrained point-set embedding $\Gamma(T, D(S))$ with at most $1 + 2\lceil k/2 \rceil$ bends per edge. This means that each edge of T that we add to complete $D(S)$ is drawable with a number of bends that is

linear in the number of vertices of $D(S)$ and that does not depend on the size of T. Notice that the bound $1 + 2\lceil k/2 \rceil$ is equal either to $k + 1$ (if k is even) or to $k + 2$ (if k is odd). This implies that the difference between this upper bound and the lower bound given in Lemma 2 is at most 5.

We start by providing two lemmas that are the technical foundation of our drawing technique. The first lemma sheds more light on a point-set embeddability problem studied by Kaneko and Kano [9,10].

Lemma 3. *Let G consist of a forest of trees $T_0, T_1, \ldots, T_{h-1}$. Let $T_i = (V_i, E_i)$ for all $0 \leq i < h$. Let $S = \{p_0, p_1, \ldots, p_{n-1}\}$ be a set of points in general position such that $p_0, p_1, \ldots, p_{h-1}$ are points of the convex hull of S. There exists an $O(n^2 \log n)$-time algorithm that computes a geometric point-set embedding $\Gamma(G, S)$ such that the root of T_i is on p_i $(0 \leq i < h)$.*

Sketch of Proof: Let $CH(S)$ be the convex hull of S. Without loss of generality, assume that $p_0, p_1, \ldots, p_{h-1}$ occur in this order on the boundary of $CH(S)$ in clockwise order (if this is not the case, we can simply reorder them).

We first show that we can find a line a with the following properties: (i) a does not intersect any point p_i of S and there are points from S on both sides of a; (ii) denoted by $I \subset \{0, 1, \ldots, h - 1\}$ the set of indices for which all convex hull points p_j, with $j \in I$, lie on one side of a, we have that the total number of points on that side is equal to $\sum_{j \in I} |V_j|$.

We call such a line a *dividing line*. An example of dividing line is shown in Figure 4. We can use a ham-sandwich type argument to prove that a dividing line exists. We say that a side of a is too light if we have convex hull points p_j with $j \in I \subset \{0, 1, \ldots, h - 1\}$ to that side of a and the total number of points to that side of a is smaller than $\sum_{j \in I} |V_j|$. If one side of a is too light, the other side is said to be too heavy.

Consider points p_0 and p_1 on $CH(S)$. Let a_0 and a_1 be lines through p_0 and p_1 and such that any other point of a_0 and a_1 is outside the polygon defined by $CH(S)$ (refer to Figure 4 for an illustration). Let p be the intersection point of a_0 and a_1 (the proof will still work if p is a point at infinity). We start with line $a = a_0$ and rotate a around p in the counterclockwise direction until $a = a_1$. We can always slightly perturb a_0 and a_1 (and hence p) in such a way that a never intersects two points of S at the same time. Without loss of generality assume that the range of motions for a does not include a horizontal line and that when $a = a_0$, all remaining points of S lie to the right of a when moving along a toward p. If we rotate a slightly away from a_0, only p_0 lies to the left of a. If T_0 consists only of its root, a is a dividing line and we are done; otherwise the left side of a is too light. If we place a such that only p_1 is on its right, then either T_1 consists only of a root and we are done, or the left side of a is too heavy. If the left side of a is too light and during the rotation of a from a_0 to a_1 it passes a point p_j with $0 \leq j < h$, the left side of a remains too light. Since during rotation at any time at most one point moves from the right to the left side of a, and since in the beginning the left side of a is too light and at the end the left side of a is too heavy, it follows that at some moment a is a dividing line.
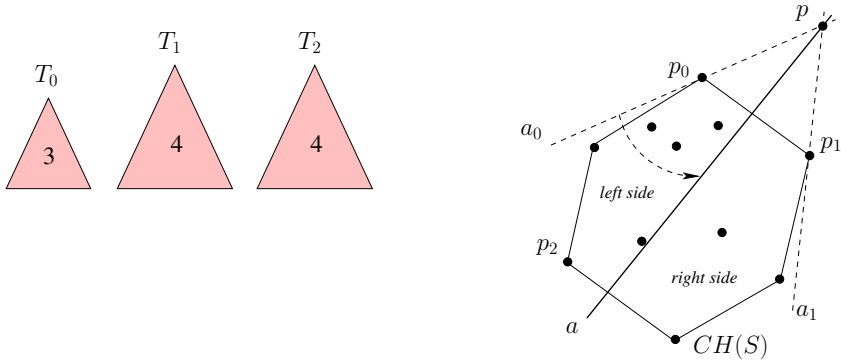
Fig. 4. An example of dividing line a. G consists of three trees T_0, T_1, and T_2, where T_0 has 3 vertices and T_1, T_2 have 4 vertices each. On one side of a there are the root points p_0 and p_2 for trees T_0 and T_2, and a total number of points equal to the number of vertices of T_0 and T_2; on the other side of a there are the points for drawing T_1.

The complexity of finding a dividing line is $O(n \log n)$, because we can first radially sort the points of S around p and then execute a scan-line algorithm from a_0 to a_1 to find a dividing line. Also, if p must be perturbed by an $\epsilon > 0$ to avoid that it is collinear with any two points of S before starting the search of a dividing line, such an ϵ can be determined in $O(n)$ time, using the radial sorting of the points around p itself.

Once we have found a dividing line, the polygon whose boundary is $CH(S)$ is divided into two subregions. By recursively applying the same procedure on each of the two subregions we can find dividing lines that split $CH(S)$ into convex subregions $P_0, P_1, \ldots, P_{h-1}$ such that each P_i contains $|V_i|$ vertices. Therefore we find the required drawing by executing the following algorithm:

Step A. Divide $CH(S)$ into convex subregions $P_0, P_1, \ldots, P_{h-1}$ such that each P_i contains $|V_i|$ vertices.

Step B. Draw each T_i inside P_i with the technique of Bose et al. [3].

Since all h dividing lines can be found in $O(h \cdot n \log n)$ time, where $h \leq n$, and the algorithm of Bose et al. [3] runs in $O(n \log n)$ time, it follows that the given algorithm runs in $O(n^2 \log n)$. □

The next lemma extends the previous result to the case where the roots of the trees are placed on the boundary of a non-convex polygon. In this case, the number of bends along the edges depend on the number of reflex corners of the polygon.

Lemma 4. *Let G consist of a forest of trees $T_0, T_1, \ldots, T_{h-1}$. Let $T_i = (V_i, E_i)$ for all $0 \leq i < h$. Let $S = \{p_0, p_1, \ldots, p_{n-1}\}$ be a set of points in general position such that $p_0, p_1, \ldots, p_{h-1}$ are points along the boundary of a polygon P and the remaining points of S are inside P. Also, let k be the number of reflex*

corners of P. There exists an $O(n^2 \log n)$-time algorithm that computes a point-set embedding $\Gamma(G, S)$ inside P such that the root of T_i is on p_i ($0 \le i < h$) and each edge of $\Gamma(G, S)$ has at most $2\lceil k/2 \rceil$ bends.

Sketch of Proof: For an illustration of this proof, refer to Figure 5. In the figure the forest to be drawn consists of two trees, T_0 and T_1, and the polygon P has three reflex corners. We prove the lemma by construction:

Step 1. We partition P into $k + 1$ convex polygons, for example by iteratively drawing a bisector from each reflex vertex until this bisector hits another line segment. We perturb the subdivision in such a way that no point from S lies on any of the added subdivision edges and so that none of the convex polygons has an angle equal to π. We call the added subdivision edges *dummy edges* (the dashed edges in the figure).

Step 2. Consider the dual graph of this subdivision, find a spanning tree T of the dual graph and select a node r of T with the property that the number of edges in T from r to any leaf node of T is at most $\lceil k/2 \rceil$. Make r the root of T. In the following, for any node v of T, P_v will denote the convex polygon corresponding to v (P_r is the convex polygon corresponding to the root). In the figure, the nodes of the dual graph of the subdivision are represented by big squares and the edges of the selected spanning tree are in bold.

Step 3. The objective of this step is to add extra points on the boundary of P_r so that each of these points corresponds to a distinct point of S that does not lie in P_r. Let $S' = S$. We add *dummy points* to S' by executing a post-order traversal of T. For each visited node v of T distinct from r we do the following. Let e be the dummy edge that separates P_v from the polygon corresponding to its parent node in T. Recall that no points from S lie on e, except possibly at its end-points. Let S_v be the set of points in $S' \cap P_v$ except possibly those at some end-point of e (we include every other point of S that is on the boundary of P_v). Place $|S_v|$ dummy points on e in such a way that none of the points on e lies on a line through two points from S_v. Construct a straight-line perfect planar matching from the $|S_v|$ points in P_v to the $|S_v|$ points on e. Add to S' the dummy points placed on e. In the figure, the dummy points are represented by empty circles.

Step 4. After the execution of Step 3 all nodes of T have been visited except the root r of T. Notice that there are n points in $S' \cap P_r$. In order to guarantee that no three points of S' are collinear, we slightly modify the boundary of P_r, by replacing each dummy edge of P_r with a "slightly convex" polygonal chain. More precisely, if e is a dummy edge of P_r such that n_e dummy points are placed on e, we replace e with a convex polygonal chain C_e such that C_e has n_e vertices and it does not change the inside/outside relations of the points of S with respect to P_r. Then we move each of the n_e dummy points on a distinct vertex of C_e, in such a way that the linear ordering of these points along C_e is the same they had along e.

Step 5. Compute a straight-line drawing of $\Gamma(G, S' \cap P_r)$ inside P_r by using Lemma 3; the root of each T_i ($i \in \{0, \ldots, h-1\}$) is placed either on p_i (if

p_i belongs to the boundary of P_r) or on the dummy point of the boundary of P_r that corresponds to p_i. Finally, we replace all edges of the drawing connected to a dummy point by narrow tunnels and we use these tunnels to planarly draw the edges of the tree; the number of dummy nodes traversed by an edge of the tree corresponds to the number of bends of that edge in the final drawing.

In Step 5, any edge of G is drawn from a point p of S via dummy points until it reaches P_r. Every time an edge passes through a point on a dummy edge, a bend is added. Since the longest path in the spanning tree T of the subdivision passes through $\lceil k/2 \rceil$ dummy edges and an edge of the drawing of G may connect two points that lie in two (possibly coincident) polygons whose corresponding nodes are at distance $\lceil k/2 \rceil$ from the root r of T, the number of bends per edge is at most $2\lceil k/2 \rceil$.
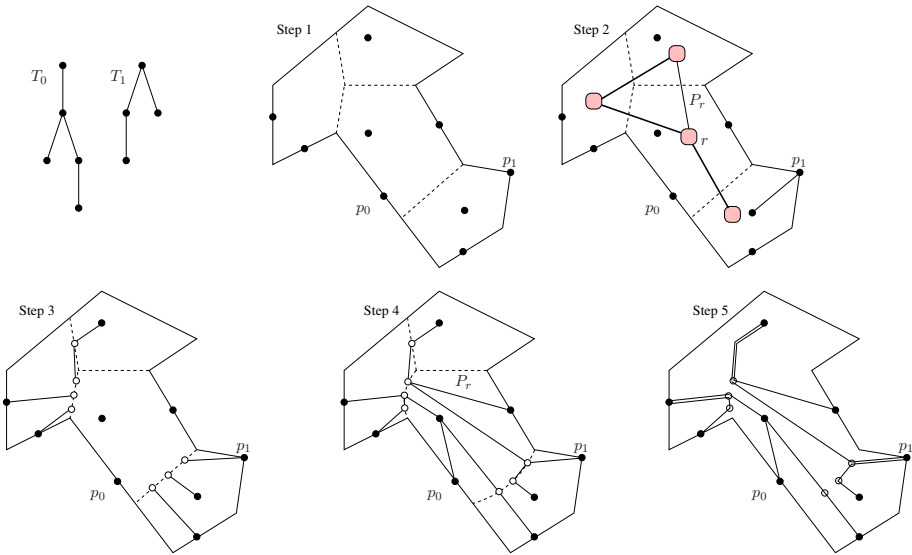

Fig. 5. Steps of the constructive proof of Lemma 4

We now briefly discuss the time complexity of the drawing algorithm described above. Step 1 can be performed in $O(n^2)$ time by using standard partitioning techniques of a polygon into convex regions (see, e.g., [14]). Step 2 is executed in $O(k)$ time, because T has $k + 1$ vertices. Regarding Step 3, for each polygon $P_v \neq P_r$ we need to project the points of S_v on a dummy edge e of the boundary of P_v, so that the straight-line edges used in the projection do not intersect. To accomplish this we can for example radially sort the points of S_v by rotating counterclockwise the line containing e around the middle point q of e, and then project all points of S_v with slope less than $\pi/2$ onto points of e to the right of q, and points of S_v with slope greater than $\pi/2$ onto points of e to the left of q (see

Figure 6 for an example). Since each S_v contains at most n points, this step can be executed in $O(n^2 \log n)$ time. Step 4 can be executed in $O(n)$ time, and the complexity of Step 5 is dominated by the complexity of the drawing algorithm of Lemma 3, which is $O(n^2 \log n)$.   □
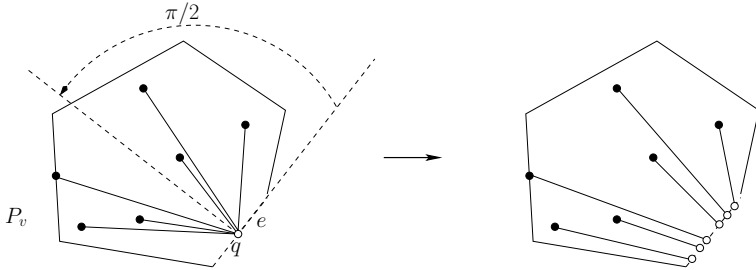

Fig. 6. Illustration of the technique used in Step 3 in the proof of Lemma 4

We can now prove the following theorem.

Theorem 1. *Let T be a tree with n vertices and let S be a set of n points in general position. Let $D(S)$ be a partial drawing of T representing a subtree with k vertices. There exists a constrained point-set embedding $\Gamma(T, D(S))$ with at most $1 + 2\lceil k/2 \rceil$ bends per edge, which can be computed in $O(n^2 \log n)$ time.*

Sketch of Proof: Since $D(S)$ is a partial drawing of T on S, it is a straight-line drawing of a subtree T' of T. We first construct a polygon P that follows the boundary of T' and that leaves T' outside (refer to Figure 7 for an illustration). More precisely, draw a convex polygon P that properly contains S and then modify it as follows: Find a location p on a side of P from which we can draw a straight-line segment to a location q on an edge of $D(S)$. Cut P at p and draw a line segment from p in the direction of q until it almost reaches q. We then continue to draw line segments that trace around the edges of $D(S)$. Once we have gone around the tree T' and are almost back at q, we draw a line segment back to the original boundary of P, close to p; in other words we have cut a tracing of T' out of P, while keeping inside the polygon all other points of S. For each vertex v of T', polygon P has $deg(v)$ corners close to v, where $deg(v)$ denotes the degree of v in T'. At the corner of P closest to a leaf of T' there is an angle almost equal to 2π (i.e., this vertex is a reflex corner of P). At the corners of P close to an internal vertex of T' there is at most one angle larger than π. Near the points p and q the polygon P has angles less than π. Therefore, P has at most one reflex corner for each vertex of T', and hence it has at most k reflex corners.

We now place k dummy points on the boundary of P close to the k vertices of T'. Namely, for each vertex v of T' we place a dummy point p_v close to v on the boundary of P. For a vertex v of T', let T_v denote the subtree of T rooted at v and consisting only of the edges of T that do not belong to T'. Using Lemma 4,

we construct each subtree T_v so that its root is placed on p_v instead of v. From Lemma 4 we know that this drawing has at most $2\lceil k/2 \rceil$ bends per edge, and can be computed in $O(n^2 \log n)$ time. Then, for each v we connect p_v to v with a straight-line segment and remove the dummy point p_v, thus creating one more bend per edge. The theorem follows.   □
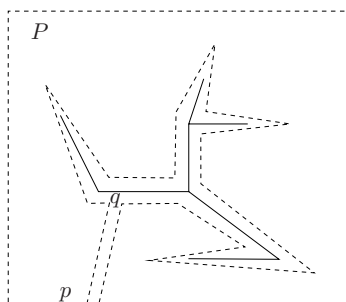

Fig. 7. Cutting a tree out of a polygon P. The tree is represented by solid edges while P has dashed segments.

The next result is a consequence of the proof of Theorem 1. Indeed, in that proof T' is an arbitrarily chosen subtree of T among those isomorphic to $D(S)$.

Corollary 1. *Let T be a tree with n vertices, S a set of n points in general position, and T' any subtree of T with k vertices. If $\Gamma(T', S)$ is a geometric point-set embedding of T' on a subset of S, then T has a point-set embedding $\Gamma(T, S)$ on S such that $\Gamma(T', S) \subset \Gamma(T, S)$ and every edge that does not belong to T' has at most $1 + 2\lceil k/2 \rceil$ bends. Also, $\Gamma(T, S)$ can be computed in $O(n^2 \log n)$ time.*

4   Conclusions and Open Problems

This paper introduced the problem of computing a point-set embedding of a graph G on a set S of points, with the constraint that a partial straight-line planar drawing of G on a subset of S is given. We concentrated on trees, and presented lower and upper bounds to the maximum number of bends per edge. We showed a lower bound equal to $k-3$ and an upper bound equal to $1 + 2\lceil k/2 \rceil$, where k is the number of vertices of the partial drawing. The upper bound is proved by means of an $O(n^2 \log n)$-time drawing algorithm. The drawing technique exploits a partial solution of a well-investigated and still unsolved computational geometry problem.

We mention in the following three open problems related to the results of this paper and that could be the subject of further investigation: (i) Extend the study to families of graphs other than trees. (ii) Compute constrained point-set embeddings with the minimum number of bends. (iii) Study the constrained

point-set embeddability problem in the case that the partial drawing to be extended contains some bends along its edges.

References

1. Badent, M., Di Giacomo, E., Liotta, G.: Drawing colored graphs on colored points. In: WADS 2007. LNCS, Springer, Heidelberg (2007)
2. Bose, P.: On embedding an outer-planar graph on a point set. Computational Geometry: Theory and Applications 23, 303–312 (2002)
3. Bose, P., McAllister, M., Snoeyink, J.: Optimal algorithms to embed trees in a point set. Journal of Graph Algorithms and Applications 2(1), 1–15 (1997)
4. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing. Prentice-Hall, Upper Saddle River (1999)
5. Di Giacomo, E., Didimo, W., Liotta, G., Meijer, H., Trotta, F., Wismath, S.K.: k-colored point-set embeddability of outerplanar graphs. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 318–329. Springer, Heidelberg (2007)
6. Di Giacomo, E., Liotta, G., Trotta, F.: On embedding a graph on two sets of points. International Journal of Foundations of Computer Science, Special Issue on Graph Drawing 17(5), 1071–1094 (2006)
7. Halton, J.H.: On the thickness of graphs of given degree. Information Sciences 54, 219–238 (1991)
8. Ikebe, Y., Perles, M., Tamura, A., Tokunaga, S.: The rooted tree embedding problem into points in the plane. Discrete Comput. Geometry 11, 51–63 (1994)
9. Kaneko, A., Kano, M.: Straight line embeddings of rooted star forests in the plane. Discrete Appl. Mathematics 101, 167–175 (2000)
10. Kaneko, A., Kano, M.: Semi-balanced partitions of two sets of points and embeddings of rooted forests. Int. J. Comput. Geom. Appl. 15(3), 229–238 (2005)
11. Kaufmann, M., Wagner, D. (eds.): Drawing Graphs. LNCS, vol. 2025. Springer, Heidelberg (2001)
12. Kaufmann, M., Wiese, R.: Embedding vertices at points: Few bends suffice for planar graphs. Journal of Graph Algorithms and Applications 6(1), 115–129 (2002)
13. Nishizeki, T., Rahman, M.S.: Planar Graph Drawing. Lecture Notes Series on Computing, vol. 12. World Scientific, Singapore (2004)
14. O'Rourke, J.: Art Gallery Theorems and Algorithms. Oxford Univ. Press, Oxford (1987)
15. Pach, J., Wenger, R.: Embedding planar graphs at fixed vertex locations. Graphs and Combinatorics 17, 717–728 (2001)
16. Patrignani, M.: On extending a partial straight-line drawing. International Journal of Foundations of Computer Science, Special Issue on Graph Drawing 17(5), 1061–1069 (2006)
17. Preparata, F.P., Shamos, M.I.: Computational Geometry: An Introduction, 3rd edn. Springer, Heidelberg (1990)