Turkish Journal of Mathematics

http://journals.tubitak.gov.tr/math/

Research Article

Turk J Math
(2015) 39: 820 – 829
© TÜBİTAK
doi:10.3906/mat-1410-12

Planar embedding of trees on point sets without the general position assumption

Asghar ASGHARIAN SARDROUD, Alireza BAGHERI*
Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran

Received: 06.10.2014 • **Accepted/Published Online:** 17.04.2015 • **Printed:** 30.11.2015

Abstract: The problem of point-set embedding of a planar graph G on a point set P in the plane is defined as finding a straight-line planar drawing of G such that the nodes of G are mapped one to one on the points of P. Previous works in this area mostly assume that the points of P are in general position, i.e. P does not contain any three collinear points. However, in most of the real applications we cannot assume the general position assumption. In this paper, we show that deciding the point-set embeddability of trees without the general position assumption is NP-complete. Then we introduce an algorithm for point-set embedding of n-node binary trees with at most $\frac{n}{3}$ total bends on any point set. We also give some results when the problem is limited to degree-constrained trees and point sets having constant number of collinear points.

Key words: Point-set embedding, tree embedding, general position assumption, graph drawing, minimum bend embedding

1. Introduction

Given an n-node planar graph G, its embedding on a point set P of n points in the plane is a planar straight-line drawing of G in which the nodes of G are mapped one to one on the points of P.

Trees were the first class of graphs for which point-set embeddability was investigated [14], and it was shown that they are always embeddable on point sets in general position [11]. A point set is in general position if it does not contain any three collinear points. Bose et al. introduced an optimal $O(n \log n)$ time algorithm for embedding rooted trees on point sets in general position such that the root node is placed on a given point [3]. Castaneda and Urrutia showed that the most general class of graphs that are always embeddable on any point set in general position is the outer-planar graphs [6]. Later, Bose proposed a faster algorithm for embedding outer-planar graphs [2].

The point-set embeddability of general planar graphs is NP-complete [5], even if the point set is in general position. Moreover, Kaufmann et al. [12] showed that 1-bend embeddability of planar graphs is NP-complete, while any planar graph is 2-bend embeddable on any point set. Graph G is k-bend embeddable on P if it has a drawing on P with at most k bends on each of its edges. Pach and Wenger [13] considered a variant of the problem in which the mapping between the vertices of the graph and the points is fixed. They showed that, in this case, some edges may need $O(n)$ bends to be embedded, and the resulting embedding may contain $O(n^2)$ bends in total.

*Correspondence: ar_bagheri@aut.ac.ir
2010 *AMS Mathematics Subject Classification:* 68R10, 68W01, 05C85.

Colored embedding and simultaneous embedding are other variants of the point-set embedding problem that have been considered in this area. In colored embedding, the nodes of the given graph and the given points are colored and each node should be mapped on a point of the same color [1, 7]. In simultaneous embedding, two given graphs with the same set of vertices should be embedded on the same point set [4, 8, 9, 10].

Most of the above-mentioned algorithms assume that the point set is in general position, which may not be a realistic assumption in some applications. For example, in VLSI design applications and grid graph drawing, the point set on which a graph is to be embedded contains many collinear points. In this paper, we consider embedding of trees on point sets that are not in general position.

In the remainder of the paper, Section 2 shows that without the general position assumption the point-set embeddability of trees is NP-complete. In Section 3, we first show that trees are 1-bend embeddable on any point set, and then we describe our 1-bend embedding algorithm for n-node binary trees, which produces at most $\frac{n}{3}$ bends in total. We present a straight-line embedding algorithm for ternary trees on point sets having no four collinear points in Section 4, which shows that ternary trees are always embeddable on point sets that have no four collinear points. In this section, we also give two examples that show that the results of this section cannot be extended to 4-ary trees or point sets without five collinear points. Finally, the conclusion and summary are given in Section 5.

2. The NP-completeness result

In this section, we prove that the point-set embeddability of trees without the general position assumption is NP-complete. We prove this by reducing the well-known NP-complete problem 3-partition to the embedding problem.

Given a positive integer B and a multiset $S = \{a_1, .., a_n\}$ containing $n = 3m$ positive integers between $\frac{B}{4}$ and $\frac{B}{2}$, the *3-partition* problem is to decide whether S can be partitioned into m multisets S_1, S_2, .., S_m such that the sums of the numbers of multisets are equal (i.e. for any $1 \leq i \leq m$, $\sum_{a_j \in S_i} a_j = B$). Note that, due to the constraints $\frac{B}{4} < a_i < \frac{B}{2}$, each subset S_i should contain exactly 3 elements and B should be at least 3. The 3-partition problem is strongly NP-complete, i.e. it is NP-complete even if the value of each integer $a_i \in S$ is bounded by a polynomial on n. Thus, we assume $N = mB$ is polynomially bounded with respect to n.

Theorem 2.1 *Let T be an n-node tree and P be a set of n points with integer coordinates on the plane. Deciding whether T is embeddable on P is an NP-complete problem.*

Proof Obviously, the problem is in NP, because given an embedding of T on the points of P one can check its correctness (including planarity) in polynomial time. For NP-hardness, given an instance of the 3-partition problem as defined before, we construct a tree T and a point set P as follows:

Tree T consists of a root node u that has $(B+1)N+n$ children $v_1, v_2, .., v_n$ and $w_1, w_2, .., w_{(B+1)N}$. Each node v_i, $1 \leq i \leq n$, is connected by an edge to the end node of a path π_i consisting of a_i nodes (see Figure 1).

Point set P consists of $(B+2)N+n+1$ points as follows:

- point p located at $(0,0)$
- m points $q_1, q_2, .., q_m$ respectively located at $(0,1), (1,1), ..., (m-1,1)$ (illustrated by circles in Figure 2)

- $2m$ points $q_{m+1}, q_{m+2}, .., q_{3m}$ with the coordinates $(\pm 1, B+1), ((B+1)\pm 1, B+1), ..., ((m-1)(B+1)\pm 1, B+1)$ (illustrated by squares in Figure 2)
- $N = mB$ points $s_1, s_2, .., s_N$ located at $(2i, 2), (3i, 3), ..., ((B+1)i, B+1)$, $0 \le i < m$ (illustrated by bullets in Figure 2)
- $(B+1)N$ points $r_1, r_2, .., r_{(B+1)N}$ located at $(-i^2N + j, i)$, $1 \le i \le B+1$ and $0 \le j < N$ (illustrated by crosses in Figure 2)

To complete the proof, we need to show that T is embeddable on P if and only if the answer to the given instance of the 3-partition is YES.

First, assume that T is embeddable on P. In such an embedding, the root node u has to be embedded on the point p, because u has $(B+1)N+n$ children and each point $p_i \in P \setminus \{p\}$ with coordinates (x_i, y_i) is collinear with $N+m-1$ points of P on the line $y = y_i$, and therefore the number of points visible from p_i is not more than $(B+1)N + 2m + 4$, which is less than $(B+1)N+n$ (without loss of generality we can assume that $m > 4$).

Furthermore, all the children of u have to be embedded on distinct points of the point set $\{q_1, q_2, .., q_n, r_1, r_2, .., r_{(B+1)N}\}$, because u has $(B+1)N+n$ children and only $(B+1)N+n$ points, i.e. $q_1, q_2, .., q_n$ and $r_1, r_2, .., r_{(B+1)N}$, are visible from p. Moreover, because the nodes $v_1, v_2, .., v_n$ have some descendants, they have to be embedded on points $q_1, q_2, .., q_n$, and the remaining children of u, i.e. $w_1, w_2, .., w_{(B+1)N}$, have to be embedded on points $r_1, r_2, .., r_{(B+1)N}$.

After embedding u and its children, the remaining points of P are partitioned into m groups of points, each consisting of B collinear points such that no point of any group is visible from any point of another group. Therefore, each path in the set $\Pi = \{\pi_1, \pi_2, .., \pi_n\}$ should be embedded entirely on one of these groups. This partitions the set Π into m subsets such that the total number of nodes in all paths of a subset is B. Based on the partitioning of Π, one can construct the desired partitioning for the multiset S.

Conversely, given a solution for the 3-partition problem, we can construct an embedding of T on the point set P as follows:

- embed the root node u on the point p
- for each $S_i = \{a_{i_1}, a_{i_2}, a_{i_3}\}$
  - embed $v_{i_1}, v_{i_2}, v_{i_3}$ respectively on points $(i, 1)$, $(i(B+1)+1, B+1)$ and $(i(B+1)-1, B+1)$
  - embed the three paths $\pi_{i_1}, \pi_{i_2}, \pi_{i_3}$ on points lying on the line $y = ix$ as shown in Figure 3
- embed nodes $w_1, w_2, .., w_{(B+1)N}$ respectively on points $r_1, r_2, .., r_{(B+1)N}$

$\square$

3. 1-Bend embeddability of trees

Kaufmann and Wiese [12] have shown that any plane graph having an external Hamiltonian cycle has a 1-bend embedding on any point set. An external Hamiltonian cycle is a Hamiltonian cycle containing at least one edge of the outer face. This shows that trees are 1-bend embeddable on any point set, because it is possible to add some edges to any tree so that it contains an external Hamiltonian cycle. However, in this section, we

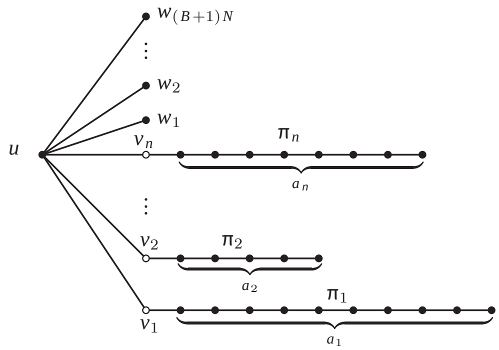

Figure 1. The constructed tree T of Theorem 2.1.

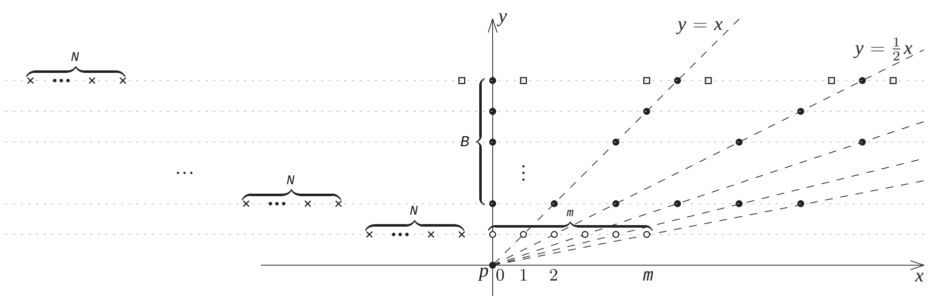

Figure 2. Configuration of the constructed point set P of Theorem 2.1.

introduce another method for 1-bend embedding of trees, which can be modified to produce 1-bend embeddings with at most $\frac{n}{3}$ total bends for n-node binary trees. Our 1-bend embedding algorithm is an extension of a tree embedding algorithm on point sets in general position in which the root of the given tree is placed on a convex hull point, the remaining points are partitioned, and subtrees are embedded recursively [14, 3]. By convex hull points of a point set, we mean the extreme points of its convex hull, i.e. the points of the convex hull that do not lie in any open line segment joining two other points of the convex hull. A point p is visible from q in a point set P, if no other point in P is on the line segment connecting p and q. Furthermore, in the remainder of this section let δ be the smallest angle among all the angles formed by the triples of noncollinear points of the point set P. The following lemma describes our algorithm.

Lemma 3.1 *Any n-node tree T has a 1-bend embedding on any point set P of n points such that its root is embedded on a given convex hull point p of P.*

Proof We prove the lemma by giving a 1-bend embedding algorithm. Let r be the root of T and $r_1, .., r_m$ be its children and T_i be the subtree of T rooted at r_i. First, we partition the point set P into m point sets $P_1, .., P_m$ of sizes $|T_1|, .., |T_m|$. Let p' be the convex hull point of P consequent to p, and $< p_1, .., p_{n-1} >$ be the sorted list of points in $P \setminus \{p\}$, sorted in increasing order lexicographically based on the pair (α_i, d_i), where α_i is the angle $\angle p_i p p'$ and d_i is the Euclidean distance between points p_i and p. We partition P into

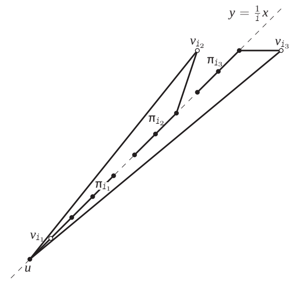

Figure 3. Desired embeddings of three paths π_{i1}, π_{i2}, and π_{i3} in the proof of Theorem 2.1
.

point sets $P_1 = \{p_1, .., p_{|T_1|}\}$, $P_2 = \{p_{|T_1|+1}, .., p_{|T_2|}\}$, .., $P_m = \{p_{n-|T_m|}, .., p_{n-1}\}$. Because of our partitioning method, the convex hulls of the point sets $P_1, P_2, .., P_m$ are disjoint. Therefore, we embed r on point p, and for $1 \leq i \leq m$, we recursively embed tree T_i on point set P_i such that r_i is embedded on the first point of P_i say p_{s_i}. If p_{s_i} is visible from p, we embed edge (r, r_i) on line segment $\overline{p_{s_i}p}$. Otherwise, there should be some points of P, say $p_j, .., p_{s_i-1}$ for some $j < s_i$, on the line segment $\overline{p_{s_i}p}$. In this case, let q_i be a point in the plane such that $\angle q_ipp_1 > \angle p_{s_i}pp_1$, and p, q_i and p_{s_i} form an isosceles triangle with base edge $p_{s_i}p$ and base angles equal to $\frac{i\delta}{m+1}$, and embed edge (r, r_i) on line segments $\overline{pq_i}$ and $\overline{q_ip_{s_i}}$ (see Figure 4(a) for illustration).

Because subtrees are embedded on subsets of P with disjoint convex hulls, the only thing to be proved for the correctness of the algorithm is that line segments $\overline{pq_i}$ and $\overline{q_ip_{s_i}}$ do not cross any other edge of the embedding. Assume, as a contradiction, that an edge of the drawing connecting points p_x and p_y crosses line segments $\overline{pq_i}$ or $\overline{q_ip_{s_i}}$. Because of the magnitude of the angle $\angle pq_ip_{s_i}$, no two edges of embedding incident to p can cross. Thus, without loss of generality let p_x and p_y be different from p. Both p_x and p_y should be on the same side of the line going through p and p_{s_i}, i.e. the side on which q_i lies, as illustrated in Figure 4(b), as otherwise p_x and p_y should be on the different subsets of points of P and the algorithm does not embed any edge between them. However, this shows that each triple of points p, p_{s_i}, p_x, and p_y forms an angle less than δ, a contradiction. □

The algorithm, given by Lemma 3.1, may produce $n-1$ bends in embedding of an n-node binary tree. In the following, we present an improved version of the algorithm for 1-bend embedding of binary trees that produces at most $n/3$ total bends. The algorithm uses Lemma 3.2 to partition $P \setminus \{p\}$ into two point sets and recursively embed the subtrees. The problematic case is when all the points of P are collinear. In this case r can be connected to each of its children by a polyline.

Lemma 3.2 *Let P be a set of n points, not all of them collinear, and p be one of its convex hull points. For any pair of integers n_1 and n_2 where $n_1 + n_2 + 1 = n$, there is a partitioning of $P \setminus \{p\}$ into two point sets P_1 and P_2 of sizes n_1 and n_2 such that convex hulls of P_1 and P_2 are disjoint and each of them has a point visible from p.*

Proof Let q be the convex hull point of P adjacent to p and $< p_1, .., p_{n-1} >$ be the sorted list of points in $P \setminus \{p\}$, sorted in increasing order lexicographically based on the pair (α_i, d_i), where α_i is the angle $\angle p_ipq$ and d_i is the Euclidean distance between points p_i and p. Partition $P \setminus \{p\}$ into two point sets $P_1 = \{p_1, .., p_{n_1}\}$ and $P_2 = \{p_{n_1+1}, .., p_{n-1}\}$. Clearly, convex hulls of P_1 and P_2 are disjoint and p_1 is visible from p. Thus, if P_2 has a point visible from p, sets P_1 and P_2 form the desired partitioning. Otherwise, there must be more than n_2 points in $P \setminus \{p\}$ on the line passing through p and p_{n-1}, say points $p_j, .., p_{n-1}$, for some $j \leq n_1$. In this case, the sets $P_2 = \{p_j, .., p_{j+n_2-1}\}$ and $P_1 = P \setminus (P_2 \cup \{p\})$ form the desired partitioning. □

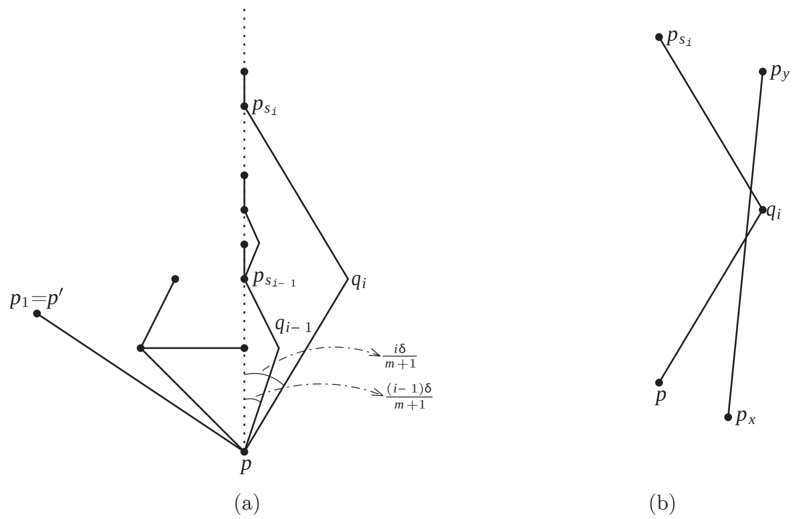

Figure 4. Embedding (a) the edges that require one bend in their embedding, and (b) the situation that two edges of the embedding may cross.

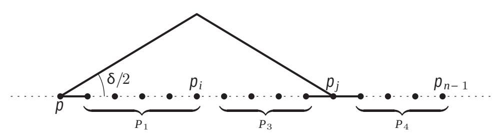

Figure 5. Embedding a binary tree on a set of collinear points.

Algorithm 1 describes our solution for the 1-bend embedding problem. In this algorithm, if the root of tree T has only one child, it will be embedded as described in lines 3 and 4. In the case of two child nodes, if all the points of P are not collinear, the algorithm partitions the point set by Lemma 3.2 and embeds subtrees recursively as described in lines 23 to 26, and connects the root to its children using straight-line edges. Otherwise, in lines 14 to 21, it embeds subtrees recursively, and connects the root to one of its children by a straight line and the other one via two line segments, as illustrated in Figure 5.

Lemma 3.3 *If r is the root of an n-node binary tree T and p is a convex hull point of a set of n points P, procedure EmbedBinaryTree(T, r, P, p) given by Algorithm 1 creates a 1-bend embedding of T on P.*

Proof The algorithm embeds each node of T on a distinct point of P, and each edge on one or two line segments. Therefore, it suffices to show that no two edges of the embedding cross each other. However, similar to the proof of Lemma 3.1, this also follows from the fact that the algorithm embeds subtrees recursively on point sets with disjoint convex hulls and the angle $\frac{\delta}{2}$ is smaller than any angle created by triples of points of P. $\square$

In an embedding of a tree T, node u is a *bend node* if one of the edges of the embedding, connecting u to its children, has some bends.

Algorithm 1 The embedding algorithm

procedure EmbedBinaryTree(T,r,P,p)

input:

an n-node binary tree T rooted at r, a set of n points P, and p a convex hull point of P

output:

a 1-bend embedding of T on P in which r is embedded on p and has at most $\frac{n}{3}$ total bends

1: embed r on p
2: **if** r has only one child r_1 **then**
3: **let** p_1 be the nearest convex hull point of $P \setminus \{p\}$ to p
4: EmbedBinaryTree($T \setminus \{r\}$, r_1, $P \setminus \{p\}$, p_1)
5: **end if**
6: **if** r has two children r_1 and r_2 **then**
7: **let** T_1 and T_2 be the two subtrees of T respectively rooted at r_1 and r_2
8: **if** both r_1 and r_2 have two children **then**
9: **let** r_1 be the child of r such that: $|T_1| \neq 3k'$ unless also $|T_2| = 3k$, $(k, k' \in \mathbb{Z}^+)$
10: **else**
11: **let** r_1 be the child of r that does not have two children
12: **end if**
13: **if** all the points of P are collinear **then**
14: **let** $p, p_1, p_2, ..., p_{n-1}$ be the sorted list of points in P along the line going through all of them
15: **let** T_3 and T_4 be the (possibly empty) subtrees of T_2 rooted at r_3 and r_4 respectively
16: **let** $i = |T_1|$ and $j = |T_1| + |T_3| + 1$
17: **let** $P_1 = \{p_1, ..., p_i\}$ and $P_3 = \{p_{i+1}, ..., p_{j-1}\}$ and $P_4 = \{p_{j+1}, ..., p_{n-1}\}$
18: EmbedBinaryTree(T_1, r_1, P_1 , p_1)
19: embed r_2 on p_j, and embed edge (r, r_2) on the polyline between p and p_j consisting of two line segments, which make angle $\frac{\delta}{2}$ with the supporting line of the point set, see Figure 5
20: EmbedBinaryTree(T_3, r_3, P_3 , p_{j-1}) if T_3 is not empty
21: EmbedBinaryTree(T_4, r_4, P_4 , p_{j+1}) if T_4 is not empty
22: **else**
23: partition P into two point sets P_1 and P_2 around p by Lemma 3.2
24: **let** q_1, q_2 be respectively the nearest convex hull points of P_1 and P_2 to p
25: EmbedBinaryTree(T_1, r_1, P_1 , q_1)
26: EmbedBinaryTree(T_2, r_2, P_2 , q_2)
27: **end if**
28: **end if**

Lemma 3.4 *Let b be the total number of bends of an embedding of an n-node binary tree T produced by Algorithm 1. Then $b \leq \frac{n}{3}$ if the root of T is a bend node and $b < \frac{n}{3}$ otherwise.*

Proof We prove the lemma by induction on n. For $n \leq 3$, the lemma holds clearly. Assuming the lemma is true for any binary tree of less than n nodes, we show that it is also true for an n node binary tree T rooted at r. When r has only one child r_1, $T \setminus \{r\}$ has $n-1$ nodes and by the induction hypothesis it is embedded with $b_1 \leq \frac{n-1}{3}$ bends in line 4 of the algorithm. Since edge (r, r_1) is embedded as straight line, we should have $b = b_1 < \frac{n}{3}$.

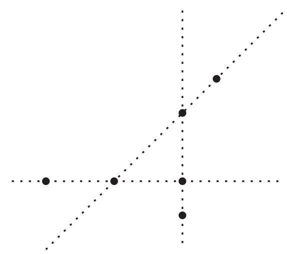

Figure 6. Set of six points in which for each point there is exactly one nonvisible point

In the case that r has two children r_1 and r_2, if all the points of P are not collinear, the algorithm embeds edges (r, r_1) and (r, r_2) without bends, and so applying the induction hypothesis to the recursive calls in lines 25 and 26, we have $b < \frac{n}{3}$. However, if all the points of P are collinear, r should be a bend node and we should prove that $b \leq \frac{n}{3}$.

In this case, because embedding of edge (r, r_2) in line 19 has a bend, b is equal to $b_1 + b_3 + b_4 + 1$, where b_1, b_3, and b_4 are respectively the total number of bends of the recursively created embeddings of T_1, T_3, and T_4 in lines 18, 20, and 21. If $|T_1| = 3k$ and $|T_2| = 3k'$ ($k, k' \in \mathbb{Z}^+$), either $b_3 < \frac{|T_3|}{3}$ or $b_4 < \frac{|T_4|}{3}$ because both $|T_3|$ and $|T_4|$ cannot be multiples of three. Otherwise, considering lines 9 and 11 of the algorithm, we have $b_1 < \frac{|T_1|}{3}$ because either $|T_1|$ is not a multiple of three or r_1 has only one child. Therefore, since $|T_1| + |T_2| + |T_3| = n - 2$, we have $b < \frac{n+1}{3}$, which implies that $b \leq \frac{n}{3}$ because b and n are integers. □

Theorem 3.5 summarizes the result.

Theorem 3.5 *Any n-node binary tree is 1-bend embeddable on any point set of n points with at most $\frac{n}{3}$ total bends.*

4. Straight-line embedding of trees on near-general-position point sets

The construction of the NP-completeness proof provided in Section 2 uses trees of maximum degree O(n) and point sets containing O(n) collinear points. Therefore, the complexity of the problem is still open if we limit the problem to the degree-constrained trees or limit the maximum number of collinear points in the point set to a constant. The following lemma shows that if we limit the point set P to have no four collinear points, the ternary trees are always embeddable on P.

Lemma 4.1 *Any n-node ternary tree T is embeddable on a set of n points P if there are no four collinear points in P.*

Proof Let r be the root of T, p be a convex hull point of P, and T_1, T_2 and T_3 be the possibly empty subtrees of T respectively rooted at r_1, r_2, and r_3 children of r. For the cases when $n = 1, 2$, the embedding is trivial. Moreover, when $n = 3$, if the three points of P are collinear, we should embed r on the middle point if r has degree two and on a nonmiddle point otherwise. Because point set P has no four collinear points, the number of visible points from each point of P is at least $\lceil \frac{n-1}{2} \rceil$. Therefore, when $n \geq 6$, at least three points of P are visible from each convex hull point p of P. Similarly, when $n = 4, 5$, there is always convex hull point p visible at least from three other points of P. Hence, in these cases, similar to the proof of Lemma 3.2, we can partition $P \setminus \{p\}$ to point sets P_1, P_2, and P_3 of sizes respectively $|T_1|$, $|T_2|$, and $|T_3|$ with disjoint convex hulls such that each partition contains at least one point visible from p. We embed r on p and for each subtree T_i, $1 \leq i \leq 3$, we recursively embed T_i on P_i such that r_i is embedded on a convex hull point of P_i visible from p. Note that, when $|T_i| = 3, 4, 5$, we should embed r_i on the suitable convex hull point of P_i as described before. □

The example point set shown in Figure 6 shows that not any 4-ary tree is embeddable on a point set having no four collinear points. A 4-ary tree consisting of a node of degree five adjacent to five nodes of degree one is not embeddable on the point set shown in this figure because the point set does not contain any point visible from the other five points. In addition, in the case that the point set P has four collinear points, even binary trees are not always embeddable. As an example, a binary tree consisting of a node of degree three adjacent to three nodes of degree one is not embeddable on a point set consisting of four collinear points. Therefore, Lemma 4.1 cannot be strengthened to the 4-ary trees or point sets without five collinear points.

5. Conclusions

We have shown that embeddability of trees on point sets is NP-complete. Our results also show that the embedding is always possible when the problem is limited to ternary trees and point sets without four collinear points, and this result cannot be strengthened any further. We also introduced an algorithm for embedding n-node binary trees on any set of n points with at most $\frac{n}{3}$ total bends. As future works, we suggest research on the problems of point-set embeddability of degree constrained trees and embeddability of planar graphs on point sets that have few collinear points.

References

[1] Badent M, Di Giacomo E, Liotta G. Drawing colored graphs on colored points. Theor Comput Sci 2008; 408: 129–142.

[2] Bose P. On embedding an outer-planar graph in a point set. Comp Geom-Theor Appl 2002; 23: 303–312.

[3] Bose P, McAllister M, Snoeyink J. Optimal algorithms to embed trees in a point set. J Graph Algorithms Appl 1997; 2: 1–15.

[4] Brass P, Cenek E, Duncan CA, Efrat A, Erten C, Ismailescu DP, Kobourov SG, Lubiw A, Mitchell JSB. On simultaneous planar graph embeddings. Comp Geom-Theor Appl 2007; 36: 117–130.

[5] Cabello S. Planar embeddability of the vertices of a graph using a fixed point set is NP-hard. J Graph Algorithms Appl 2006; 10: 353–366.

[6] Castañeda N, Urrutia J. Straight line embeddings of planar graphs on point sets. In: Fiala F, Kranakis E, Sack JR, editors. Proceedings of the 8th Canadian Conference on Computational Geometry; 12–15 August 1996; Ottawa, Ontario, Canada. Ottawa: Carleton University Press, 1996, pp. 312–318.

[7] Di Giacomo E, Liotta G, Trotta F. On embedding a graph on two sets of points. Int J Found Comput S 2006; 17: 1071–1094.

[8] Erten C, Kobourov SG. Simultaneous embedding of planar graphs with few bends. In: Pach J, editor. 12th International Symposium on Graph Drawing; 29 September–2 October 2004; New York, NY, USA. Berlin: Springer, 2005, pp. 195–205.

[9] Estrella-Balderrama A, Gassner E, Jünger M, Percan M, Schaefer M, Schulz M. Simultaneous geometric graph embeddings. In: Hong SH, Nishizeki T, Quan W,editors. 15th International Symposium on Graph drawing; 24–26 September 2007; Sydney, Australia. Berlin: Springer, 2008, pp. 280–290.

[10] Gassner E, Jünger M, Percan M, Schaefer M, Schulz M. Simultaneous graph embeddings with fixed edges. In: Fomin, FV, editor. 32nd Workshop on Graph-Theoretic Concepts in Computer Science, 22–24 June 2006; Bergen, Norway. Berlin: Springer, 2006, pp. 325–335.

[11] Ikebe Y, Perles MA, Tamura A, Tokunaga S. The rooted tree embedding problem into points in the plane. Discrete Comput Geom 1994; 11: 51–63.

[12] Kaufmann M, Wiese R. Embedding vertices at points: few bends suffice for planar graphs. J Graph Algorithms Appl 2002; 6: 115–129.

[13] Pach J, Wenger R. Embedding planar graphs at fixed vertex locations. Graph Combinator 2001; 17: 717–728.

[14] Pack J, Torocsik J. Layout of rooted trees. In: Trotter WT, editor. Planar Graphs, volume 9 of DIMACS Series. USA: American Mathematical Society, 1993, pp. 131–137.