PLANE 3-TREES: EMBEDDABILITY & APPROXIMATION*

STEPHANE DUROCHER† AND DEBAJYOTI MONDAL‡

Abstract. We give an $O(n \log^3 n)$-time linear-space algorithm that, given a plane 3-tree G with n vertices and a set S of n points in the plane, determines whether G has a point-set embedding on S (i.e., a planar straight-line drawing of G where each vertex is mapped to a distinct point of S), improving the $O(n^{4/3+\varepsilon})$-time $O(n^{4/3})$-space algorithm of Moosa and Rahman (2011). Given an arbitrary plane graph G and a point set S, Kaufmann and Wiese (2002) gave an algorithm to compute 2-bend point-set embeddings of G on S. Later, Di Giacomo and Liotta (2010) showed how such a drawing can be computed using $O(W^3)$ area, where W is the length of the longest edge of the bounding box of S. Their algorithm uses $O(W^3)$ area even when the input graphs are restricted to plane 3-trees. We introduce new techniques for computing 2-bend point-set embeddings of plane 3-trees that takes only $O(W^2)$ area. We also give approximation algorithms for point-set embeddings of plane 3-trees. Our results on 2-bend point-set embeddings and approximate point-set embeddings hold for partial plane 3-trees (e.g., series-parallel graphs and Halin graphs).

Key words. graph drawing, point-set embedding, plane 3-tree, approximation

AMS subject classifications. 05C10, 68R10

1. Introduction. A *planar drawing* of a graph G is an embedding (i.e., a mapping) of G onto the Euclidean plane $\mathbb{R}^2$, where each vertex in G is assigned a unique point in $\mathbb{R}^2$ and each edge in G is a simple curve in $\mathbb{R}^2$ joining the points corresponding to its endvertices such that no two curves intersect except possibly at their endpoints. A graph is *planar* if it has a planar drawing. A *straight-line drawing* of a planar graph is a planar drawing, where each edge is drawn as a straight line segment. The straight-line drawing style is popular since it naturally produces drawings that are easier to read and to display on smaller screens [13, 33, 36]. To meet the requirements of different practical applications, researchers have examined graph drawing problems under various constraints, e.g., when the vertices are constrained to be placed on a set of pre-specified locations [4, 9], or when a bijection between the vertices and locations are given [32]. If the pre-specified locations for placing the vertices of the input graph are points on the Euclidean plane, then we call the problem a point-set embedding problem. Such problems have applications in VLSI circuit layout, where different circuits need to be mapped onto a fixed printed circuit board [25], simultaneous display of different social and biological networks [7], and construction of a desired network among a set of fixed locations. Formally, a *point-set embedding* of a *plane graph* G (i.e., a fixed combinatorial planar embedding of G) with n vertices on a set S of n points is a straight-line drawing of G, where the vertices are placed on distinct points of S. Figures 1(a), (b) and (c) illustrate a plane graph G, a point set S and a point-set embedding of G on S, respectively.

Point-Set Embeddings. In 1994, Ikebe et al. [24] gave an $O(n^2)$-time algorithm to embed any tree with n vertices on any set of n points in *general position*, i.e., no three

*A preliminary version of the paper appeared in Debajyoti Mondal's M.Sc. thesis [28] and in the Proceedings of the 13th International Symposium on Algorithms and Data Structures (WADS 2013) [20].

†Department of Computer Science, University of Manitoba, Canada, durocher@cs.umanitoba.ca.
Work of the author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

‡Department of Computer Science, University of Manitoba, Canada, jyoti@cs.umanitoba.ca.
Work of the author is supported in part by a University of Manitoba Graduate Fellowship.

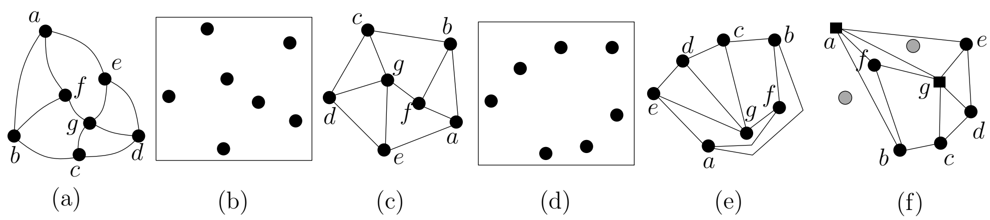

FIG. 1. *(a) A plane graph G. (b) A point set S. (c) A point-set embedding of G on S. (d) A point set S'. (e) A 2-bend point set embedding of G on S'. (f) A straight-line embedding of G, where $S'(\Gamma) = 5$. The vertices that are not mapped to any points of S' are shown in black square, and the points of S' that does not contain any vertex of G are shown in gray.*

points are collinear. Later, Bose et al. [5] devised a divide and conquer algorithm that runs in $O(n \log n)$ time. In 1996, Castañeda and Urrutia [10] gave an $O(n^2)$-time algorithm to construct point-set embeddings of maximal outerplanar graphs. Later, Bose [4] improved the running time of their algorithm to $O(n \log^3 n)$ using a dynamic convex hull data structure. In the same paper Bose posed an open problem that asks to determine the time complexity of testing the point-set embeddability for planar graphs. In 2006, Cabello [9] proved the problem to be NP-complete for graphs that are 2-connected and 2-outerplanar. The problem remains NP-complete for 3-connected planar graphs [19], even when the treewidth is constant [3].

In the last few years researchers have examined the point-set embeddability problem restricted to *plane* 3*-trees* (also known as stacked polytopes, Apollonian networks, and maximal planar graphs with treewidth three) because of their wide range of applications in many theoretical and applied fields [1, 2, 14], e.g., in structure analysis of polydisperse granular packings, as a model for porous media and for the analysis of electrical supply systems. Nishat et al. [30] first gave an $O(n^2)$-time algorithm for deciding point-set embeddability of plane 3-trees, and proved an $\Omega(n \log n)$-time lower bound. Later, Durocher et al. [21] and Moosa and Rahman [29] independently improved the running time to $O(n^{4/3+\varepsilon})$, for any $\varepsilon > 0$. Since $\Omega(n^{4/3})$ is a lower bound on the worst-case time complexity for solving various geometric problems [22], it may be natural to accept the possibility that the $O(n^{4/3+\varepsilon})$-time algorithm could be asymptotically optimal. In fact, Moosa and Rahman mention that an $o(n^{4/3})$-time algorithm seems unlikely using currently known techniques. However, in this paper we prove that the $\Omega(n \log n)$ lower bound is nearly tight, giving an $O(n \log^3 n)$-time algorithm for deciding point-set embeddability of plane 3-trees.

Theorem 2.4 *Let G be a plane 3-tree with n vertices and let S be a set of n points in general position in $\mathbb{R}^2$. We can decide in $O(n \log^3 n)$ time and linear space, whether G admits a point-set embedding on S and compute such an embedding if it exists.*

Universal Point Set. Observe that a planar graph may not always admit point-set embedding on a given point set. Attempts have been made at constructing a set S of $k \geq n$ points such that every planar graph with n vertices admits a point-set embedding on a subset of S [6, 12, 18, 23, 27]. Such a point set that *supports* all planar graphs with n vertices is called a *universal point set of size n*. A long standing open question in graph drawing asks to design a set of $O(n)$ points that is universal for all planar graphs with n vertices [6]. The best known upper bound on the size of a universal point set is $O(n^2)$, which is implied by existing algorithms for drawing planar graphs on an $O(n) \times O(n)$ integer grid [13, 33]. Recently, Everett et al. [23] have designed a 1-*bend universal point set* S_n for planar graphs with n vertices, i.e., every planar graph with n vertices admits a straight-line drawing on S_n such that each vertex is mapped to a distinct point and each edge is drawn as a chain of at most two straight line segments.

The point-set embeddability problem seems to have close relation with the universal point set problem. Castañeda and Urrutia [10] proved that any set of n points in general position is universal for all outerplanar graphs with n vertices. Later, Kaufmann and Wiese [26] proved that any set S of n points is 2-*bend universal* for n (i.e., every planar graph with n vertices admits a straight-line drawing on S such that each vertex is mapped to a distinct point and each edge is drawn as a chain of at most three straight line segments, as shown in Figures 1(d)–(e)). However, the area required for the drawing could be exponential in W, where W is the length of the side of the smallest axis-parallel square that encloses S. Di Giacomo and Liotta [16, Theorem 7] showed that using the concept of monotone topological book embedding [15] one can reduce the area requirement to $O(W^3)$. Even when restricted to simpler classes of graphs (e.g., series parallel graphs or plane 3-trees), the technique of Di Giacomo and Liotta is the best known, which still requires $O(W^3)$ area. In this paper, we contribute a new technique that uses only $O(W^2)$ area to compute 2-bend point set embeddings of plane 3-trees, and hence also for partial plane 3-trees (e.g., series-parallel graphs and Halin graphs).

Theorem 3.1 *Given a plane* 3*-tree G with n vertices and a point set S of n points in general position, we can compute a* 2*-bend point-set embedding of G in $O(n \log^3 n)$ time with $O(W^2)$ area, where W is the length of the side of the smallest axis-parallel square that encloses S.*

Approximate Point-Set Embeddings. Although any set of n points in general position is universal for n-vertex outerplanar graphs [10], a plane 3-tree with n vertices may not admit a point-set embedding on a given set of n points [30]. On the other hand, while allowing two bends per edge, any set of n points in general position is 2-bend universal for plane 3-trees. Due to this apparent difficulty of defining algorithms that simultaneously minimize area, the number of bends, and running time, we consider algorithms that provide approximate solutions, that is, at least a fraction ρ of the vertices of the input graph are mapped to distinct points of the given point set. Specifically, let G be a plane graph and let Γ be a straight-line drawing of G. Let S be a set of n points in general position. Then $S(\Gamma)$ denotes the number of vertices in Γ that are mapped to distinct points of S, e.g., see Figure 1(f). The *optimal point-set embedding* of G is a straight-line drawing Γ^* such that $S(\Gamma^*) \geq S(\Gamma')$ for any straight-line drawing Γ' of G. A *ρ-approximation point-set embedding* algorithm computes a straight-line drawing Γ of G such that $S(\Gamma)/S(\Gamma^*) \geq \rho$. As opposed to embedding only a subgraph of G (which could be highly disconnected and, therefore, significantly easier to embed) the output is a straight-line planar embedding of the entire graph G for which the fraction of the vertices of G drawn on points of S is optimized. This distinction is important, since otherwise it would suffice to map the vertices of any independent set of size $n/4$ onto S to achieve a constant-factor approximation.

We show that given a plane 3-tree G with n vertices, we can construct a straight-line drawing Γ of G such that $S(\Gamma)$ is $\Omega(\sqrt{n})$, and hence the point-set embeddability is approximable with factor $\Omega(1/\sqrt{n})$ for plane 3-trees. Specifically, if the input points are in general position, then we prove that the point-set embeddability of plane 3-trees is approximable with factor $\Omega(1/\sqrt{n})$.

Theorem 4.3 *If the input points are in general position, then the point-set embeddability of plane* 3*-trees is approximable with factor $\Omega(1/\sqrt{n})$.*

2. Faster Point-Set Embeddings of Plane 3-Trees. Nishat et al. [30] gave an $O(n^2)$ algorithm for deciding point-set embeddability of plane 3-trees. Recently, Durocher et al. [21] and Moosa et al. [29] proposed different techniques to improve the running time of the algorithm of Nishat et al. to $O(n^{4/3+\varepsilon})$, for any $\varepsilon > 0$, using a triangular range search data structure. We use the basic techniques of Nishat et al.'s algorithm, but we do not use any range search data structure. Instead, we make some simple but crucial observations that help exploit a dynamic convex hull data structure to achieve even a faster algorithm. Our algorithm takes $O(n \log^3 n)$ time to decide point-set embeddability of plane 3-trees, and constructs the embedding within the same time if such an embedding exists. Before going into details, we review a few definitions.

A *plane* 3-*tree* G with $n \geq 3$ vertices is a triangulated plane graph such that if $n > 3$, then G contains a vertex whose deletion yields a plane 3-tree with $n-1$ vertices. Let r, s, t be a cycle of three vertices in G. By G_{rst} we denote the subgraph induced by r, s, t and the vertices that lie interior to the cycle. Every plane 3-tree G with $n > 3$ vertices contains a vertex that is the common neighbor of all the three outer vertices of G. We call this vertex the *representative vertex* of G. Let p be the representative vertex of G and let a, b, c be the three outer vertices of G in clockwise order. Then each of the subgraphs G_{abp}, G_{bcp} and G_{cap} is a plane 3-tree. Let S be a set of n points in the plane. Let p, q and r be three points that do not necessarily belong to S. Then $S(pqr)$ consists of the points of S that lie either on the boundary or in the interior of the triangle pqr.

Overview of Known Algorithms. Let G be a plane 3-tree with n vertices, and let a, b, c and p be the three outer vertices and the representative vertex of G, respectively. Nishat et al. [30] used the following steps to compute a point-set embedding of G on S.

Step 1. If the number of points on the boundary of the convex hull C of S is not exactly three, then G does not admit a point-set embedding on S. Otherwise, let x, y, z be the points on C.

Step 2. For each of the possible six different mappings of the outer vertices a, b, c to the points x, y, z, execute Step 3.

Step 3. Let n_1, n_2 and n_3 be the number of vertices of G_{abp}, G_{bcp} and G_{cap}, respectively. Without loss of generality assume that the current mapping of a, b and c is to x, y and z, respectively. Find the unique mapping of the representative vertex p of G to a point $w \in S$ such that the triangles xyw, yzw and zxw contain exactly n_1, n_2 and n_3 points, respectively. If no such mapping of p exists, then G does not admit a point-set embedding on S for the current mapping of a, b, c; hence go to Step 2 for the next mapping. Otherwise, recursively compute point-set embeddings of G_{abp}, G_{bcp} and G_{cap} on $S(xyw), S(yzw)$ and $S(zxw)$, respectively. See Figures 2(a)–(d).

The time complexity of any algorithm that uses Steps 1–3 is dominated by the cost of Step 3 and the bottleneck of this step is the recursive computation of the valid mappings. Observe that the recurrence relation for the time taken in Step 3 is $T(n) = T(n_1) + T(n_2) + T(n_3) + \mathcal{T}$, where $\mathcal{T}$ denotes the time required to find the mapping of the representative vertex. The algorithm of Nishat et al. [30] preprocesses the set S in $O(n^2)$ time so that the computation for the mapping of a representative vertex takes $O(n)$ time. Hence $\mathcal{T} = O(n)$ and the overall time complexity becomes $O(n^2)$. Moosa and Rahman [29] used a binary search technique

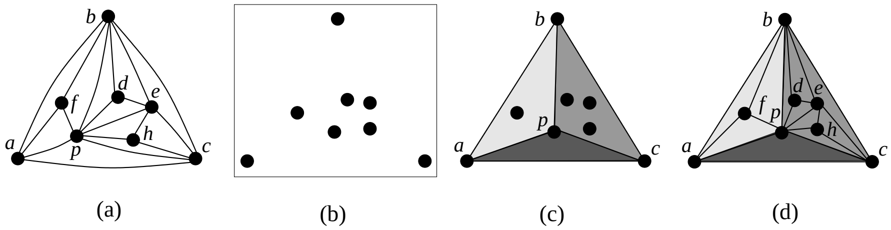

FIG. 2. *(a) A plane 3-tree G. (b) A point set S. (c)–(d) A valid mapping of the representative vertex of G, and the recursive computation of the three subproblems.*

with the help of a triangular range search data structure of Chazelle et al. [11] to obtain $\mathcal{T} = \min\{n_1, n_2, n_3\} \cdot n^{1/3+\varepsilon}$ and $T(n) = O(n^{4/3+\varepsilon})$. Durocher et al. [21] use the same idea, but instead of a binary search they use a randomized search.

Embedding Plane 3-trees in $O(n \log^3 n)$ time. We speed up the mapping of the representative vertex as follows. We first select $O(\min\{n_1, n_2, n_3\})$ points interior to the triangle xyz in $O(\min\{n_1 + n_2, n_2 + n_3, n_1 + n_3\} \log^2 n)$ time using a dynamic convex hull data structure. We prove that these are the only candidates for the mapping of the representative vertex. We then make some non-trivial observations to test and compute a mapping for the representative vertex in $O(\min\{n_1, n_2, n_3\})$ time. Hence we obtain $\mathcal{T} = O(\min\{n_1 + n_2, n_2 + n_3, n_1 + n_3\} \log^2 n)$ and a running time of $T(n) = O(n \log^3 n)$, which dominates the $O(n \log^2 n)$ time for building the initial dynamic convex hull data structure.

In the following we use three lemmas to obtain our main result. Lemma 2.1 selects a region R containing the candidate points inside the triangle xyz. Lemma 2.2 reduces the problem of finding a mapping inside the triangle xyz to the problem of finding a point satisfying specific criteria inside R. Lemma 2.3 gives an efficient technique to find such a point. Finally, we use these lemmas to obtain a mapping for the representative vertex in $O(\min\{n_1 + n_2, n_2 + n_3, n_1 + n_3\} \log^2 n)$ time.

Without loss of generality assume that $n_3 \leq n_2 \leq n_1$. Observe that $n_1 + n_2 + n_3 - 5 = n$. Let S be a set of n points in general position such that the convex hull of S contains exactly three points x, y, z on its boundary. Without loss of generality assume that the vertices outer vertices a, b, c are mapped to the points x, y, z, respectively.

Let u and v be two points on the straight line segment xz such that $|S(uxy)| = n_1 - 1$ and $|S(vzy)| = n_2 - 1$, as shown in Figure 3(a). Note that none of u and v belong to S. It is straightforward to verify that if a *valid mapping* for the representative vertex exists (i.e, there exists a point $w \in S$ such that $|S(wxy)| = n_1$, $|S(wyz)| = n_2$ and $|S(wzx)| = n_3$), then the corresponding point (i.e., the point w) must lie inside $S(uvy)$. Let r and s be two points on the straight line segments uy and vy, respectively, such that $|S(rux)| = |S(svz)| = n_3 - 1$. We call the region defined by the simple polygon x, u, v, z, s, y, r, x the *region of interest*. An example is shown in Figures 3(b). The following lemma shows that the region of interest must contain the edges corresponding to a valid mapping.

LEMMA 2.1. *If there exists a point $w \in S$ that corresponds to a valid mapping for the representative vertex of G, then the straight line segments wx, wy and wz lie inside the region of interest R. Moreover, the number of points in R that belong to S is $O(n_3)$, and the following properties hold.*

(*a*) *If the points s, y, z (respectively, points r, x, y) are distinct, then $|S(syz)| = n_2 - n_3 + 2$ (respectively, $|S(rxy)| = n_1 - n_3 + 2$).*

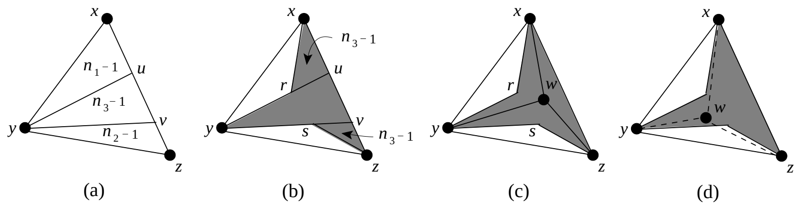

FIG. 3. *(a)–(b) Illustration for the lines uy, vy, xr and zs. The region of interest is shown in gray. (c)–(d) Illustration for the proof of Lemma 2.1.*

(*b*) *Otherwise, point s (respectively, point r) coincides with y (respectively, y) and $|S(syz)| = 2$ (respectively, $|S(rxy)| = 2$).*

Proof. The point w must be in $S(uvy)$. Otherwise, either $|S(wxy)| < n_1$ or $|S(wyz)| < n_2$ holds, which implies that w does not correspond to a valid mapping. We now claim that the straight line segments wx, wy and wz lie interior to R, as shown in Figure 3(c). Since $w \in S(uvy)$, the straight line segment wy must lie inside R. Suppose for a contradiction that either one of wx, wz or both properly crosses the boundary of R. In this situation $S(wxz)$ must contain one of $S(rux)$ and $S(svz)$ implying that $|S(wxz)| > n_3$, as shown in Figure 3(d). Consequently, wx, wy and wz must lie interior to R. By the construction of R, the number of points that lie on the boundary and the interior of R is at most $(n_3 - 1) + 2(n_3 - 1) = O(n_3)$.

We now determine the value of $|S(syz)|$. If the points s, y, z are distinct, then the triangle syz contains $(n_2 - 1) - (n_3 - 1) - 1$ points of S in its proper interior and three points (i.e., y, z and the other point on line sz) of S on its boundary. Therefore, $|S(syz)| = n_2 - n_3 + 2$. Otherwise, if s coincides with y, then $n_3 = n_2$ and $S(syz)$ consists of exactly two points of S, i.e., y and z. Since y and z are distinct, we are only left with the case when s coincides with z. But this case does not arise since s is a point of the segment yv, and v does not coincide with z since $n \geq 4$ and $n_3 \geq 3$. We can determine the value of $|S(rxy)|$ in a similar way. □

Let $S' \subseteq S$ be the set that consists of the points lying on the boundary of R and the points lying in the proper interior of R. We call S' the *set of interest*. By Lemma 2.1, $|S'| = O(n_3)$. We reduce the problem of finding a valid mapping in S to the problem of finding a point with certain properties in S', as shown in the following lemma.

LEMMA 2.2. *There exists a valid mapping for the representative vertex of G in S if and only if there exists a point $w' \in S'$ such that $|S'(w'yz)| = n_2 - |S(yzs)| + 3$, $|S'(w'xy)| = n_1 - |S(xyr)| + 3$ and $|S'(w'xz)| = n_3$.*

Proof. Assume that there exists a valid mapping for the representative vertex of G in S and the point corresponding to the valid mapping is w. By Lemma 2.1, $w \in S'$. We now prove that if we choose $w = w'$, then $|S'(w'yz)|$, $|S'(w'xy)|$ and $|S'(w'xz)|$ must have the required number of points.

Observe that the number of points in the proper interior of triangle yzs is $|S(yzs)| - 3$. All the points on the boundary of yzs are also on the boundary of R. Since w corresponds to a valid mapping in S, $|S(wyz)| = n_2$. Consequently, $|S'(w'yz)| = |S(wyz)| - |S(yzs)| + 3 = n_2 - |S(yzs)| + 3$. We can prove that $|S'(w'xy)| = n_1 - |S(xyr)| + 3$ and $|S'(w'xz)| = n_3$ in a similar way.

Assume now that there exists a point $w' \in S'$ such that $|S'(w'yz)| = n_2 - |S(yzs)| + 3$, $|S'(w'xy)| = n_1 - |S(xyr)| + 3$ and $|S'(w'xz)| = n_3$. We now prove that $|S(wyz)| = n_2$. Since the number of points in the proper interior of triangle yzs is

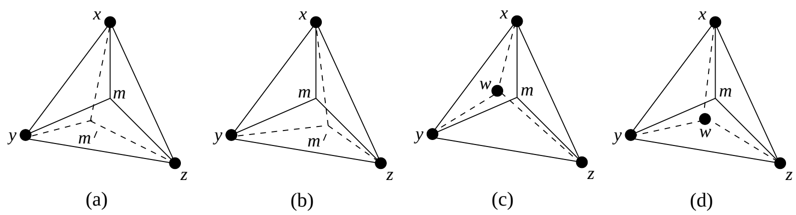

FIG. 4. *Illustration for the proof of Lemma 2.3, where* $\{m, m'\} \cap S = \emptyset$ *and* $\{x, y, z, w\} \subset S$.

$|S(yzs)| - 3$, $|S(wyz)| = |S'(w'yz)| + |S(yzs)| - 3 = n_2$. Similarly, we can verify that $|S(wxy)| = n_1$ and $|S(wyz)| = n_3$. ∎

We can use the point w' to define a partition of the set S' into three subsets $S'(w'yz)$, $S'(w'xy)$ and $S'(w'xz)$, where the values $|S'(w'yz)|$, $|S'(w'xy)|$ and $|S'(w'xz)|$ are fixed. In other words, w' acts as a valid mapping for S' with respect to the values $|S'(w'yz)|$, $|S'(w'xy)|$ and $|S'(w'xz)|$. Since a valid mapping is unique [30], the point w' must also be unique.

We call the point w' the *principal point* of S'. Observe that this principal point corresponds to the valid mapping of the representative vertex of G in S. We will use the following lemma to efficiently find a valid mapping.

LEMMA 2.3. *Let S be a set of $t \geq 4$ points in general position such that the convex hull of S is a triangle xyz. Let i, j, k be three non-negative integers, where $i \geq 3, j \geq 3$ and $k = t + 5 - i - j$. Then we can decide in $O(t)$ time whether there exists a point $w \in S$ such that $|S(wxy)| = i, |S(wyz)| = j$ and $|S(wxz)| = k$, and compute such a point if it exists.*

Proof. Consider first a variation of the problem, where we want to construct a point $m \notin S$ interior to xyz such that $|S(mxy)| = i + 1, |S(myz)| = j - 1$ and $|S(mxz)| = k - 1$. Steiger and Streinu [35] proved the existence of m and gave an $O(t)$-time algorithm to find m. If there exists a point $w \in S$ such that $|S(wxy)| = i, |S(wyz)| = j$ and $|S(wxz)| = k$, then it is straightforward to observe that there exists a point $m \notin S$ interior to xyz such that $|S(mxy)| = i+1, |S(myz)| = j-1$ and $|S(mxz)| = k - 1$. We now prove that the existence of m implies a unique partition of S. Hence we can efficiently test whether w exists.

We claim that if there exists a point $m' \neq m$, where $m' \notin S$, such that $|S(m'xy)| = i + 1, |S(m'yz)| = j - 1$ and $|S(m'xz)| = k - 1$, then the sets $S(m'xy), S(m'yz)$ and $S(m'xz)$ must coincide with the sets $S(mxy), S(myz)$ and $S(mxz)$. To verify the claim assume without loss of generality that $m' \in S(myz)$. Since the triangle $m'yz$ lies interior to the triangle myz, the sets $S(m'yz)$ and $S(myz)$ must be identical. On the other hand, either the triangle mxz lies interior to the triangle $m'xz$, or the triangle mxy lies interior to the triangle $m'xy$, as shown in Figures 4(a)–(b). Therefore, either the sets $S(mxz)$ and $S(m'xz)$, or the sets $S(mxy)$ and $S(m'xy)$ must be identical. Consequently, the remaining pair of sets must also be identical.

Observe that if the point $w \in S$ we are looking for exists, then w must lie interior to $S(mxy)$, as shown in Figure 4(c). Otherwise, if $w \in S(myz)$ (respectively, $w \in S(mxz)$), then $|S(myz)| \geq |S(wyz)| = j$ (respectively, $|S(mxz)| \geq |S(wxz)| = k$), which contradicts our initial assumption that $|S(myz)| = j - 1$ (respectively, $|S(mxz)| = k - 1$). Figure 4(d) depicts such a scenario. If w exists, then the convex hull of $S(mxy)$ must be a triangle xym'', where $m'' \in S(mxy)$. If $|S(m''xy)| = i, |S(m''yz)| = j$ and $|S(m''xz)| = k$, then m'' is the required point w.

Otherwise, no such w exists.

We can test whether the convex hull of $S(mxy)$ is a triangle in $O(t)$ time (e.g., find the leftmost point a, the rightmost point b and the point c with the largest perpendicular distance to the line determined by the line segment ab, and then test whether triangle abc contains all the points). It is also straightforward to compute the values $|S(m''xy)|, |S(m''yz)|$ and $|S(m''xz)|$ in $O(t)$ time. □

Given the set of interest $S' \subseteq S$, we use Lemmas 2.2 and 2.3 to find the principal point $w' \in S'$ in $O(n_3)$ time. Observe that this principal point corresponds to the valid mapping of the representative vertex of G in S. We now show how to compute the set S' in $O((n_2 + n_3)\log^2 n)$ time using the dynamic planar convex hull data structure of Overmars and van Leeuwen [31], which supports a single update (i.e., a single insertion or deletion) in $O(\log^2 n)$ time. We refer the reader to Figures 3(a)–(b) to recall the definition of the region of interest.

Step A. Assume that the points of S are placed in a dynamic convex hull data structure $\mathcal{D}$. We repeatedly delete the neighbor of y on the boundary of the convex hull of S starting from z in anticlockwise order. After deleting $n_2 - 2$ points, we insert all the deleted points into a new dynamic convex hull data structure $\mathcal{D}'$. We then insert a copy of y into $\mathcal{D}'$. Recall u and v from Figure 3(b). Observe that all the points of $S(vyz)$ are placed in $\mathcal{D}'$. In a similar way we construct another dynamic convex hull data structure $\mathcal{D}''$ that maintains all the points of $S(uvy)$. Consequently, $\mathcal{D}$ now only maintains the points of $S(uxy)$. Since a single insertion or deletion takes $O(\log^2 n)$ time, all the above $O(n_2 + n_3)$ insertions and deletions take $O((n_2 + n_3)\log^2 n)$ time in total.

Step B. We now construct two other dynamic convex hull data structures $\mathcal{D}_1$ and $\mathcal{D}_2$ using $\mathcal{D}$ and $\mathcal{D}'$ such that they maintain the points of $S(rux)$ and $S(svz)$, respectively. Since $|S(rux)| + |S(svz)| = O(n_3)$, this takes $O(n_3 \log^2 n)$ time.

Step C. We construct the point set S' using the points maintained in $\mathcal{D}'', \mathcal{D}_1$ and $\mathcal{D}_2$, which also takes $O(n_3 \log^2 n)$ time. In similar way we can restore the original point set S and the initial data structure $\mathcal{D}$ in $O((n_2 + n_3)\log^2 n)$ time.

The time for the construction of S' using Steps A–C is $O((n_2 + n_3)\log^2 n)$, which dominates the time required for the computation of the valid mapping of the representative vertex p. Let w be the point that corresponds to the valid mapping. We now need to construct the point sets $S(wxy), S(wyz)$ and $S(wzx)$ for recursively testing the point-set embeddability of G_{abp}, G_{bcp} and G_{cap}, respectively. We can construct $S(wxy), S(wyz)$ and $S(wzx)$ and their corresponding dynamic convex hull data structures in $O((n_2 + n_3)\log^2 n)$ time as follows. Let l be the point of intersection of the infinite straight lines determined by the line segments wy and xz. First construct the set $S(lyz)$ and then modify it to obtain the sets $S(wyz)$ and $S(lwz)$, which takes $O((n_2 + n_3)\log^2 n)$ time. Now modify the set $S(lxy)$ to construct the set $S(lwx)$, and then use the sets $S(lwx)$ and $S(lwz)$ to construct $S(wxz)$, which takes $O(n_3 \log^2 n)$ time. Observe that after the modification of the set $S(lxy)$, we are left with the set $S(wxy)$.

We now show that the total time taken is $T(n) \leq dn\log^3 n$, for some constant d, as follows. There exists $c > 0$ such that for all $d \geq c$,

$$
\begin{aligned}
T(n) &= T(n_1) + T(n_2) + T(n_3) + O((n_2+n_3)\log^2 n) \\
&\le dn_1\log^3 n_1 + dn_2\log^3 n_2 + dn_3\log^3 n_3 + c(n_2+n_3)\log^2 n \\
&\le dn_1\log^3 n + dn_2\log^2 n\log\tfrac{n}{2} + dn_3\log^2 n\log\tfrac{n}{2} + c(n_2+n_3)\log^2 n \\
&= dn_1\log^3 n + dn_2\log^2 n(\log n - 1) + dn_3\log^2 n(\log n - 1) + c(n_2+n_3)\log^2 n \\
&= d(n_1+n_2+n_3)\log^3 n - (d-c)(n_2+n_3)\log^2 n \\
&\le dn\log^3 n.
\end{aligned}
$$

Observe that the construction of the initial data structure $\mathcal{D}$ takes $O(n\log^2 n)$ time, which is dominated by $T(n)$. The dynamic planar convex hull of Brodal and Jacob [8] takes amortized $O(\log n)$ time per update. Therefore, using their data structure instead of Overmars and van Leeuwen's data structure [31] we can improve the expected running time of our algorithm. Since the algorithms of [31, 35] take linear space, the space complexity of our algorithm is $O(n)$.

THEOREM 2.4. *Given a plane 3-tree G with n vertices and a set S of n points in general position in $\mathbb{R}^2$, we can decide the point-set embeddability of G on S in $O(n\log^3 n)$ time and $O(n)$ space, and compute such an embedding if it exists.*

Our algorithm could be easily adapted to the case when the input points are not in general position, provided that the algorithms of Overmars and van Leeuwen [31] and Steiger and Streinu [35] can handle degenerate cases.

3. Universal Point Set for Plane 3-Trees. In this section we give an algorithm to compute 2-bend point-set embeddings of plane 3-trees on a set of n points in general position in $O(W^2)$ area, where W is the length of the side of the smallest axis-parallel square that encloses S.

We describe an outline of the algorithm. Given a plane 3-tree G and a set of points S in general position, we first construct a straight-line drawing Γ of G such that every point of S other than a pair of points on the convex hull of S lies in the proper interior of some distinct inner face in Γ, as shown in Figure 5(c). While constructing Γ, we compute a bijective function ϕ from the vertices of Γ to the points of S. We then extend each edge (u,v) in Γ using two bends to place the vertices u and v onto the points $\phi(u)$ and $\phi(v)$, respectively, as shown in Figure 5(d). We prove that Γ and ϕ maintain certain properties so that the resulting drawing Γ' remains planar.

In the following we describe the algorithm in detail. Let H be the convex hull of S. Construct a triangle xyz with $O(W^2)$ area such that xyz encloses H and the side yz passes through a pair of consecutive points y', z' on the boundary of H. Assume that y' is closer to y than z'. Set $\phi(y) = y'$ and $\phi(z) = z'$. Set $\phi(x) = x'$, where x' is the point on the convex hull of $S(xyz)$ for which the angle $\angle xyx'$ is smallest. Figure 5(e) illustrates the triangle xyz and the function ϕ. We call the straight line segments $x\phi(x), y\phi(y), z\phi(z)$ the *wings* of xyz. Observe that only $x\phi(x)$ among the three wings of xyz lie in the proper interior of xyz. We use this invariant throughout the algorithm, i.e., every face f in the drawing will contain at most one wing that is in the proper interior of f. We call such a wing the *major wing* of f.

Let a, b, c be the outer vertices of G in anticlockwise order and let p be the representative vertex of G. Map the vertices a, b, c to the points x, y, z. Let $S \setminus \{x', y', z'\}$ be the point set S'. Let $\hat{n}_1, \hat{n}_2$ and $\hat{n}_3$ be the number of inner vertices of G_{abp}, G_{bcp} and G_{cap}, respectively. Since the major wing of xyz is incident to x, we compute a point $w \notin S$ such that $S'(wxy) = \hat{n}_1$, $S'(wyz) = \hat{n}_2 + 1$ and $S'(wxz) = \hat{n}_3$, as shown in Figure 5(e). Steiger and Streinu [35] proved that such a point always exists and gave an $O(|S'|)$-time algorithm to find w. Since the angle $\angle xy\phi(x)$ is the smallest, if wy or wz intersects $x\phi(x)$, then by continuity there must exist another point $\bar{w}$ on the line wz such that $S'(\bar{w}xy) = \hat{n}_1, S'(\bar{w}yz) = \hat{n}_2 + 1$, $S'(\bar{w}xz) = \hat{n}_3$ hold, and we choose $\bar{w}$ as the point w. Figures 5(f)–(g) depict such scenarios. Note that $x\phi(x)$ now lies either in the triangle wxy or wxz. Set $\phi(w) = w'$, where w' is the point on the convex hull of $S'(wyz)$ for which the angle $\angle wyw'$ is smallest. Since wyz does not contain $x\phi(x)$, the mapping we compute maintains the invariant that every face contains at most one major wing.

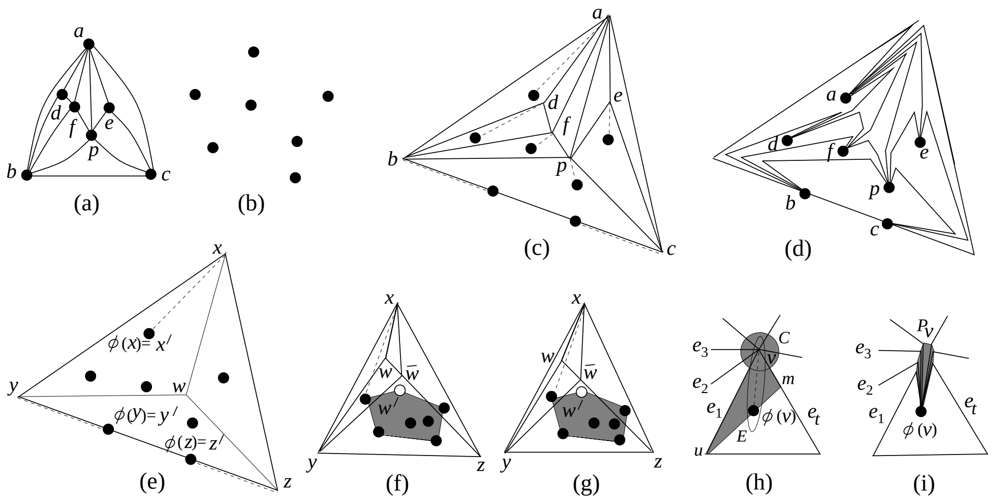

Fig. 5. *(a) A plane 3-tree G. (b) A set of points S. (c) Γ and φ, where φ is illustrated with dashed lines. (d) A 2-bend point-set embedding of G on S. (e) Illustration for the triangle xyz. (f)–(g) Construction of w and φ(w), where φ(w) = w′ is shown in white and the convex hull of S(xyz) is shown in gray. (h) The region R and ellipse E, where R is shown in gray. (i) Illustration for P_v.*

We now recursively construct the drawings of G_{abp}, G_{bcp} and G_{cap} with the point sets $S'(xyw)$, $S'(yzw) \setminus w'$ and $S'(zxw)$, respectively. Note that while recursively constructing a point w for the representative vertex inside some triangle xyz, then the triangle may not have any major wing. Also in this case, it suffices to compute w such that $S'(wxy) = \hat{n}_1, S'(wyz) = \hat{n}_2 + 1$ and $S'(wxz) = \hat{n}_3$ holds. Once we complete the recursive computation, we obtain a straight-line drawing Γ of G, and a bijective function ϕ from the vertices of Γ to the points of S. The idea now is to extend each edge (u, v) in Γ using two bends to place the vertices u and v onto the points $\phi(u)$ and $\phi(v)$, respectively. We use ϕ and the property that every face in Γ contains at most one major wing, to maintain planarity. We now describe the construction details.

For each vertex v in Γ we do the following. Let $e_1, e_2, \ldots, e_t$ be the edges adjacent to v in clockwise order such that e_1 and e_t form the smallest angle that contains $\phi(v)$. Construct a strictly convex polygon $P_v = \phi(v), v_1, v_2, \ldots, v_t$, where $v_i, 1 \leq i \leq t$, is a point on e_i and no point of S other than $\phi(v)$ lie inside the polygon. Now delete the straight line segments vv_i and draw the segments $\phi(v)v_i$, as shown in Figure 5(i). Since every face in Γ contains at most one major wing, we can construct the P_vs such that for two different vertices v_1 and v_2 in Γ, the corresponding convex polygons P_{v_1} and P_{v_2} are disjoint. Later in this section, we precisely describe such a construction for P_vs.

We claim that the resulting drawing Γ' is a 2-bend point-set embedding of G on S. Since every edge in Γ has only two endpoints, the corresponding edge e in Γ' has exactly two bends. Since ϕ is a bijective function, e does not create any loop in Γ'. It now suffices to prove that Γ' is a planar drawing of G. Observe that Γ is a planar straight-line drawing. Therefore, if Γ' is not a planar drawing, then either two of the newly added segments properly intersect (Case 1), or a newly added segment intersects an old segment that originally belongs to Γ (Case 2). Case 1 cannot happen since all P_vs are disjoint. Case 2 cannot appear since every newly added segment lie in some convex polygon P_v that does not contain any old segment in Γ'.

We can construct Γ in $O(n \log^3 n)$ time with a similar technique as in Section 2. We now describe how to construct Γ' from Γ in $O(n \log n)$ time. Let $\mathcal{S}$ be a set that consists of the points corresponding to the vertices in Γ and the points that belong to S. Let η be the Euclidean distance between the closest pair of points in $\mathcal{S}$. We can compute η in $O(n \log n)$ time [34]. For each vertex v in Γ, we now construct the convex polygon P_v in $O(\deg(v))$ time as follows.

Let C be the circle centered at v with radius $\eta/2$. Assume that $e_1 = (u, v)$, and m is the intersection point of edge e_t and the line determined by $u, \phi(v)$. Let R be the region determined by the union of C and the triangle uvm. Consider now an ellipse E with foci v and $\phi(v)$ such that the half of the ellipse (determined by the minor axis of E) that contains v lies interior to R. We can always find such an ellipse since a straight line segment can be viewed as a degenerate case of an ellipse. We now define $v_1, v_2, \ldots, v_t$ as the intersection points of E with $e_1, e_2, \ldots, e_t$, respectively, as shown in Figures 5(h)–(i). Since R does not contain any point of S other than $\phi(v)$, the polygon $P_v = \phi(v), v_1, v_2, \ldots, v_t$ does not contain any point of S other than $\phi(v)$. Since every face in Γ contains at most one major wing and C is a circle with radius $\eta/2$, any two P_vs must be disjoint.

Consequently, the algorithm takes $O(n \log^3 n) + O(n \log n) + \sum_{\forall v} O(\deg(v)) = O(n \log^3 n)$ time. The following theorem summarizes the results of this section.

Theorem 3.1. *Given a plane 3-tree G with n vertices and a point set S of n points in general position, we can compute a 2-bend point-set embedding of G in $O(n \log^3 n)$ time with $O(W^2)$ area, where W is the length of the side of the smallest axis-parallel square that encloses S.*

4. Approximate Point-Set Embeddings. Let Γ be a straight-line drawing of G. Then $S(\Gamma)$ denotes the number of vertices in Γ that are mapped to distinct points of S. The *optimal point-set embedding* of G is a straight-line drawing Γ^* such that $S(\Gamma^*) \geq S(\Gamma')$ for any straight-line drawing Γ' of G. A ρ-approximation point-set embedding algorithm computes a straight-line drawing Γ of G such that $S(\Gamma)/S(\Gamma^*) \geq \rho$. In this section we show that given a plane 3-tree G with n vertices, we can construct a straight-line drawing Γ of G such that $S(\Gamma) = \Omega(\sqrt{n})$, and hence point-set embeddability is approximable with factor $\Omega(1/\sqrt{n})$ for plane 3-trees.

We first introduce a few more definitions. Let G be a plane 3-tree with the outer vertices a, b, c and representative vertex p, and let the number of vertices of G be n. Then the *representative tree* T_{n-3} of G satisfies the following conditions [30].

(a) If $n = 3$, then T_{n-3} is empty.
(b) If $n = 4$, then T_{n-3} consists of a single vertex.
(c) If $n > 4$, then the root p of T_{n-3} is the representative vertex of G and the subtrees rooted at the three counter-clockwise ordered children p_1, p_2 and p_3 of p in T_{n-3} are the representative trees of G_{abp}, G_{bcp} and G_{cap}, respectively.

Since a rooted tree with n nodes is a partially ordered set under the 'successor' relation, by Dilworth's theorem [17], either the height or the number of leaves in the tree is at least $\sqrt{n}$. Let G be the input plane 3-tree with n vertices and let T be its representative tree with $n-3$ vertices [30].

If T has $\Omega(\sqrt{n})$ leaves, then we use the technique of Theorem 3.1 to have a straight-line drawing Γ of G such that $S(\Gamma) = \Omega(\sqrt{n})$ as follows. Find a straight-line drawing of G and the bijective function ϕ, as shown in Figure 5(c). Observe that the leaves of T (i.e., the inner vertices of degree three in G) correspond to distinct K_4s in the drawing, and hence we can place each leaf l to the point $\phi(l)$ avoiding any edge crossing. Otherwise, the height of T is $\Omega(\sqrt{n})$. In this case we prove that G has a 'canonical ordering tree' (also, called Schnyder's realizer [33]) with height $\Omega(\sqrt{n})$, as shown in Lemma 4.1. We then show (in Lemma 4.2) a simple way to compute a straight-line drawing Γ of G such that $S(\Gamma) = \Omega(\sqrt{n})$.

Before proving Lemmas 4.1 and 4.2 we recall the definition of canonical ordering. Let G be a triangulated plane graph with the outer vertices x, y and z in clockwise order on the outer face. Let $\pi = (u_1(= x), u_2(= z), ..., u_n(= y))$ be an ordering of all vertices of G. By G_k, $3 \le k \le n$, we denote the subgraph of G induced by $\{u_1, u_2, ..., u_k\}$ and by C_k the outer cycle (i.e., the boundary of the outer face) of G_k. We call π a *canonical ordering* of G with respect to the outer edge (x, z) if for each index k, $3 \le k \le n$, the following conditions are satisfied [13].

(a) G_k is 2-connected and internally triangulated.
(b) If $k + 1 \le n$, then u_{k+1} is an outer vertex of G_{k+1} and the neighbors of u_{k+1} in G_k appears consecutively on C_k.

Assume that for some $k \ge 3$, the outer cycle C_k is $w_1(= x), \ldots, w_p, w_q(= u_k), w_r \ldots, w_t(= z)$, where the vertices appear in clockwise order on C_k. Then we call the edges (w_p, u_k) and (u_k, w_r) the *left-edge* and the *right-edge* of u_k, respectively. Let E^* be the set of edges that does not belong to any C_k, $3 \le k \le n$. Then the graph induced by the edges in E^* is a tree. The graph induced by the right-edges (respectively, left-edges) of the vertices $u_k, 3 \le k \le n-1$, is also a tree. These three trees form the *Schnyder's realizer* of G, and each of them is known as a *canonical ordering tree* of G.

Lemma 4.1. *Let G be a plane 3-tree and let T be its representative tree. If the height of the representative tree is $\Omega(\sqrt{n})$, then G has a canonical ordering tree with height $\Omega(\sqrt{n})$.*

Proof. Let $P = (v_1, v_2, \ldots, v_k)$, $k = \Omega(\sqrt{n})$, be the longest path from the root v_1 of T to some leaf v_k. Without loss of generality assume that k is even. By G_i, where $1 \le i \le k$, we denote the plane 3-tree induced by the outer vertices of G and the vertices $v_1, v_2, \ldots, v_i$. We now incrementally construct G_k. First construct a triangle xyz, place the vertex v_1 interior to xyz and add the segments v_1x, v_1y, v_1z. Since v_2 is a child of v_1, v_2 must be placed interior to one of the triangles incident to v_1. Since v_{i+1}, where $i + 1 \le k$, is a child of v_i, this condition holds throughout the construction. Let T_x, T_y, T_z be the trees of the Schnyder's realizer rooted at x, y, z, respectively. Figure 6(a) illustrates the realizer of G_2, where the heights of T_x, T_y and T_z are one, one and two, respectively. By A and B we denote the rooted trees isomorphic to T_x and T_z in G_2, respectively. The nodes of $T_w, w \in \{x, y, z\}$, where the realizer grows while adding v_{i+1} to G_i, $i \ge 2$, are called the *connectors* of T_w in G_i. Figures 6(e)–(g) illustrate all possible ways to insert v_3, and show the vertices where the realizer T_y can grow in gray. Here the realizer grows only at the vertices y and v_2, and hence these are the connectors for T_y while adding v_3 to G_2. Figure 6(b) illustrates the connectors for every T_w in gray.

Consider now the steps when we obtain the graphs $G_2, G_4, G_6, \ldots, G_k$. Observe that each time some tree of the form A (or B) gets connected with some $T_w, w \in \{x, y, z\}$, of G_i, the connectors of A (or B) become the only connectors of T_w in G_{i+2}. We describe this scenario with an example of G_4, as shown in Figure 6(c). Figure 6(d) shows the connectors of T_x, T_y and T_z in gray. The tree T_x consists of two trees: one is of the form A and the other is of the form B, where the root of B coincides with some connector of A (i.e., v_2). Since the subsequent vertex v_5 must lie inside the triangle yv_1v_3, the connectors of B become the only connectors of T_x in G_4. Similarly, we can verify this condition for T_y and T_z.

Consequently, each time some tree of the form B gets connected with some $T_w, w \in \{x, y, z\}$, of G_i, the height of T_w increases by one in G_{i+2}. Since we need $k/2$ steps before we obtain G_k, one of T_x, T_y or T_z must have height at least $k/6 = \Omega(\sqrt{n})$. Since each tree of the Schnyder's realizer of G_k is a subtree of a distinct tree of the Schnyder's realizer of G, the proof is complete. □

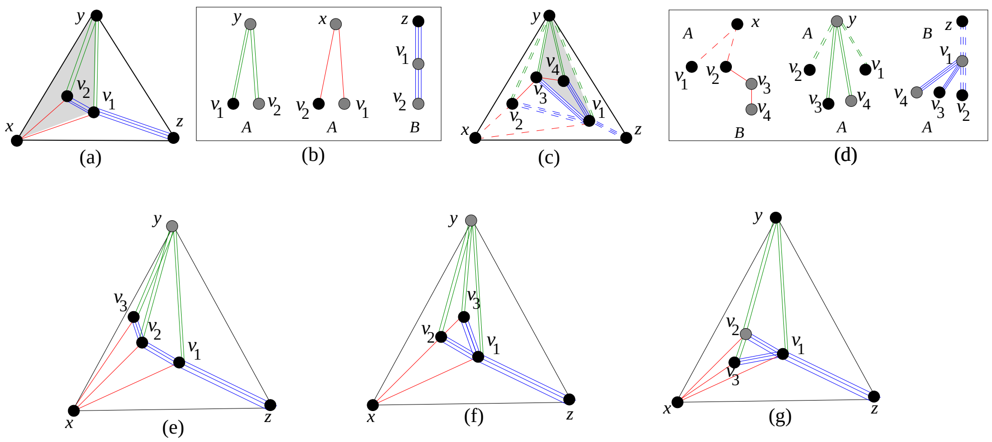

FIG. 6. *(a) Illustration for G_2, where each edge of T_x, T_y and T_z is shown in one, two and three parallel lines, respectively. (b) Illustration for the connectors, shown in gray. (c)–(d) Example of a connection of A, A, B with B, A, A, respectively. (e)–(g) Illustration for the connectors of T_y by considering the possible insertions of v_3 in G_2.*

LEMMA 4.2. *Let G be a plane 3-tree with n vertices and let S be a set of n or more points in general position. If G contains a canonical ordering tree T with height k, then G admits a straight-line drawing Γ such that $S(\Gamma) = k$.*

Proof. We prove the lemma using an induction on the number of vertices n of G. The case when $n \leq 4$ is straightforward. We assume that the lemma holds for all plane 3-trees with fewer than n vertices, and now consider the case when G has n vertices.

Let $P = (v_1, v_2, \ldots, v_k)$, be the longest path from the root v_1 of T to some leaf v_k. Let $p_1, p_2, \ldots, p_k$ be k points of S in decreasing order of their y-coordinates. Let C be a polygonal chain that consists of the straight-line segments p_i, p_{i+1}, $1 \leq i \leq k-1$.

We now construct a triangle xyz such that $x = p_1$, and all the points of the chain C are visible to both y and z, i.e., the straight-lines from the points of C to the point y or z do not cross any segment of C, as illustrated in Figure 7. Let a, b, c be the outervertices of G and let p be the representative vertex of G. Without loss of generality assume that T is rooted at b.

If the edge (b, p) is not contained in the path P, then without loss of generality assume that (b, p) is an inner edge of G_{bcp}. We now find a point w interior to the triangle xyz such that the distance between w and y is very small and all the points of C are visible to w. We then construct a straight-line drawing of G_{abp} and G_{cap} interior to the triangles xyw and yzw. By induction, G_{bcp} admits a straight-line drawing Γ' interior to zxw such that $S(\Gamma') = k$. See Figures 7(b)–(c).

Otherwise, the path P contains the edge (b, p). In this case we choose $w = p_2$. We then construct a straight-line drawing of G_{abp} and G_{bcp} interior to the triangles xyw and zxw. By induction, G_{cap} admits a straight-line drawing Γ' interior to yzw such that $S(\Gamma') = k - 1$. See Figures 7(d)–(e). ☐

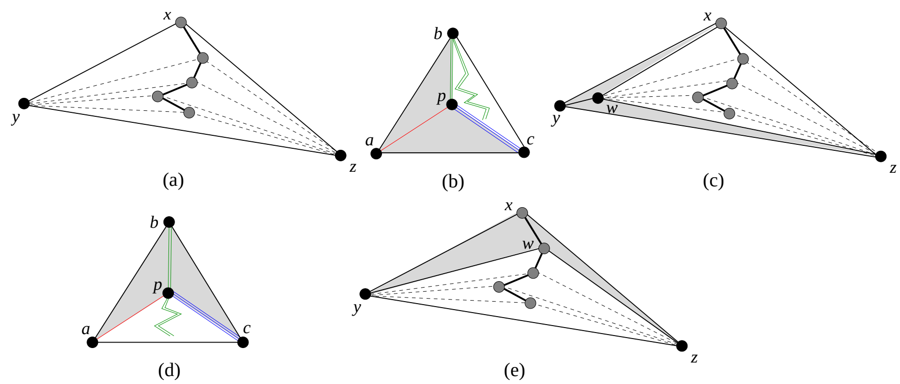

Fig. 7. *Illustration for the proof of Lemma 4.2. The points of S are shown in gray.*

The following theorem summarizes the result of this section.

Theorem 4.3. *Given a plane 3-tree G with n vertices and a point set S of n points in general position in $\mathbb{R}^2$, we can compute a straight-line drawing Γ of G in polynomial time such that the number of vertices in Γ that are mapped to distinct points of S is $1/(6\sqrt{n})$ times to the optimal. Hence the point-set embeddability of plane 3-trees is approximable with factor $\Omega(1/\sqrt{n})$.*

Observe that we can use the technique of Theorem 3.1 to have a 1-bend straight-line drawing of G that uses $n/4$ distinct points of S (e.g., choose an independent set of $n/4$ vertices and place those vertices on $n/4$ distinct points of S determined by ϕ using at most one bend per edge). Consequently, 1-bend point-set embeddability is approximable with at least factor 0.25 for plane 3-trees.

5. Conclusion. Using techniques that are completely different from those used in the previously best known approaches for testing point-set embeddability of plane 3-trees (achieving $O(n^{4/3+\varepsilon})$ time and $O(n^{4/3})$ space), in Section 2 we described an algorithm that solves the problem for a given plane 3-tree in $O(n \log^3 n)$ time using $O(n)$ space. As suggested by an anonymous reviewer, one possibility for potentially reducing the running time further might be to apply the algorithm of Moosa and Rahman [29], where an orthogonal range search would be used instead of a triangular range search. Specifically, given points x and y and an integer k, a triangle wxy that contains k points can be found by encoding each point w using two values: the slopes of wx and wy. The triangle wxy is then mapped to a two-sided axis-aligned orthogonal range query. It is not obvious, however, how this technique would be applied in recursive levels. One possibility might be to use a dynamic orthogonal range counting data structure. A natural optimization question in this direction is as follows.

Open Question 1. *Given a plane* 3*-tree* G *with* n *vertices and a set* S *of* n *points in general position, how fast can we compute a straight-line embedding of* G *such that the number of vertices placed on the points of* S *is maximized?*

In Section 3 we proved that every plane 3-tree admits a 2-bend point-set embedding on any set of n points in general position in $O(W^2)$ area. An important issue here is to examine the amount of scale up required to ensure that the vertices and bend points of the drawings produced in Section 3 lie on integer coordinates, i.e., the area requirement under minimum resolution assumption. While our result holds for partial plane 3-trees, one may try to characterize the graphs that admit 2-bend point-set embeddings in small area.

Open Question 2. *Characterize the planar graphs with* n *vertices that admit* 2*-bend point-set embeddings on any set of* n *points in general position in* $O(W^2)$ *area, where* W *is the length of the side of the smallest axis parallel square that encloses the given point set.*

In Section 4 we proved that the point-set embeddability problem (respectively, the 1-bend point-set embeddability problem) is approximable with factor at least $\Omega(1/\sqrt{n})$ (respectively, 0.25) for plane 3-trees within $O(n \log^3 n)$ time, which motivates us to ask the following question.

Open Question 3. *Design an* $o(n^2)$*-time algorithm that can approximate point-set embeddability for plane* 3*-trees with a constant factor, or prove that no such algorithm exists.*

Acknowledgement. We thank Valentin Polishchuk and the other anonymous reviewers for many constructive and helpful comments.

REFERENCES

[1] José S. Andrade, Jr., Hans J. Herrmann, Roberto F. S. Andrade, and Luciano R. da Silva, *Apollonian networks: Simultaneously scale-free, small world, euclidean, space filling, and with matching graphs*, Physical Review Letters, 94 (2005).

[2] Reza Mahmoodi Baram, *Polydisperse Granular Packings and Bearings*, PhD thesis, University of Stuttgart, Germany, 2005.

[3] Therese Biedl and Martin Vatshelle, *The point-set embeddability problem for plane graphs*, in Proc. of the 28th Annual Symposium on Computational geometry (SoCG), ACM, 2012, pp. 41–50.

[4] Prosenjit Bose, *On embedding an outer-planar graph in a point set*, Comput. Geom., 23 (2002), pp. 303–312.

[5] Prosenjit Bose, Michael McAllister, and Jack Snoeyink, *Optimal algorithms to embed trees in a point set*, J. Graph Algorithms Appl., 1 (1997), pp. 1–15.

[6] Franz J. Brandenburg, David Eppstein, Michael T. Goodrich, Stephen G. Kobourov, Giuseppe Liotta, and Petra Mutzel, *Selected open problems in graph drawing*, in Proc. of the 11th International Symposium on Graph Drawing (GD), vol. 2912 of LNCS, Springer, 2004, pp. 515–539.

[7] Peter Brass, Eowyn Cenek, Christian A. Duncan, Alon Efrat, Cesim Erten, Dan Ismailescu, Stephen G. Kobourov, Anna Lubiw, and Joseph S. B. Mitchell, *On simultaneous planar graph embeddings*, Comput. Geom., 36 (2007), pp. 117–130.

[8] Gerth Stølting Brodal and Riko Jacob, *Dynamic planar convex hull*, in Proc. of the 43rd Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2002, pp. 617–626.

[9] Sergio Cabello, *Planar embeddability of the vertices of a graph using a fixed point set is NP-hard*, J. Graph Algorithms Appl., 10 (2006), pp. 353–363.

[10] Netzahualcoyotl Castañeda and Jorge Urrutia, *Straight line embeddings of planar graphs on point sets*, in Proc. of CCCG, 1996, pp. 312–318.

[11] Bernard Chazelle, Micha Sharir, and Emo Welzl:, *Quasi-optimal upper bounds for simplex range searching and new zone theorems*, Algorithmica, 8 (1992), pp. 407–429.

[12] Marek Chrobak and Howard J. Karloff, *A lower bound on the size of universal sets for planar graphs*, SIGACT News, 20 (1989), pp. 83–86.

[13] Hubert de Fraysseix, János Pach, and Richard Pollack, *How to draw a planar graph on a grid*, Combinatorica, 10 (1990), pp. 41–51.

[14] Eric D. Demaine and André Schulz, *Embedding stacked polytopes on a polynomial-size grid*, in Proc. of the 22nd Annual ACM-SIAM symposium on Discrete Algorithms (SODA), ACM, 2011, pp. 77–80.

[15] Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, and Stephen K. Wismath, *Curve-constrained drawings of planar graphs*, Comput. Geom., (2005), pp. 1–23.

[16] Emilio Di Giacomo and Giuseppe Liotta, *The Hamiltonian augmentation problem and its applications to graph drawing*, in Proc. of the 4th Workshop on Algorithms and Computation (WALCOM), vol. 5942, Springer, 2010, pp. 35–46.

[17] Robert P. Dilworth, *A decomposition theorem for partially ordered sets*, Ann. of Math., 51 (1950), pp. 161–166.

[18] Vida Dujmović, William Evans, Sylvain Lazard, William Lenhart, Giuseppe Liotta, David Rappaport, and Stephen K. Wismath, *On point-sets that support planar graphs*, Comput. Geom., (2012).

[19] Stephane Durocher and Debajyoti Mondal, *On the hardness of point-set embeddability*, in Proc. of the 6th International Workshop on Algorithms and Computation (WALCOM), vol. 7157 of LNCS, Springer, 2012, pp. 148–159.

[20] ———, *Plane 3-trees: Embeddability & approximation (extended abstract)*, in Proc. of the 13th International Symposium on Algorithms and Data Structures (WADS), vol. 8037, Springer, 2013, pp. 291–303.

[21] Stephane Durocher, Debajyoti Mondal, Rahnuma Islam Nishat, Md. Saidur Rahman, and Sue Whitesides, *Embedding plane 3-trees in $\mathbb{R}^2$ and $\mathbb{R}^3$*, in Proc. of the 19th International Symposium on Graph Drawing (GD), vol. 7034 of LNCS, Springer, 2012.

[22] Jeff Erickson, *On the relative complexities of some geometric problems*, in Proc. of the 7th Canadian Conference on Computational Geometry (CCCG), 1995, pp. 85–90.

[23] Hazel Everett, Sylvain Lazard, Giuseppe Liotta, and Stephen K. Wismath, *Universal sets of n points for one-bend drawings of planar graphs with n vertices*, Discrete & Computational Geometry, 43 (2010), pp. 272–288.

[24] Yoshiko Ikebe, Micha A. Perles, Akihisa Tamura, and Shinnichi Tokunaga, *The rooted tree embedding problem into points in the plane*, Discrete & Comp. Geometry, 11 (1994), pp. 51–63.

[25] Bastian Katz, Marcus Krug, Ignaz Rutter, and Alexander Wolff, *Manhattan-geodesic embedding of planar graphs*, in Proc. of the 17th International Symposium on Graph Drawing (GD), vol. 5849 of LNCS, Springer, 2010, pp. 207–218.

[26] Michael Kaufmann and Roland Wiese, *Embedding vertices at points: Few bends suffice for planar graphs*, J. Graph Algorithms Appl., 6 (2002), pp. 115–129.

[27] Maciej Kurowski, *A 1.235 lower bound on the number of points needed to draw all n-vertex planar graphs*, Information Processing Letters, 92 (2004), pp. 95–98.

[28] Debajyoti Mondal, *Embedding a planar graph on a given point set*, master's thesis, University of Manitoba, Canada, 2012.

[29] Tanaeem Md. Moosa and Md. Sohel Rahman, *Improved algorithms for the point-set embeddability problem for plane 3-trees*, in Proc. of the 17th Annual International Conference on Computing and Combinatorics (COCOON), vol. 6842, Springer, 2011, pp. 204–212.

[30] Rahnuma Islam Nishat, Debajyoti Mondal, and Md. Saidur Rahman, *Point-set embeddings of plane 3-trees*, Comput. Geom., 45 (2012), pp. 88–98.

[31] Mark H. Overmars and Jan van Leeuwen, *Maintenance of configurations in the plane*, Journal of Computer and System Sciences, 23 (1981), pp. 166–204.

[32] János Pach and Rephael Wenger, *Embedding planar graphs at fixed vertex locations*, Graphs Combin., 17 (2001), pp. 717–728.

[33] Walter Schnyder, *Embedding planar graphs on the grid*, in Proc. of the 1st Annual ACM-SIAM symposium on Discrete algorithms (SODA), ACM, 1990, pp. 138–148.

[34] Michael Ian Shamos and Dan Hoey, *Closest-point problems*, in Proc. of the 16th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 1975, pp. 151–162.

[35] William L. Steiger and Ileana Streinu, *Illumination by floodlights*, Comput. Geom., 10 (1998), pp. 57–70.

[36] Roberto Tamassia, Giuseppe Di Battista, and Carlo Batini, *Automatic graph drawing and readability of diagrams*, IEEE Transactions on Systems, Man and Cybernetics, 18 (1988), pp. 61–79.