Minimum-Area Drawings of Plane 3-Trees
(Extended Abstract)

Debajyoti Mondal,[*] Rahnuma Islam Nishat,[†] Md. Saidur Rahman,[‡] and Muhammad Jawaherul Alam[§]

Abstract

A straight-line grid drawing of a plane graph G is a planar drawing of G, where each vertex is drawn at a grid point of an integer grid and each edge is drawn as a straight-line segment. The area of such a drawing is the area of the smallest axis-aligned rectangle on the grid which encloses the drawing. A minimum-area drawing of a plane graph G is a straight-line grid drawing of G where the area of the drawing is the minimum. Although it is NP-hard to find minimum-area drawings for general plane graphs, in this paper we obtain minimum-area drawings for plane 3-trees in polynomial time. Furthermore, we show a $\lfloor \frac{2n}{3} - 1 \rfloor \times 2\lceil \frac{n}{3} \rceil$ lower bound for the area of a straight-line grid drawing of a plane 3-tree with $n \geq 6$ vertices, which improves the previously known lower bound $\lfloor \frac{2(n-1)}{3} \rfloor \times \lfloor \frac{2(n-1)}{3} \rfloor$ for plane graphs.

1 Introduction

Straight-line drawing of plane graphs is a classical area of investigation of Graph Drawing. Schnyder [9] and de Fraysseix *et al.* [4] independently showed that every plane graph with n vertices has a straight-line grid drawing on area $(n-2) \times (n-2)$ and $(2n-4) \times (n-2)$, respectively. Krug and Wagner proved that the problem of finding minimum-area drawings for plane graphs is NP-hard [7]. A variant of straight-line drawing style is *layered drawing* of plane graphs where the vertices are drawn on a set of horizontal lines called *layers* and the edges are drawn as straight line segments. A *minimum-layer drawing* of a plane graph G is a layered drawing of G where the number of layers is the minimum.

In this paper, we give an $O(n^9 \log n)$ time algorithm to obtain minimum-area drawings of "plane 3-trees". We also show that, there exists a plane 3-tree with $n \geq 6$ vertices for which $\lfloor \frac{2n}{3} - 1 \rfloor \times 2\lceil \frac{n}{3} \rceil$ area is necessary for any planar straight-line grid drawing. As a side result, we give an $O(nh_m^4)$ time algorithm to compute a minimum-layer drawing of a plane 3 tree G, where h_m is the minimum number of layers required for any layered drawing of G. Note that, Dujmović *et al.* gave an algorithm to decide whether a plane graph G admits a planar drawing in h layers using a "path decomposition" of G [5]. But the algorithm currently known to obtain a path decomposition of a plane 3-tree takes at least $\Omega(n^{15})$ time [2].

A *plane* 3*-tree* G with $n \geq 3$ vertices is a plane graph for which the following (a) and (b) hold: (a) G is a triangulated plane graph; (b) if $n > 3$, then G has a vertex x whose deletion gives a plane 3-tree G' of $n-1$ vertices. Note that, vertex x may be an inner vertex or an outer vertex of G. We use "dynamic programming" to test whether G has a drawing on a given area or on a set of layers. We show that the testing problem can be divided into three subproblems. More precisely, we prove that G can be divided into three subgraphs which can be used as the input of the subproblems of the testing problem. We solve those subproblems recursively and combine their results to obtain the result of the testing problem.

2 Preliminaries

For graph theoretic terminologies see [8]. In the rest of this section we present some preliminary results. The following results are known on plane 3-trees [1].

Lemma 1 *Let G_n be a plane 3-tree with n vertices where $n > 3$. Then the following (a) and (b) hold. (a) G_n has an inner vertex x of degree three such that the removal of x gives the plane 3-tree G_{n-1}. (b) G_n has exactly one inner vertex p such that p is the neighbor of all the three outer vertices of G_n.*

We call vertex p mentioned in Lemma 1(b) the *representative vertex* of G_n. For a cycle C in G_n, we denote by $G_n(C)$ the plane subgraph of G_n inside C (including C). We now have the following lemma.

Lemma 2 *Let G_n be a plane 3-tree and C be any triangle of G_n. Then the subgraph $G_n(C)$ is a plane 3-tree.*

Let p be the representative vertex and a, b and c be the outer vertices of G_n. We call the triangles abp, bcp and cap the three *nested triangles around* p.

We now define a *representative tree* of G_n as an ordered rooted tree T_{n-3} satisfying the following (a) and (b). (a) If $n = 3$, T_{n-3} consists of a single vertex. (b) If $n > 3$, then the root p of T_{n-3} is the representative vertex of G_n and the subtrees rooted at the three counterclockwise ordered children q_1, q_2 and q_3 of p in T_{n-3} are

[*]Department of CSE, BUET, debajyoti@cse.buet.ac.bd
[†]Department of CSE, BUET, nishat.buet@gmail.com
[‡]Department of CSE, BUET, saidurrahman@cse.buet.ac.bd
[§]Department of CSE, BUET, jawaherul@yahoo.com

the representative trees of $G_n(C_1)$, $G_n(C_2)$ and $G_n(C_3)$, respectively, where C_1, C_2 and C_3 are the three nested triangles around p in counter-clockwise order.

Figure 1 depicts an example of a representative tree.

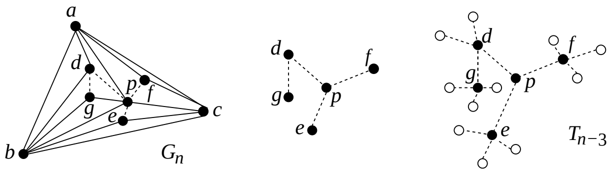


Figure 1: Representative tree T_{n-3} of G_n.

We obtain the following result using Lemma 1 and Lemma 2 .

Lemma 3 *Let G_n be a plane 3-tree with $n \geq 3$ vertices. Then G_n has a unique representative tree T_{n-3} with exactly $n-3$ internal vertices and $2n-5$ leaves. Moreover, T_{n-3} can be found in time $O(n)$.*

Let T be a tree. We denote by $T(i)$ the subtree of T rooted at vertex i. We now have the following lemma which is immediate from the definition of the representative tree and from Lemma 3.

Lemma 4 *Let T be the representative tree of a plane 3-tree G and let i be a vertex of T. Then there exists a unique triangle C in G such that $T(i)$ is the representative tree of $G(C)$.*

By Lemma 4, for any vertex u of T_{n-3}, there is a unique triangle in G_n which we denote as C_u. For the rest of this article, we shall often use an internal vertex u of T_{n-3} and the representative vertex of $G_n(C_u)$ interchangeably.

3 Minimum-Layer Drawings

Let $y(l)$ be the y-coordinate of a layer l. Let $\{l_1, l_2, ..., l_n\}$ be a set of n layers where $y(l_1) < y(l_2) < ... < y(l_n)$, then $y(l_i) = i$, $1 \leq i \leq n$.

Let G be a plane 3-tree. Since G admits a drawing on $\lfloor \frac{2n-1}{3} \rfloor$ layers [3], we give an algorithm **Min-Layer** to generate all the feasible y-coordinate assignments of the vertices of G iterating height h from 1 to $\lfloor \frac{2n-1}{3} \rfloor$. Then we give an algorithm **Feasibility-Check** to check, in each iteration, whether G admits a layered drawing on h layers for a particular y-coordinate assignment of its outer vertices. We now formally define the decision problem *Feasibility Checking.*

Input: A plane 3-tree G and y-coordinate assignments of the three outer vertices a, b and c of G.

Output: If G admits a layered drawing with the given y-coordinates of a, b and c, the output is $True$, and $False$ otherwise.

We use a dynamic programming approach to solve the Feasibility Checking problem. To obtain a recursive solution of the problem, we need the following lemmas.

Lemma 5 *Let G be a plane 3-tree with representative vertex u. Let Γ_u be a layered drawing of G and let $\Gamma(C_u)$ be the layered drawing of C_u in Γ_u. Let $\Gamma'(C_u)$ be another layered drawing of C_u where a, b and c have the same y-coordinates as in $\Gamma(C_u)$. Then G has a layered drawing Γ'_u having $\Gamma'(C_u)$ as the drawing of C_u.*

Proof. The case for $n = 3$ is trivial since for this case Γ'_u coincides with $\Gamma'(C_u)$. We may thus assume that $n > 3$ and the claim holds for any plane 3-tree of less than n vertices. Let $u_y = y(l)$, where u_y is the y-coordinate of u in Γ_u. The layer l intersects $\Gamma'(C_u)$ at two points (x_1, u_y) and (x_2, u_y), $x_1 \neq x_2$. We place u on l in between x_1 and x_2 to obtain $\Gamma'(C_{q_1})$, $\Gamma'(C_{q_2})$ and $\Gamma'(C_{q_3})$ where C_{q_1}, C_{q_2} and C_{q_3} are the nested triangles around u. By induction hypothesis $G(C_{q_1})$, $G(C_{q_2})$ and $G(C_{q_3})$ admit layered drawings Γ'_{q_1}, Γ'_{q_2} and Γ'_{q_3} which contain the drawings $\Gamma'(C_{q_1})$, $\Gamma'(C_{q_2})$ and $\Gamma'(C_{q_3})$, respectively. Clearly, one can obtain Γ'_u by patching Γ'_{q_1}, Γ'_{q_2} and Γ'_{q_3} inside $\Gamma'(C_{q_1})$, $\Gamma'(C_{q_2})$ and $\Gamma'(C_{q_3})$, respectively. $\Box$

Lemma 6 *Let G be a plane 3-tree with the representative tree T. Let u be any internal vertex of T with the three children q_1, q_2, q_3 in T and let a, b, c be the three outer vertices of $G(C_u)$. Then $G(C_u)$ admits a layered drawing Γ_u for the assignment (a_y, b_y, c_y) if and only if Γ_{q_1}, Γ_{q_2} and Γ_{q_3} admit layered drawings for the assignments (a_y, b_y, u_y), (b_y, c_y, u_y) and (c_y, a_y, u_y), respectively, where $min(a_y, b_y, c_y) < u_y < max(a_y, b_y, c_y)$.*

Proof. The necessity is trivial, and proof of the sufficiency can be obtained in a similar technique as described in the proof of Lemma 5. $\Box$

We now give the recursive solution of the Feasibility Checking problem as in the following theorem.

Theorem 1 *Let G be a plane 3-tree with the representative tree T and u be any vertex of T. Let a, b, c be the three outer vertices of $G(C_u)$ and q_1, q_2, q_3 be the three children of u if u is an internal vertex of T. Let $F_u(a_y, b_y, c_y)$ denote the* Feasibility Checking *problem of u where a_y, b_y, c_y are the y-coordinates of a, b, c. Then $F_u(a_y, b_y, c_y)$ has the following recursive formula.*

$$F_u(a_y, b_y, c_y) = \begin{cases} False \text{ if } max\{a_y, b_y, c_y\} - min\{a_y, b_y, c_y\} = 0; \\ True \text{ if } max\{a_y, b_y, c_y\} - min\{a_y, b_y, c_y\} \geq 1, \\ \qquad \text{where } u \text{ is a leaf;} \\ False \text{ if } max\{a_y, b_y, c_y\} - min\{a_y, b_y, c_y\} \leq 1, \\ \qquad \text{where } u \text{ is an internal vertex;} \\ \bigvee_{u_y} \{F_{q_1}(a_y, b_y, u_y) \wedge F_{q_2}(b_y, c_y, u_y) \wedge F_{q_3}(c_y, a_y, u_y)\} \\ \qquad \text{where } min\{a_y, b_y, c_y\} < u_y < max\{a_y, b_y, c_y\}, \\ \qquad \text{otherwise.} \end{cases}$$

Proof. Consider the case when $max\{a_y, b_y, c_y\} - min\{a_y, b_y, c_y\} = 0$. Then we assign $F_u(a_y, b_y, c_y) = False$, since a triangle cannot be drawn on a single layer. The next case is $max\{a_y, b_y, c_y\} - min\{a_y, b_y, c_y\} \geq 1$ when u is a leaf. Then we assign $F_u(a_y, b_y, c_y) = True$ since two layers are sufficient to draw a triangle. The next case is $max\{a_y, b_y, c_y\} - min\{a_y, b_y, c_y\} \leq 1$ when u is an internal vertex. Then we assign $F_u(a_y, b_y, c_y) = False$ for this case since the outer face needs two layers to be drawn and the inner vertex u cannot be placed on any of them. The remaining case is $max\{a_y, b_y, c_y\} - min\{a_y, b_y, c_y\} > 1$ when u is an internal vertex. Then we define $F_u(a_y, b_y, c_y)$ recursively by Lemma 6. □

For each vertex i of T we associate a table $FC_i[1{:}\lfloor \frac{2n+2}{3} \rfloor, 1{:}\lfloor \frac{2n+2}{3} \rfloor, 1{:}\lfloor \frac{2n+2}{3} \rfloor]$, where the solution of $F_i(a_y, b_y, c_y)$ is stored. To store the computed y-coordinates of the vertices of G, we maintain another table $\mathrm{Y}_i[1{:}\lfloor \frac{2n+2}{3} \rfloor, 1{:}\lfloor \frac{2n+2}{3} \rfloor, 1{:}\lfloor \frac{2n+2}{3} \rfloor]$ for each vertex i of T. Here, $Y_i(a_y, b_y, c_y) =$

$$\begin{cases} False \text{ if } FC_i[a_y, b_y, c_y] = \text{False}; \\ True \text{ if } i \text{ is a leaf and } FC_i[a_y, b_y, c_y] = \text{True}; \\ i_y \text{ if } i \text{ is an internal vertex and } FC_i[a_y, b_y, c_y] = \text{True}. \end{cases}$$

Let a, b, c be the outer vertices and u be the representative vertex of G. If $\mathrm{Y}_u[a_y, b_y, c_y]$ is $False$, then G has no layered drawing for the given y-coordinate assignment a_y, b_y, c_y. If the entry is *True*, then G has no inner vertex and G has a layered drawing for the given y-coordinate assignment. Otherwise, G has a layered drawing for the given y-coordinate assignment and the entry $\mathrm{Y}_u(a_y, b_y, c_y)$ contains the y-coordinate of the representative vertex u.

One can obtain the y-coordinate assignment of each internal vertex of G, using Y_u by a preorder traversal of the representative tree. Since, by Lemma 3 T has $n-3$ internal vertices, this process takes $O(n)$ time.

We now describe Algorithm **Min-Layer** which computes the minimum number of layers required to draw G using Algorithm **Feasibility-Check**. We assume that G admits a layered drawing on h layers and iterate h from 1 to $\lfloor \frac{2n-1}{3} \rfloor$. At each iteration we traverse T in preorder and for each vertex i of T, Algorithm **Min-Layer** generates all possible y-coordinate assignments for the outer vertices a, b and c of $G(C_i)$ within h layers. For each such assignment a_y, b_y and c_y, Algorithm **Feasibility-Check** is called to check whether $G(C_i)$ is drawable. The first h within which G is drawable is the minimum number of layers h_m required to draw G. We now have the following theorem.

Theorem 2 *Given a plane 3-tree G with n vertices, Algorithm **Min-Layer** computes the minimum number of layers h_m required to draw G on layers in $O(nh_m^4)$ time.*

Outline of the Proof. By Lemma 3 the representative tree T of G can be constructed in $O(n)$ time. We then assume a height h and iterate h from 2 to $\lfloor \frac{2n-1}{3} \rfloor + 1$. At each iteration, for each internal vertex i of T, we check the drawability of $G(C_i)$ for only the new combinations of y-coordinates of the outer vertices a, b, c of $G(C_i)$. More precisely, for each vertex $v \in \{a, b, c\}$ we put v on the h-th layer and check the drawability assigning different y-coordinates to the other two outer vertices. One can observe that the new combinations possible at each iteration is $O(h^2)$. Hence, after all the iterations of h, for all the internal vertices of T, we have to check the drawability for $h \times n \times O(h^2) = O(nh_m^3)$ times. If, for the different y-coordinate assignments of the representative vertex i, we use the stored results of the subproblems to obtain the solution, we can check the drawability in $O(h)$ time at each iteration. Thus Algorithm **Min-Layer** takes $O(h) \times O(nh_m^3) = O(nh_m^4)$ time in total. □

4 Minimum-Area Drawings

Like the *Feasibility Checking* problem for minimum-layer drawings, one can formulate a problem *Area Checking* for minimum-area drawings. We denote the x-coordinate and y-coordinate of a vertex v by v_x and v_y, respectively. We now have the following theorem.

Theorem 3 *Let G be a plane 3-tree with the representative tree T and u be any vertex of T. Let a, b, c be the three outer vertices of $G(C_u)$ and q_1, q_2, q_3 be the three children of u when u is an internal vertex of T. Let $A_u(a_x, a_y, b_x, b_y, c_x, c_y)$ be the Area Checking problem of u where a, b and c have distinct (x, y)-coordinates. Then $A_u(a_x, a_y, b_x, b_y, c_x, c_y)$ has the following recursive formula. $A_u(a_x, a_y, b_x, b_y, c_x, c_y) =$*

$$\begin{cases} False \textit{ if } (max\{a_x, b_x, c_x\} - min\{a_x, b_x, c_x\} = 0) \\ \qquad \vee\ (max\{a_y, b_y, c_y\} - min\{a_y, b_y, c_y\} = 0); \\ True \textit{ if } (max\{a_x, b_x, c_x\} - min\{a_x, b_x, c_x\} \geq 1) \\ \qquad \wedge\ (max\{a_y, b_y, c_y\} - min\{a_y, b_y, c_y\} \geq 1) \\ \qquad \wedge\ u \textit{ is a leaf;} \\ False \textit{ if } ((max\{a_x, b_x, c_x\} - min\{a_x, b_x, c_x\} \leq 1) \\ \qquad \vee\ (max\{a_y, b_y, c_y\} - min\{a_y, b_y, c_y\} \leq 1)) \\ \qquad \wedge\ u \textit{ is an internal vertex;} \\ \bigvee_{u_x, u_y} \{A_{q_1}(a_x, a_y, b_x, b_y, u_x, u_y) \wedge \\ \qquad A_{q_2}(b_x, b_y, c_x, c_y, u_x, u_y) \wedge \\ \qquad A_{q_3}(c_x, c_y, a_x, a_y, u_x, u_y)\} \\ \qquad \textit{where } (u_x, u_y) \textit{ is inside the triangle with the} \\ \qquad \textit{vertices } a,\ b,\ c, \textit{ otherwise.} \end{cases}$$

To store the solution of $A_u(a_x, a_y, b_x, b_y, c_x, c_y)$ and the (x, y)-coordinates of the vertices of G, we use the same technique as used for computing y-coordinates in Section 3. We now describe Algorithm **Min-Area** to obtain minimum-area drawings. Since the upper bound of the area of straight-line grid drawings of planar graphs is $O(n^2)$ [9], this bound also holds for any plane 3-tree G with n vertices. Since the minimum number of layers required for any straight-line grid drawing of G is h_m, the upper bound for width is $\lceil n^2/h_m \rceil$. Therefore, we assume a width w and a height h for G. We iterate h from 1 to n and for each h, we iterate w from 1 to $min(\lceil \frac{n^2}{h} \rceil, \lceil \frac{n^2}{h_m} \rceil)$. At each iteration we traverse T in preorder. For each internal vertex i of T, **Min-Area** generates all possible (x, y)-coordinate assignments for the outer vertices a, b and c of $G(C_i)$ within area $w \times h$. For each such (x, y)-coordinate assignment of a, b and c, we check whether $G(C_i)$ is drawable. Each time a drawing of G with smaller area is found, the stored area is replaced by the smaller area and at the end of the algorithm, the stored area is the minimum. We now have the following theorem.

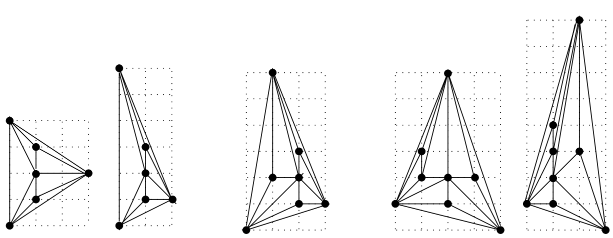
Theorem 4 *Given a plane 3-tree G with $n \geq 3$ vertices, Algorithm **Min-Area** gives a minimum-area drawing of G in $O(n^9 \log n)$ time.*

5 Lower Bound

It is known that there exists a plane graph with n vertices for which any straight-line grid drawing requires at least $(\frac{2n}{3} - 1) \times (\frac{2n}{3})$ area where n is a multiple of three [6]. For general n, the lower bound on area is known to be $\lfloor \frac{2(n-1)}{3} \rfloor \times \lfloor \frac{2(n-1)}{3} \rfloor$ area [3] which we improve to $\lfloor \frac{2n}{3} - 1 \rfloor \times 2\lceil \frac{n}{3} \rceil$ area for $n \geq 6$.

Theorem 5 *For each $n \geq 6$, there is a plane graph G with n vertices such that the area required to obtain a straight-line grid drawing of G is at least $\lfloor \frac{2n}{3} - 1 \rfloor \times 2\lceil \frac{n}{3} \rceil$.*

Proof. The lower bound on area can be obtained in a similar technique as shown in [6] by nesting the graphs of Figure 2 inside the "nested triangles graphs". □



(a) (b) (c)

Figure 2: Illustration of Theorem 5 when (a) $n = 6$, (b) $n = 7$ and (c) $n = 8$.

6 Conclusion

We have shown that for a fixed planar embedding of a plane 3-tree G, a minimum-area drawing can be obtained in polynomial time. Since a plane 3-tree G has only linear number of planar embeddings, we can compute the area requirements of all the embeddings of G and determine the planar embedding which gives the best area bound; and thus we can obtain a minimum-area drawing of G in polynomial time when the embedding of G is not fixed. It is left as a future work to find a simpler algorithm for obtaining minimum-area drawings of plane 3-trees. It is also a challenge to find other classes of planar graphs for which the area minimization problem can be solved in polynomial time.

Acknowledgment. This work is done in Graph Drawing and Information Visualization Laboratory, Department of CSE, BUET, under the project "Minimum-Area Drawings of Plane Graphs" supported by CASR, BUET.

References

[1] T. Biedl and L. E. R. Velasquez. Drawing planar 3-trees with given face-areas. In *The 17th International Symposium on Graph Drawing*, volume 5849 of *LNCS*, pages 316–322. Springer, 2010.

[2] H. L. Bodlaender and T. Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21(2):358–402, 1996.

[3] M. Chrobak and S. Nakano. Minimum width grid drawings of plane graphs. *Computational Geometry*, 11(1):104–110, 1998.

[4] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990.

[5] V. Dujmović, M. Fellows, M. Hallett, M. Kitching, G. Liotta, C. Mccartin, N. Nishimura, P. Ragde, F. Rosamond, M. Suderman, S. Whitesides, and D. R. Wood. On the parameterized complexity of layered graph drawing. *Algorithmica*, 52(2):267–292, 2008.

[6] F. Frati and M. Patrignani. A note on minimum area straight-line drawings of planar graphs. In *The 15th International Symposium on Graph Drawing*, volume 4875 of *LNCS*, pages 339–344, 2008.

[7] M. Krug and D. Wagner. Minimizing the area for planar straight-line grid drawings. In *The 15th International Symposium on Graph Drawing*, volume 4875 of *LNCS*, pages 207–212, 2008.

[8] T. Nishizeki and M. S. Rahman. *Planar Graph Drawing.* World Scientific, Singapore, 2004.

[9] W. Schnyder. Embedding planar graphs on the grid. In *The first annual ACM-SIAM Symposium on Discrete Algorithms*, pages 138–148, 1990.