Area-Efficient Drawings of Outerplanar Graphs*

Ashim Garg and Adrian Rusu

Department of Computer Science and Engineering
University at Buffalo
Buffalo, NY 14260
{agarg,adirusu}@cse.buffalo.edu

Abstract. It is well-known that a planar graph with n nodes admits a planar straight-line grid drawing with $O(n^2)$ area [3, 8], and in the worst case it requires $\Omega(n^2)$ area. It is also known that a binary tree with n nodes admits a planar straight-line grid drawing with $O(n)$ area [6]. Thus, there is wide gap between the $\Theta(n^2)$ area-requirement of general planar graphs and the $\Theta(n)$ area-requirement of binary trees. It is therefore important to investigate special categories of planar graphs to determine if they can be drawn in $o(n^2)$ area.
Outerplanar graphs form an important category of planar graphs. We investigate the area-requirement of planar straight-line grid drawings of outerplanar graphs. Currently the best known bound on the area-requirement of such a drawing of an outerplanar graph with n vertices is $O(n^2)$, which is that same as for general planar graphs. Hence, a fundamental question arises that can be draw an outerplanar graph in this fashion in $o(n^2)$ area?
In this paper, we provide a partial answer to this question by proving that an outerplanar graph with n vertices and degree d can be drawn in this fashion in area $O(dn^{1.48})$ in $O(n \log n)$ time. This implies that an outerplanar graph with n vertices and degree d, where $d = o(n^{0.52})$, can be drawn in this fashion in $o(n^2)$ area.
From a broader perspective, our contribution is in showing a sufficiently large natural category of planar graphs that can be drawn in $o(n^2)$ area.

1 Introduction

A *drawing* Γ of a graph G maps each vertex of G to a distinct point in the plane, and each edge (u, v) of G to a simple Jordan curve with endpoints u and v. Γ is a *straight-line* drawing, if each edge is drawn as a single line-segment. Γ is a *polyline* drawing, if each edge is drawn as a connected sequence of one or more line-segments, where the meeting point of consecutive line-segments is called a *bend*. Γ is a *grid* drawing if all the nodes have integer coordinates. Γ is a *planar* drawing, if edges do not intersect each other in the drawing. In this paper, we concentrate on grid drawings. So, we will assume that the plane is covered by a

* Research supported by NSF CAREER Award IIS-9985136 and NSF CISE Research Infrastructure Award No. 0101244, and Mark Diamond Research Grant No. 13-Summer-2003 from GSA of The State University of New York.

rectangular grid. Let Γ be a grid drawing. Let R be the smallest rectangle with sides parallel to the X-and Y-axes, respectively, that covers Γ completely. The *width* (*height*) of Γ is equal to 1+ width of R (1+height of R). The *area* of Γ is equal to (1+width of R)$\cdot$(1+height of R), which is equal to the number of grid points contained within R. The *degree* of a graph is equal to the maximum number of edges incident on a vertex.

It is well-known that a planar graph with n vertices admits a planar straight-line grid drawing with $O(n^2)$ area [3, 8], and in the worst case it requires $\Omega(n^2)$ area. It is also known that a binary tree with n nodes admits a planar straight-line grid drawing with $O(n)$ area [6]. Thus, there is wide gap between the $\Theta(n^2)$ area-requirement of general planar graphs and the $\Theta(n)$ area-requirement of binary trees. It is therefore important to investigate special categories of planar graphs to determine if they can be drawn in $o(n^2)$ area.

Outerplanar graphs form an important category of planar graphs. We investigate the area-requirement of planar straight-line grid drawings of outerplanar graphs. Currently the best known bound on the area-requirement of such a drawing of an outerplanar graph with n vertices is $O(n^2)$, which is that same as for general planar graphs. Hence, a fundamental question arises: can we draw an outerplanar graph in this fashion in $o(n^2)$ area?

In this paper, we provide a partial answer to this question by proving that an outerplanar graph with n vertices and degree d can be drawn in this fashion in area $O(dn^{1+0.48}) = O(dn^{1.48})$ in $O(n)$ time. This implies that an outerplanar graph with n vertices and degree $O(n^\delta)$, where $0 \leq \delta < 0.52$ is a constant, can be drawn in this fashion in $o(n^2)$ area.

From a broader perspective, our contribution is in showing a sufficiently large natural category of planar graphs that can be drawn in $o(n^2)$ area.

In Section 4, we present our drawing algorithm. This algorithm is based on a tree-drawing algorithm of [2]. The connection between the two algorithms comes from the fact that the dual of a maximal outerplanar graph is a tree.

2 Previous Results

There has been little work done on planar straight-line grid drawings of outerplanar graphs. Let G be an outerplanar graph with n vertices. Currently the best known bound on the area-requirement of such a drawing of an outerplanar graph with n vertices is $O(n^2)$, which is that same as for general planar graphs. However, in 3D, we can construct a crossings-free straight-line grid drawing of G with $O(n)$ volume [4, 5].

[1] shows that G admits a planar polyline drawing as well as a visibility representation with $O(n \log n)$ area. [7] shows that G admits a planar polyline drawing with $O(n)$ area, if G has degree 4. The technique of [7] can be easily extended to construct a planar polyline drawing of G with $O(d^2 n)$ area, if G has degree d [1].

3 Preliminaries

We assume a 2-dimensional Cartesian space. We assume that this space is covered by an infinite rectangular grid, consisting of horizontal and vertical channels.

We denote by $|G|$, the number of vertices (nodes) in a graph (tree) G. A *rooted* tree is one with a pre-specified root. An *ordered* tree is a rooted tree with a pre-specified left-to-right order of the children for each node. Let T be an ordered binary tree with n nodes. Let p and δ be two constants such that $p = 0.48$ and $0 < \delta \leq 0.0004$. A *spine* S of T is a path $v_0 v_1 v_2 \dots v_m$, where $v_0, v_1, v_2, \dots, v_m$ are nodes of T, that is defined recursively as follows (as defined in the proof of Lemma A.1 in [2]):

- v_0 is the same as the root of T, and v_m is a leaf of T;
- let α_i and β_i be the the left and right subtrees with the maximum number of nodes among the subtrees that are rooted at any of the nodes in the path $v_0 v_1 \dots v_i$; let L_i and R_i be the subtrees rooted at the left and right children of v_i respectively. Then,
  - if $|\alpha_i|^p + |R_i|^p \leq (1-\delta)n^p$ and $|L_i|^p + |\beta_i|^p > (1-\delta)n^p$, set v_{i+1} to be the left child of v_i,
  - if $|\alpha_i|^p + |R_i|^p > (1-\delta)n^p$ and $|L_i|^p + |\beta_i|^p \leq (1-\delta)n^p$, set v_{i+1} to be the right child of v_i,
  - if $|\alpha_i|^p + |R_i|^p \leq (1-\delta)n^p$ and $|L_i|^p + |\beta_i|^p \leq (1-\delta)n^p$, we terminate the construction as follows:
    - if $|L_i| \leq |R_i|$, set the spine to be the concatenation of path $v_0 v_1 \dots v_i$ and the leftmost path from v_i to a leaf v_m,
    - otherwise (i.e. $|L_i| > |R_i|$), set the spine to be the concatenation of the path $v_0 v_1 \dots v_i$ and the rightmost path from v_i to a leaf v_m.
  - in [2] it is shown that the case $|\alpha_i|^p + |R_i|^p > (1-\delta)n^p$ and $|L_i|^p + |\beta_i|^p > (1-\delta)n^p$ is not possible.

$v_0, v_1, \dots, v_m$ are called *spine nodes*. A subtree T' of T is a *subtree of* S, if it is rooted at the non-spine child c of a spine node v_i; T' is a *left (right)* subtree of S, if c is the left (right) child of v_i.

We will use Lemma A.1 of [2], which is given below:

Lemma 1 (Lemma A.1 of [2]). *Let $p = 0.48$. For any left subtree α and right subtree β of a spine, $|\alpha|^p + |\beta|^p \leq (1-\delta)n^p$, for any constant δ, $0 < \delta \leq 0.0004$.*

An *outerplanar* graph is a planar graph for which there exists an embedding with all vertices on the exterior face. Throughout this paper, by the term *outerplanar* graph we will mean a *maximal* outerplanar graph, i.e., an outerplanar graph to which no edge can be added without destroying its outerplanarity. It is easy to see that each internal face of a maximal outerplanar graph is a triangle. Two vertices of a graph are *neighbors*, if they are connected by an edge. The *dual tree* T_G of an outerplanar graph G is defined as follows:

- there is a one-to-one correspondence between the nodes of T_G and the internal faces of G, and

– there is an edge $e = (u, v)$ in T_G if and only if the faces of G corresponding to u and v share an edge e' on their boundaries. e and e' are *duals* of each other.

For example, Figure 1(b), shows the dual tree of the outerplanar graph of Figure 1(a).

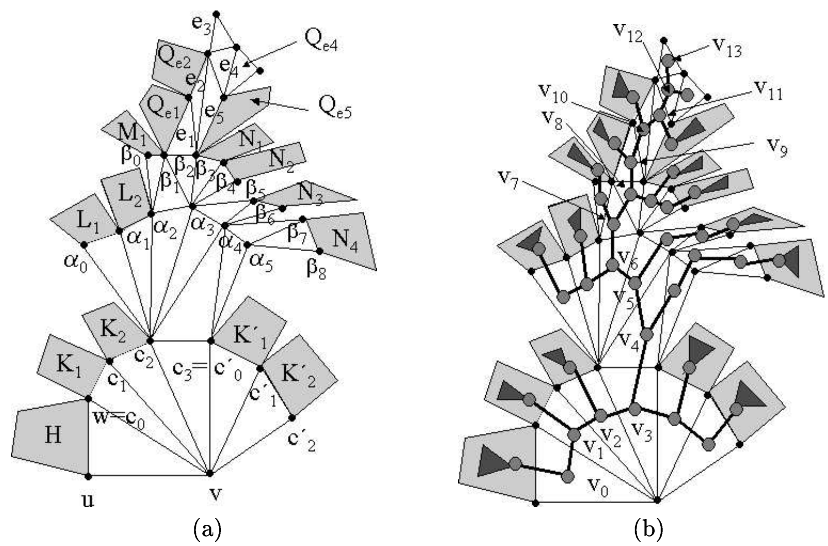

Fig. 1. (a) An outerplanar graph G. Here, H, K_1, K_2, K_1', K_2', L_1, L_2, M_1, N_1, N_2, N_3, N_4, Q_{e1}, Q_{e2}, Q_{e4}, and Q_{e5} are subgraphs of G, and are themselves outerplanar graphs. (b) The dual tree T_G of G. The edges of T_G are shown with dark lines. Note that $v_0v_1 \dots v_{13}$ is a spine of T_G.

Let $P = v_0v_1 \dots v_q$ be a path of T_G. Let H be the subgraph of G corresponding to P. A *beam* drawing of H is shown in Figure 2, where the vertices of H are placed on two horizontal channels, and the faces of H are drawn as triangles.

A line-segment with end-points a and b is a *flat* line-segment if a and b are grid points, and either belong to the same horizontal channel, or belong to adjacent horizontal channels.

Let B be a flat line-segment with end-points a and b, such that b is at least one unit to the right of a. Let G be an outerplanar graph with two distinguished adjacent vertices u and v, such that the edge (u, v) is on the external face of G; u and v are called the *poles* of G. Let D be a planar straight-line drawing of G. D is a *feasible* drawing of G with base B if:

– the two poles of G are mapped to a and b each,
– each non-pole vertex of G is placed at least one unit above the lower of a and b, and is placed at least one unit to the right of a and at least one unit to the left of b.

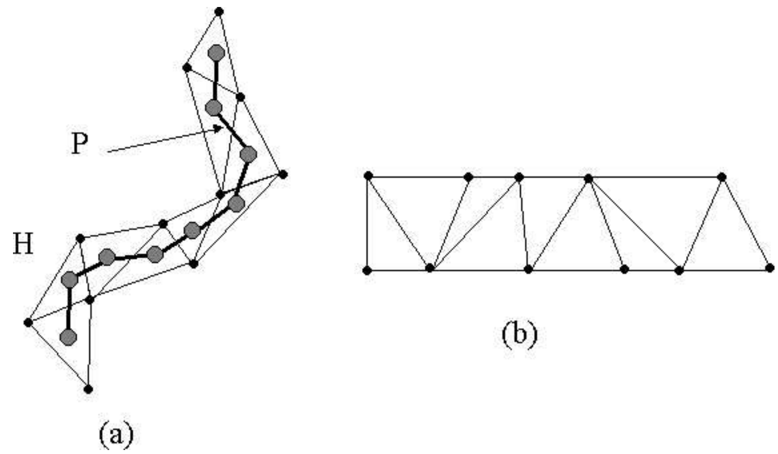

Fig. 2. (a) A path P and its corresponding graph H. (b) A beam drawing of H.

4 Outerplanar Graph Drawing Algorithm

The drawing algorithm, which we call *Algorithm OpDraw*, is recursive in nature. In each recursive step, it takes as input an outerplanar graph G with pre-specified poles, and a long-enough flat line-segment B, and constructs a feasible drawing D of G with base B by constructing a drawing M of the subgraph Z corresponding to a spine of T_G, splitting G into several smaller outerplanar graphs after removing Z and some other vertices from it, constructing feasible drawings of each smaller outerplanar graph, and then combining their drawings with M to obtain D.

We now give the details of the actions performed by *Algorithm OpDraw* in each recursive step (see Figure 3)(a):

- Let u and v be the poles of G. Let T_G be the dual tree of G. Let r be the node of T_G that corresponds to the internal face F of G that contains both u and v. Convert T_G into an ordered tree as follows:
  - make T_G a rooted tree by making r its root,
  - and for each node w, let w' be the parent of w in T_G (which now is a rooted tree). Let c (d) be the children of w such that the face corresponding to c immediately follows (precedes) the face corresponding to w' in the counter-clockwise order of internal faces incident on the face corresponding to w. Make c the leftmost child of w, and d the rightmost child of w. Assign the children of w the same left-to-right order as the counter-clockwise order in which the faces that correspond to them are incident on the face corresponding to w.

  Note that T_G is a binary tree because each internal face of G is a triangle.
- Draw F as a triangle such that u and v coincide with the end-points of B, and the third vertex w of F is placed one unit above the lower of u and v.

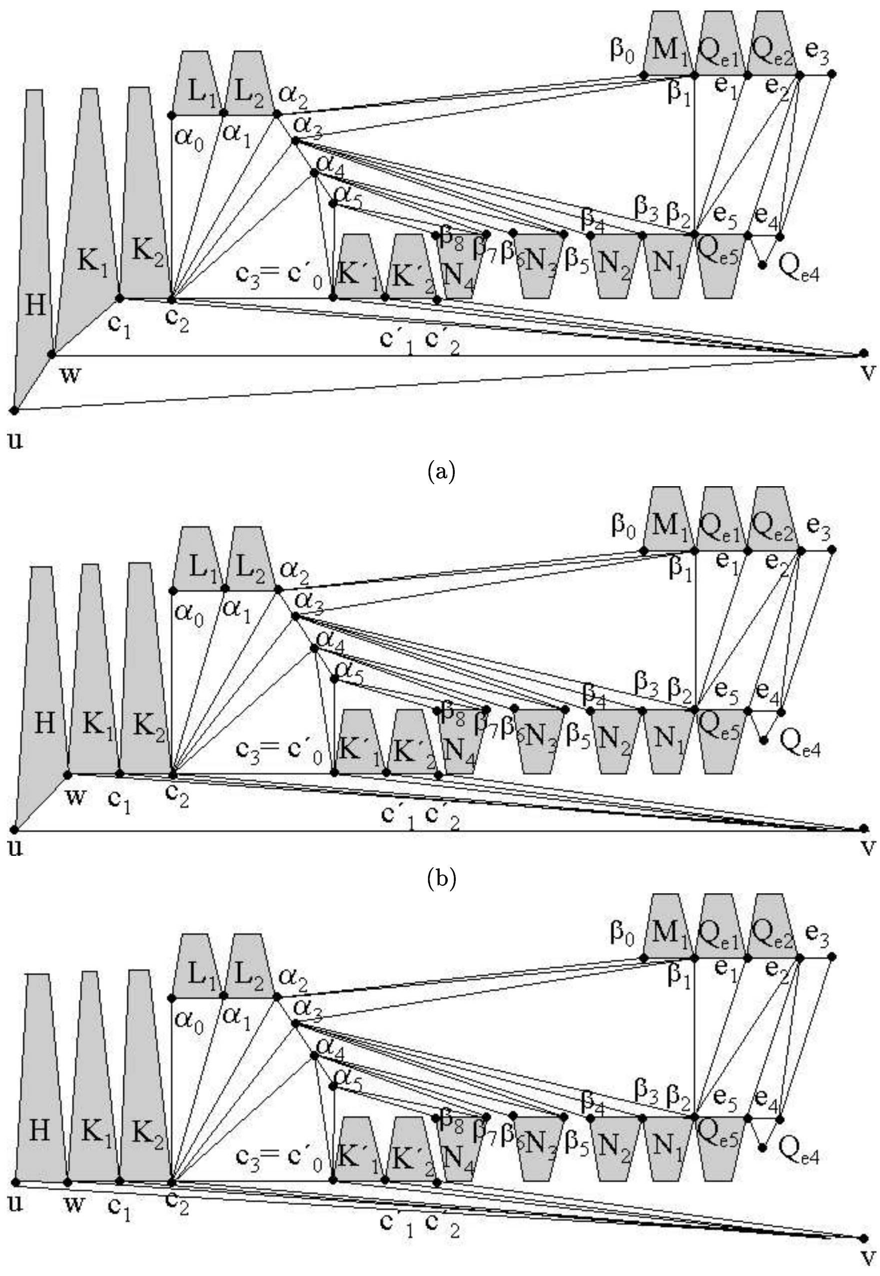

Fig. 3. The drawing of the outerplanar graph of Figure 1(a) constructed by *Algorithm OpDraw*: (a) When v is one unit above u, (b) when u and v are in the same horizontal channel, and (c) when u is one unit above v.

(We will determine later on the horizontal distances of w from u and v, when we analyze the area-requirement of the drawing.) In the rest of this section, we will assume that v is placed one unit above u. (The cases, where u and v are in the same horizontal channel, and where u is placed one unit above v are similar, and are shown in Figures 3(b) and 3(c), respectively).

- Let $P = v_0 v_1 \dots v_q$ be the spine of G, where $v_0 = r$. Assume that the edge (v_0, v_1) is the dual of edge (v, w) (the case where (v_0, v_1) is the dual of edge (u, w) is symmetrical). Let (v_0, v') be the dual of edge (u, w). Let H be the subgraph of G corresponding to the subtree of T_G rooted at v'. Recursively construct a feasible drawing D_H of H with $\overline{uw}$ as the base.
- Let $c_0 = w, c_1, \dots, c_m (= c'_0), c'_1, c'_2, \dots, c'_s$ be the clockwise order of the neighbors of v different from u, where, for each i $(1 \le i \le m)$, the face $c_{i-1} c_i v$ corresponds to the spine node v_i, and for each i $(1 \le i \le s)$, the face $c'_{i-1} c'_i v$ corresponds to a non-spine node v'_i of T_G. (In Figure 3(a), $m = 3$, and $s = 2$.) Place the vertices $c_1, \dots, c_m (= c'_0), c'_1, c'_2, \dots, c'_s$ in the same horizontal channel one unit above w. (We will determine later on the horizontal distances between these vertices.)
- Let (v_i, x_i) be the dual of edge (c_{i-1}, c_i). Let K_i be the subgraph of G corresponding to the subtree of T_G rooted at x_i. For each i, where $1 \le i \le m-1$, recursively construct a feasible drawing of K_i with $\overline{c_{i-1} c_i}$ as the base.
- Let (v'_i, x'_i) be the dual of edge (c'_{i-1}, c'_i). Let K'_i be the subgraph of G corresponding to the subtree of T_G rooted at x_i. For each i, where $1 \le i \le s$, recursively construct a feasible drawing D'_i of K'_i with $\overline{c'_{i-1} c'_i}$ as the base.
- Let $\alpha_0, \alpha_1, \dots, \alpha_t$ be the vertices of K_m, such that $\alpha_0, \alpha_1, \dots, \alpha_h$ $(0 \le h \le t)$ is the clockwise order of the neighbors of c_{m-1} in K_m, and $\alpha_h, \alpha_{h+1}, \dots, \alpha_t$ is the clockwise order of the neighbors of c_m in K_m. For example, in Figure 3(a), $h = 4$, and $t = 5$. Let j be the index such that the dual of edge (α_{j-1}, α_j) belongs to P (if no such j exists, then we can do the following: if K_m consists of only one internal face, namely, $c_{m-1} c_m \alpha_0$, then set $j = 0$. Otherwise, the leaf v_q of P will correspond to either the face $\alpha_0 \alpha_1 c_{m-1}$ or the face $\alpha_{t-1} \alpha_t c_m$; in the first case, set $j = 1$, and in the second case, set $j = t$). For example, in Figure 3(a), $j = 3$. Place $\alpha_0, \alpha_1, \dots, \alpha_{j-1}$ in the same horizontal channel, and $\alpha_{j-1}, \alpha_j, \dots, \alpha_t$ along a line making $45°$ angle with the horizontal channels, such that
  - α_t is in the same vertical channel as c_m, and at least one unit above the horizontal channel X occupied by c'_s (we will give the exact value of the vertical distance between α_t and X a little while later),
  - for each k, where $j - 1 \le k \le t - 1$, α_k is one unit above and one unit to the left of α_{k+1}, and
  - α_0 is in the same vertical channel as c_{m-1}.

  (We will determine later on the horizontal distances between $\alpha_0, \alpha_1, \dots, \alpha_{j-1}$.)
- For each i, where $0 \le i \le t$, removing α_{i-1} and α_i, splits K_m into two subgraphs, one containing c_{m-1} and c_m, and another subgraph L'_i. Let L_i be the subgraph of K_m consisting of the vertices of L'_i, α_{i-1} and α_i, and the edges between them. Recursively construct a feasible drawing of each L_i, where $0 \le i \le j-1$, with $\overline{\alpha_{i-1} \alpha_i}$ as the base.

– Let $S = \beta_0, \beta_1, \ldots, \beta_\mu$ be the clockwise order of the neighbors of $\alpha_{j-1}, \alpha_j, \ldots, \alpha_t$ in the subgraphs $L_j, L_{j+1}, \ldots, L_t$, where each β_k is different from $\alpha_{j-1}, \alpha_j, \ldots, \alpha_t$. In S, we first place the neighbors of α_{j-1}, then of α_j, and so on, finally placing the neighbors of α_t. For each k, where $j-1 \leq k \leq t$, we place the neighbors of α_k into S in the same order as their clockwise order around α_k. For example, in Figure 3(a), $\mu = 8$. Let ϵ be the index such that the dual of the edge $(\beta_{\epsilon-1}, \beta_\epsilon)$ belongs to P (if no such ϵ exists, then we can do the following: if L_j consists of only one internal face, namely, $\alpha_{j-1}\alpha_j\beta_0$, then set $\epsilon = 0$. Otherwise, the leaf v_q of P will correspond to either the face $\beta_0\beta_1\alpha_{j-1}$ or the face $\beta_{\mu-1}\beta_\mu\alpha_j$; in the first case, set $\epsilon = 1$, and in the second case, set $\epsilon = \mu$). For example, in Figure 3(a), $\epsilon = 2$.
– Place $\beta_0, \beta_1, \ldots, \beta_{\epsilon-1}$ in the same horizontal channel from left-to-right, and place $\beta_\epsilon, \beta_{\epsilon+1}, \ldots, \beta_\mu$ in another horizontal channel from right-to-left, such that:
  - $\beta_0, \beta_1, \ldots, \beta_{\epsilon-1}$ are placed one unit above α_{j-1},
  - $\beta_\epsilon, \beta_{\epsilon+1}, \ldots, \beta_\mu$ are placed one unit below α_t,
  - β_0 and β_μ are at either to the right of, or on the same vertical channel as c'_s,
  - $\beta_{\epsilon-1}$ and β_ϵ are on the same vertical channel, and
  - the distance between $\beta_{\epsilon-1}$ and β_ϵ is equal to 2 plus the vertical distance between α_{j-1} and α_t.
– For each i, where $0 \leq i \leq \epsilon - 1$, if there is an edge $e = (\beta_{i-1}, \beta_i)$ in G, then do the following: Notice that removing e from G, split it into two subgraphs, one that contains $\alpha_{j-1}, \alpha_j, \ldots, \alpha_t$, and another subgraph M'_i that does not contain any of them. Let M_i be the subgraph of G consisting of β_{i-1}, β_i, the vertices of M'_i, and the edges between them. Recursively construct a feasible drawing of M_i with $\overline{\beta_{i-1}\beta_i}$ as its base.
– For each i, where $\epsilon \leq i \leq \mu$, if there is an edge $e = (\beta_{i-1}, \beta_i)$ in G, then do the following: Notice that removing e from G, splits it into two subgraphs, one that contains $\alpha_{j-1}, \alpha_j, \ldots, \alpha_t$, and another subgraph N'_i that does not contain any of them. Let N_i be the subgraph of G consisting of β_{i-1}, β_i, the vertices of N'_i, and the edges between them. Recursively construct a feasible drawing D''_i of N_i with $\overline{\beta_{i-1}\beta_i}$ as its base, and then flip D''_i upside-down.
– Let $(v_{\rho-1}, v_\rho)$ be the edge of P that is the dual of the edge $(\beta_{\epsilon-1}, \beta_\epsilon)$. For example, in Figure 1(b), $\rho = 9$. Let R be the subgraph of G that corresponds to the subpath $v_\rho v_{\rho+1} \ldots v_q$. Construct a beam drawing E of R. For each edge e on the external face of R, do the following: Let $e = (\gamma_1, \gamma_2)$. Removing γ_1 and γ_2 from G splits it into two subgraphs, one containing $\beta_0, \beta_1, \ldots, \beta_\mu$, and the other subgraph Q'_e not containing them. Let Q_e be the subgraph of G containing γ_1, γ_2, and the vertices of Q'_e, and the edges between them. If e is on the top or bottom boundary of E, then recursively construct a feasible drawing D_e of Q_e with $\overline{\gamma_1\gamma_2}$ as its base. If e is on the bottom boundary of E, then flip Q_e upside down. (Note that if e is on the right boundary of E, then Q_e will contain just the edge e because v_q is a leaf of T_G.)
– We are now ready to give the vertical distance between α_t and X: it is equal to $1+\theta$, where θ is maximum height of any of D'_i, D''_i, and D_e, where e is on

the bottom boundary of E. Note that this will guarantee that the vertices of each D''_i and D_e will occupy horizontal channels that are either above or the same as the horizontal channel that contains $c_0 = w, c_1, \ldots, c_m(= c'_0), c'_1, c'_2, \ldots, c'_s$. This ensures that there are no crossings between the edges of any D''_i or D_e, and any edge of the form (v, c'_j).

Let $h(n)$ and $w(n)$ be the height and width, respectively, of a feasible drawing D of G with base B, constructed by the Algorithm *OpDraw*. Here, n is the number of vertices in G. Let d be the degree of G. Note that, by the definition of feasible drawings, $w(n)$ will be equal to one plus the horizontal separation between the end-points of B.

It is easy to prove using induction that $w(n) = n$ is sufficient. As for the horizontal distances between u and w, between c_{i-1} and c_i (for $1 \le i \le m-1$), between c'_{i-1} and c'_i (for $1 \le i \le s$), between α_{i-1} and α_i (for $1 \le i \le j-1$), between β_{i-1} and β_i (for $1 \le i \le \epsilon-1$), and between β_{i-1} and β_i (for $\epsilon+1 \le i \le \mu$), it is sufficient to set them to be equal to $|H|-1$, $|K_i|-1$, $|K'_i|-1$, $|L_i|-1$, $|M_i|-1$, and $|N_i|-1$, respectively. It is also sufficient to set the distance between the end-points of each edge e on the top or bottom boundary of E, to be equal to $|Q_e|-1$.

As for $h(n)$, first notice that, because G has degree d, $t-(j-1)$ is less than $2d$, and hence, the distance between $\beta_{\epsilon-1}$ and β_ϵ is less than $2d+2$.

Let h' be a function, such that $h'(f) = h(n)$, where f is the number of internal faces in G, i.e., the number of nodes in the dual tree T_G of G.

From the construction of D, we have that:

$$\begin{aligned} h'(f) \le \max\{ & \max_{1\le i\le s}\{h'(|T_{K'_i}|)\}, \max_{1\le i\le \mu-\epsilon}\{h'(|T_{N_i}|)\}, \max_{\textit{edge } e \textit{ on bottom boundary of } E}\{h'(|T_{Q_e}|)\}\} \\ &+ \max\{h'(|T_H|), \max_{1\le i\le m-1}\{h'(|T_{K_i}|)\}, \max_{1\le i\le j-1}\{h'(|T_{L_i}|)\}, \max_{1\le i\le \epsilon-1}\{h'(|T_{M_i}|)\}, \\ &\max_{\textit{edge } e \textit{ on top boundary of } E}\{h'(|T_{Q_e}|)\}\} + O(d), \end{aligned}$$

Since P is a spine of T_G, and

- the dual trees of H, K_i, L_i, M_i, and Q_e (in the case when edge e is on top boundary of E), are either right subtrees of P, or belong to the right subtrees of P, and
- the dual trees of K'_i, N_i, and Q_e (in the case when edge e is on bottom boundary of E), are either left subtrees of P, or belong to the left subtrees of P,

from Lemma 1, it follows that:

$$h'(f) \le \max_{f_1^p + f_2^p \le (1-\delta) f^p}\{h'(f_1) + h'(f_2) + O(d)\}.$$

Using induction, we can show that $h'(f) = O(df^{0.48})$ (see also [2]). Since $f = O(n)$, $h(n) = h'(f) = O(df^{0.48}) = O(dn^{0.48})$.

Theorem 1. *Let G be an outerplanar graph with degree d and n vertices. We can construct a planar straight-line grid drawing of G with area $O(dn^{1.48})$ in $O(n)$ time.*

Proof. Arbitrarily select any edge $e = (u, v)$ on the external face of G, and designate u and v as the poles of G. Let B be any horizontal line-segment with length $n - 1$, such that the end-points of B are grid points. Let δ be any user-defined constant in the range $(0, 0.0004]$. Construct a feasible drawing of G with base B using Algorithm *OpDraw*. From the discussion given above, it follows immediately that the area of the drawing is $O(dn^{1+0.48}) = O(dn^{1.48})$. It is easy to see the algorithm runs in $O(n)$ time.

Corollary 1. *Let G be an outerplanar graph with n vertices and degree d, where $d = o(n^{0.52})$. We can construct a planar straight-line grid drawing of G with $o(n^2)$ area in $O(n)$ time.*

References

1. T. Biedl. Drawing outer-planar graphs in $o(n \log n)$ area. In *Proc. 10th International Symposium on Graph Drawing (GD 2002)*, volume 2528 of *Lecture Notes in Computer Science*, pages 54–65. Springer-Verlag, 2002.
2. T.M. Chan. A near-linear area bound for drawing binary trees. *Algorithmica*, 34(1):1–13, 2002.
3. H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.
4. V. Dujmovic and D.R. Wood. Tree-partitions of k-trees with applications in graph layout. In *Proc. 29th Workshop on Graph Theoretic Concepts in Computer Science (WG '03)*. To appear.
5. S. Felsner, G. Liotta, and S. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. In *Proc. 9th International Symposium on Graph Drawing (GD 2001)*, volume 2265 of *Lecture Notes in Computer Science*, pages 328–342. Springer-Verlag, 2001.
6. A. Garg and A. Rusu. Straight-line drawings of binary trees with linear area and arbitrary aspect ratio. In *Proc. 10th International Symposium on Graph Drawing (GD 2002)*, volume 2528 of *Lecture Notes in Computer Science*, pages 320–331. Springer-Verlag, 2002.
7. C. E. Leiserson. Area-efficient graph layouts (for VLSI). In *Proc. 21st Annu. IEEE Sympos. Found. Comput. Sci.*, pages 270–281, 1980.
8. W. Schnyder. Embedding planar graphs on the grid. In *Proc. 1st ACM-SIAM Sympos. Discrete Algorithms*, pages 138–148, 1990.