View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Elsevier - Publisher Connector

Computational Geometry 45 (2012) 524–533

Contents lists available at SciVerse ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Straight-line drawings of outerplanar graphs in $O(dn \log n)$ area ☆

Fabrizio Frati

Dipartimento di Informatica e Automazione, Università Roma Tre, Italy

ARTICLE INFO

Article history:
Received 19 March 2008
Received in revised form 8 March 2010
Accepted 29 March 2010
Available online 28 January 2012
Communicated by P. Bose and P. Carmi

Keywords:
Graph Drawing
Grid drawings
Straight-line drawings
Minimum area
Outerplanar graphs

ABSTRACT

We show an algorithm for constructing $O(dn \log n)$ area outerplanar straight-line drawings of n-vertex outerplanar graphs with degree d. Also, we settle in the negative a conjecture (Biedl, 2002 [1]) on the area requirements of outerplanar graphs by showing that snowflake graphs admit linear-area drawings.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Constructing small-area straight-line drawings of planar graphs is a classical research topic in Graph Drawing. Ground-breaking works of the end of the 80's have shown that every planar graph admits a planar straight-line drawing on a grid whose size is quadratic in the number of vertices of the graph [2,3]. It turns out that such an area requirement is worst-case the best possible [4,2,5]. Consequently, the problem of finding non-trivial sub-classes of planar graphs that admit sub-quadratic area drawings has been widely investigated. For example, it is known that every n-node tree can be drawn in $O(n \log n)$ area (a simple modification of the *h–v drawing algorithm* in [6]) and that every n-node tree whose degree is $O(\sqrt{n})$ can be drawn in $O(n)$ area [7].

One of the classes of graphs that has attracted more research interest is the one of *outerplanar graphs*. An outerplanar graph is a graph that admits a planar drawing in which all vertices are on the same face. Almost thirty years ago, Dolev and Trickey [8] showed that every n-vertex outerplanar graph whose degree is bounded by four admits a poly-line drawing in $O(n)$ area. The techniques presented in [8] can be modified in order to obtain poly-line drawings of outerplanar graphs with degree d in $O(d^2n)$ area, as pointed out in [1]. More recently, the problem of obtaining minimum-area drawings of outerplanar graphs has been tackled by Biedl [1], who first provided a sub-quadratic area upper bound for poly-line drawings of general outerplanar graphs. Namely, she proved that outerplanar graphs admit poly-line drawings in $O(n \log n)$ area. Moreover, she conjectured that there exists a class of outerplanar graphs called "snowflake graphs" requiring $\Omega(n \log n)$ area in any planar straight-line or poly-line drawing.

Concerning straight-line drawings, in [9] (further published in [10]) Garg and Rusu have shown that every n-vertex outerplanar graph with degree d has a straight-line drawing with $O(dn^{1.48})$ area. The first sub-quadratic area upper bound has been proved in [11] (further published in [12]), where Di Battista and Frati showed that $O(n^{1.48})$ area always suffices for

☆ This work is partially supported by the Italian Ministry of Research, Grant No. RBIP06BZW8, FIRB project "Advanced tracking system in intermodal freight transportation".

E-mail address: frati@dia.uniroma3.it.

0925-7721/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.comgeo.2010.03.007

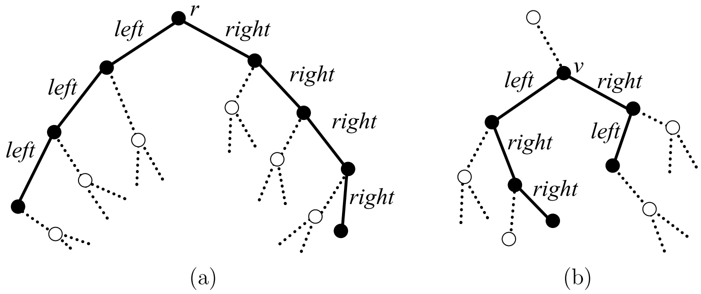

Fig. 1. (a) The leftmost and the rightmost path of an ordered rooted binary tree. (b) The left–right path and the right–left path of a node v in an ordered rooted binary tree.

straight-line drawings of outerplanar graphs. For restricted classes of outerplanar graphs, namely outerplanar graphs whose dual tree has a small diameter [1], "complete outerplanar graphs" [12], and "label-constrained outerplanar graphs" [13], better area bounds are known.

In this paper we show an algorithm for obtaining straight-line drawings of outerplanar graphs in $O(dn\log n)$ area. Clearly, this improves the upper bound on the area requirement of all the outerplanar graphs whose degree is $O(n^{0.48}/\log n)$. Further, we prove that snowflake graphs admit linear-area straight-line drawings, settling in the negative the above cited conjecture appeared in [1].

The rest of the paper is organized as follows. Section 2 contains some preliminaries. Section 3 presents an algorithm for constructing straight-line drawings of outerplanar graphs in $O(dn\log n)$ area. Section 4 presents an algorithm for constructing straight-line drawings of snowflake graphs in $O(n)$ area; in the same section we give conclusions and a conjecture concerning the area requirement of straight-line drawings of outerplanar graphs.

2. Preliminaries

We assume familiarity with Graph Drawing. For basic definitions see also [14].

A *drawing* of a graph is a mapping of each vertex to a point of the plane and of each edge to a Jordan curve between its endpoints. A drawing is *planar* if no two edges cross, but, possibly, at common endpoints. A *grid drawing* is such that all the vertices have integer coordinates. A *straight-line drawing* (resp. a *poly-line drawing*) is such that each edge is represented by a segment (resp. by a sequence of consecutive segments). Clearly, a straight-line drawing of a graph is fully determined by the placement of its vertices. In the following, unless otherwise specified, for *drawing* we always mean planar straight-line grid drawing. The *area* of a drawing is the number of grid points in the smallest rectangle with sides parallel to the axes that covers the drawing completely.

A planar drawing of a graph determines a circular ordering of the edges incident to each vertex. Two drawings of the same graph are *equivalent* if they determine the same circular ordering around each vertex. A *planar embedding* is an equivalence class of planar drawings. A planar drawing partitions the plane into topologically connected regions, called *faces*. The unbounded face is the *outer face*. Two equivalent drawings of the same graph have the same faces. An embedding of a graph G determines its *dual graph*, that is the graph with one vertex per face of G and with one edge between two vertices if the corresponding faces share an edge in G.

An *outerplanar embedding* is a planar embedding in which all the vertices are incident to the same face, say the outer face. *An outerplanar graph* is a graph that admits an outerplanar embedding. The dual graph of an outerplanar embedded graph is a tree when not considering the vertex corresponding to the outer face. A *maximal outerplanar graph* is a graph that admits an outerplanar embedding in which all the faces, except for the outer one, are triangles. The dual graph of a maximal outerplanar embedded graph is a binary tree.

The *degree of a vertex* is the number of edges incident to the vertex. The *degree of a graph* is the maximum degree of one of its vertices. A *binary tree* is a tree such that each node has degree at most three. A *rooted tree* is one with a distinguished node, called *root*. A tree is *ordered* if a left-to-right order of the children of each node is fixed. A drawing Γ of an ordered tree T is *order-preserving* if the order in which the edges are incident to each node in Γ is the same as specified in T.

Let T be an ordered binary tree rooted at node r. Let $T(v)$ denote the subtree of T rooted at node v. The *leftmost path* $L(T)$ (resp. the *rightmost path* $R(T)$) of T is the path $v_0, v_1, \dots, v_k$ such that $v_0 = r$, v_{i+1} is the left child (resp. the right child) of v_i, $\forall i$: $0 \leqslant i \leqslant k-1$, and v_k does not have a left child (resp. a right child). See Fig. 1(a).

The *left–right path* (resp. the *right–left path*) of a node $v \in T$ is the path $v_0, v_1, \dots, v_k$ such that $v_0 = v$, v_1 is the left (resp. right) child of v_0, and $v_1, v_2, \dots, v_k$ is the rightmost (resp. leftmost) path of $T(v_1)$. See Fig. 1(b).

Consider any drawing Γ of T. The *left polygon of the neighbors* $P_l(v)$ (resp. *the right polygon of the neighbors* $P_r(v)$) of a node $v \in T$ is the polygon of the segments representing in Γ the edges of the left–right path (resp. of the right–left path) plus a segment connecting v_k and v_0.

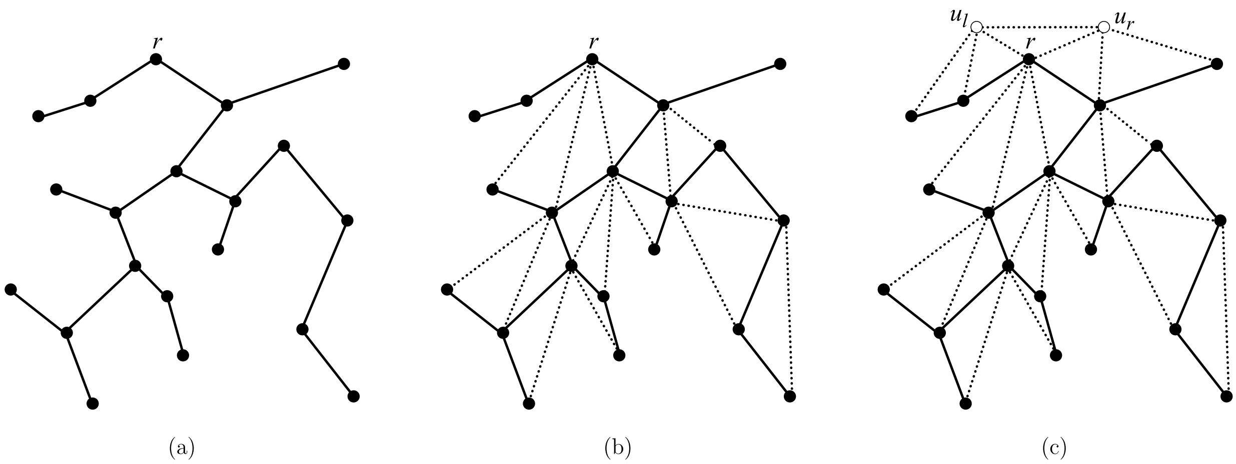

Fig. 2. (a) A star-shaped drawing of a tree T, dual to an outerplanar graph G. (b) A straight-line drawing of the internal subgraph I of G. (c) A straight-line drawing of G.

A planar straight-line order-preserving drawing Γ of an ordered rooted binary tree T is *star-shaped* if:

1. For each node $v \in T$, $P_l(v) = (v, v_1, \ldots, v_k)$ and $P_r(v) = (v, u_1, \ldots, u_p)$ are simple polygons and each segment (v, v_i), with $2 \leqslant i \leqslant k-1$ (resp. (v, u_j), with $2 \leqslant j \leqslant p-1$), belongs to the interior of $P_l(v)$ (resp. of $P_r(v)$), except for its endpoints.
2. For each pair of nodes $v_a, v_b \in T$, each of $P_l(v_a)$ and $P_r(v_a)$ does not intersect $P_l(v_b)$ and $P_r(v_b)$, except, possibly, at common endpoints or at common edges.
3. There exist points p_l and p_r from which it is possible to draw edges to each node of $L(T)$ and to each node of $R(T)$, respectively, and to draw edge (p_l, p_r), without creating crossings with the edges of Γ.

Fig. 2(a) shows an example of a star-shaped drawing of an ordered rooted binary tree.

Let $\mathcal{E}$ be an outerplanar embedding of a maximal outerplanar graph G and let T be the dual binary tree of $\mathcal{E}$. Select any edge (u_l, u_r) on the outer face of $\mathcal{E}$ and root T at the internal face containing (u_l, u_r). We call vertices u_l and u_r *poles* of G and we also call *internal subgraph* the graph obtained by deleting u_l, u_r, and their incident edges from G.

In [11,12] the straight-line drawability of an n-vertex outerplanar graph G was strictly related to the star-shaped drawability of its dual tree T, by means of the following lemma.

Lemma 1. *(See Di Battista and Frati [11,12].) If T admits a star-shaped drawing with $f(n)$ area, then G has an outerplanar straight-line drawing where the area of the drawing of the internal subgraph of G is $f(n)$.*

The proof of the previous lemma is based on the observation that the dual binary tree T of an embedded maximal outerplanar graph G is a subgraph of G itself. The internal subgraph I of G can be obtained by adding to T, for each node $v \in T$, edges connecting v to the nodes of the left–right path of v and of the right–left path of v (see Fig. 2(b)). The outerplanar graph G can be obtained by further adding to I two extra vertices (the poles of G) and the edges connecting the two poles to the nodes of the leftmost and rightmost path of T (see Fig. 2(c)). Such a relationship is stated in the following lemma.

Lemma 2. *(See Di Battista and Frati [11,12].) Let T be an ordered rooted binary tree dual to a maximal outerplanar graph G. Add to T edges connecting each node $v \in T$ to the nodes of its left–right path and to the nodes of its right–left path, and add to T two extra vertices connected to the nodes of the leftmost and rightmost path of T, thus obtaining a graph T^+. Then, $T^+ = G$.*

It can be observed that a star-shaped drawing of T satisfies conditions sufficient for placing the poles of G and drawing the edges necessary to augment T to G, still maintaining a planar drawing. Further, in a star-shaped drawing of T with area $f(n)$, it is sufficient to guarantee a placement for the poles of G not asymptotically increasing $f(n)$ to be able to construct a drawing of G in $O(f(n))$ area.

Further, observe that if a maximal outerplanar graph G has degree d, then, for each node $v \in T$, $L(T(v))$ and $R(T(v))$ have each at most d nodes, as all the nodes in $L(T)$ (resp. all the nodes in $R(T)$) are connected to the same node of G.

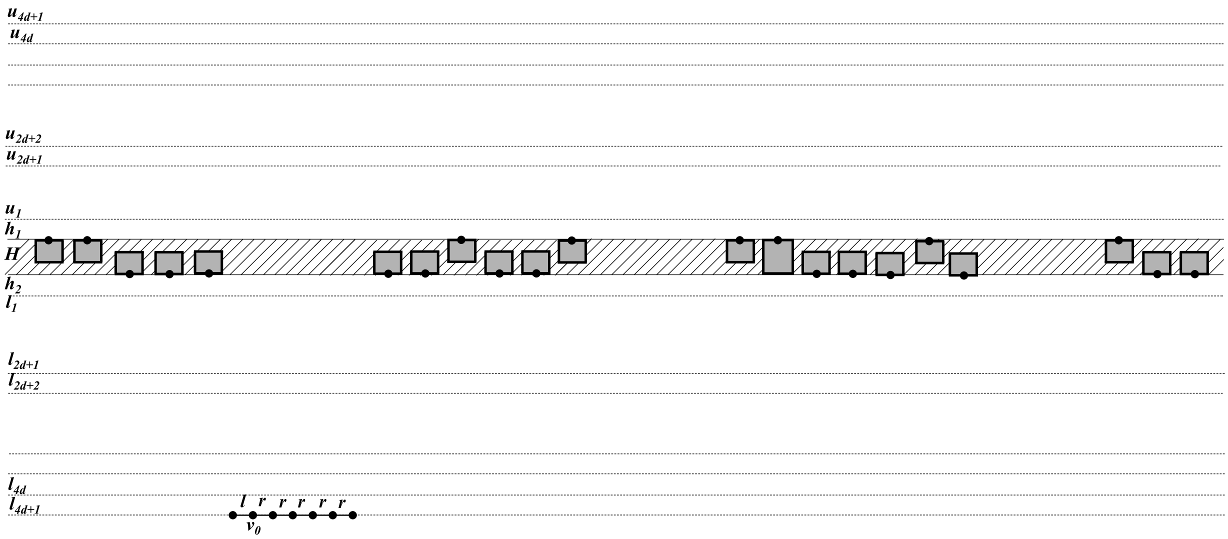

Fig. 3. The strip H, the upper and the lower part of the drawing Γ, the placement of $L(T)$ and $R(T)$ in Γ.

3. The algorithm

In this section we show an algorithm for constructing straight-line outerplanar drawings of outerplanar graphs with degree d in $O(dn\log n)$ area. The algorithm consists of the following steps.

(i) Augment the input outerplanar graph G to a maximal outerplanar graph G'.
(ii) Select any edge (u_l, u_r) incident to the outer face of the outerplanar embedding $\mathcal{E}$ of G' and root the dual binary tree T of $\mathcal{E}$ at the internal face r of $\mathcal{E}$ containing (u_l, u_r); construct a star-shaped drawing Γ of T.
(iii) Insert the poles of G' and the edges that are needed to augment Γ in a drawing Γ' of G'.
(iv) Remove the dummy edges inserted during step (i) to obtain a drawing of G.

Step (i) can be performed by means of an algorithm described in [15], where it is shown how to augment an outerplanar graph to maximal by inserting dummy edges that do not asymptotically increase the degree of the graph.

Now we describe how to perform step (ii), that is, we describe how to construct a star-shaped drawing Γ of T. The outline of such a construction is as follows. A path S is removed from T, together with the edges incident to the vertices of S. The subtrees that are disconnected from the removal of S are recursively drawn. Path S is chosen so that each one of such subtrees is "small", that is, has at most $n/2$ nodes. The drawings of the recursively drawn subtrees are horizontally aligned, namely they are all contained in a horizontal strip H. The vertical extension of such a strip is given by the height of the highest drawing of a subtree recursively drawn. The nodes of S are drawn in the *upper part* and in the *lower part* of the drawing, that is, in the $4d+1$ horizontal grid lines above and below H, respectively. In particular, S is partitioned into subpaths, and each subpath "cuts" H, that is, the subpath is drawn in part above and in part below H. The drawing is constructed to satisfy the visibility properties of a star-shaped drawing. In particular, the leftmost and the rightmost path of T are placed on the lowest line intersecting the drawing.

Denote by h_1 and h_2 the horizontal grid lines delimiting H with h_1 above h_2 at a vertical distance that will be determined later. The $4d+1$ horizontal grid lines above h_1 (resp. below h_2), that compose the upper part (resp. the lower part) of the drawing, are labeled by $u_1, u_2, \dots, u_{4d+1}$ (resp. by $l_1, l_2, \dots, l_{4d+1}$) from the lowest to the highest (resp. from the highest to the lowest). All the nodes of $L(T)$ and of $R(T)$ lie on l_{4d+1}. See Fig. 3.

Assume T is rooted at any node v_0 of degree at most two. Select a *spine* $S = (v_0, v_1, \dots, v_m)$ in T, that is a path such that, for $1 \leqslant i \leqslant k$, v_i is the child of v_{i-1} that is root of the heaviest subtree of v_{i-1}. The nodes of T belonging to S (not belonging to S) are *spine nodes* (resp. *non-spine nodes*). We call *left edge* (*right edge*) an edge (v_{i-1}, v_i) of S such that v_i is the left child of v_{i-1} (resp. the right child of v_{i-1}). We prove that each non-spine node u_i child of a spine node v_i is root of a subtree with no more than $n/2$ nodes. Let v_{i+1} be the spine node child of v_i (notice that v_{i+1} exists, otherwise u_i would be chosen as a spine node). Suppose, for a contradiction, that $T(u_i)$ has at least $1 + n/2$ nodes. However, by the way in which the spine is chosen, $T(v_{i+1})$ has also at least $1 + n/2$ nodes. Hence, T would have more than n nodes, thus providing a contradiction.

Path S is partitioned into vertex-disjoint subpaths $S_0, S_1, \dots, S_q$, so that, for $0 \leqslant j \leqslant q$, path $S_j = (v^j_k, v^j_{k+1}, \dots, v^j_l, v^j_{l+1}, \dots, v^j_{f-1}, v^j_f)$ is defined as follows:

- $v_k^0 = v_0$ and, for $1 \leqslant j \leqslant q$, v_k^j is the node after v_f^{j-1} in S.
- If (v_k^j, v_{k+1}^j) is a right edge, then let v_l^j be the first node after v_k^j in S such that (v_l^j, v_{l+1}^j) is a left edge.
  - If (v_{l+1}^j, v_{l+2}^j) is a left edge then let v_f^j be the first node after v_l^j in S such that (v_f^j, v_{f+1}^j) is a right edge.
  - If (v_{l+1}^j, v_{l+2}^j) is a right edge then let v_f^j be the first node after v_l^j in S such that (v_f^j, v_{f+1}^j) is a left edge.
- If (v_k^j, v_{k+1}^j) is a left edge, then let v_l^j be the first node after v_k^j in S such that (v_l^j, v_{l+1}^j) is a right edge.
  - If (v_{l+1}^j, v_{l+2}^j) is a right edge then let v_f^j be the first node after v_l^j in S such that (v_f^j, v_{f+1}^j) is a left edge.
  - If (v_{l+1}^j, v_{l+2}^j) is a left edge then let v_f^j be the first node after v_l^j in S such that (v_f^j, v_{f+1}^j) is a right edge.

Notice that S_q could have no vertex v_l^q or v_f^q if the spine ends before a left edge or a right edge is encountered. Roughly speaking, a path S_j starts from the vertex v_k^j of S following the one where path S_{j-1} ends; further, S_j ends where the second or the third alternation among left and right edges of S is found starting from v_k^j. In particular, S_j ends at the third alternation when the second alternation comes immediately after the first one, i.e., when S contains a sequence (left edge, right edge, left edge) or a sequence (right edge, left edge, right edge) providing the first and the second alternation among left and right edges of S starting from v_k^j.

Tree T is also subdivided into subtrees: For $0 \leqslant j \leqslant q$, tree T_j is the subtree of T induced by the nodes in S_j and the nodes in the subtrees rooted at non-spine nodes children of spine nodes in S_j.

Now we show how to construct a drawing Γ_j of each T_j, for $0 \leqslant j \leqslant q$. First, we show how to draw path S_j together with some other nodes of T_j. We distinguish eight cases, based on whether:

- j is even (Cases 1–4) or odd (Cases 5–8).
- (v_k^j, v_{k+1}^j) is a right edge (Cases 1, 2, 5, 6) or a left edge (Cases 3, 4, 7, 8).
- (v_{l+1}^j, v_{l+2}^j) is a right edge (Cases 1, 3, 5, 7) or a left edge (Cases 2, 4, 6, 8).

In the cases in which j is even and $j > 0$ (resp. $j = 0$), draw $L(T_j)$ and $R(T_j)$ on l_{2d+1} (resp. on l_{4d+1}) so that the each node of $L(T_j)$ is one unit to the right of its left child and each node of $R(T_j)$ is one unit to the left of its right child. In the cases in which j is odd, draw $L(T_j)$ and $R(T_j)$ on u_{2d+1} so that the each node of $L(T_j)$ is one unit to the left of its left child and each node of $R(T_j)$ is one unit to the right of its right child. Denote by h_l the vertical grid line passing through v_l^j, and denote by h_{l+1}, h_{l+2}, h_{l-1}, and h_{l-2} the vertical grid lines one unit to the right, two units to the right, one unit to the left, and two units to the left of h_l, respectively.

Case 1. Draw v_f^j at the intersection between h_{l+1} and u_{2d+2}; draw $R(T(v_f^j))$ on h_{l+1}, with any node one unit above its right child. Draw v_{f-1}^j at the intersection between h_{l-2} and u_{4d+1}. Draw $R(T(v_{l+1}^j))$ until v_{f-1}^j on h_{l-2}, with any node one unit below its right child; draw $L(T(v_{l+1}^j))$ on h_{l-2}, with any node one unit above its left child.

Case 2. Draw v_f^j at the intersection between h_{l-2} and u_{4d+1}; draw $L(T(v_f^j))$ on h_{l-2}, with any node one unit above its left child. Draw v_{f-1}^j at the intersection between h_{l+1} and u_{2d+2}. Draw $L(T(v_{l+1}^j))$ until v_{f-1}^j on h_{l+1}, with any node one unit below its left child; draw $R(T(v_{l+1}^j))$ on h_{l+1}, with any node one unit above its right child.

Case 3. Draw v_f^j at the intersection between h_{l+2} and u_{2d+2}; draw $R(T(v_f^j))$ on h_{l+2}, with any node one unit above its right child. Draw v_{f-1}^j at the intersection between h_{l-1} and u_{4d+1}. Draw $R(T(v_{l+1}^j))$ until v_{f-1}^j on h_{l-1}, with any node one unit below its right child; draw $L(T(v_{l+1}^j))$ on h_{l-1}, with any node one unit above its left child.

Case 4. Draw v_f^j at the intersection between h_{l-1} and u_{4d+1}; draw $L(T(v_f^j))$ on h_{l-1}, with any node one unit above its left child. Draw v_{f-1}^j at the intersection between h_{l+2} and u_{2d+2}. Draw $L(T(v_{l+1}^j))$ until v_{f-1}^j on h_{l+2}, with any node one unit below its left child; draw $R(T(v_{l+1}^j))$ on h_{l+2}, with any node one unit above its right child.

Case 5. Draw v_f^j at the intersection between h_{l-1} and l_{4d+1}; draw $R(T(v_f^j))$ on h_{l-1}, with any node one unit below its right child. Draw v_{f-1}^j at the intersection between h_{l+2} and l_{2d+2}. Draw $R(T(v_{l+1}^j))$ until v_{f-1}^j on h_{l+2}, with any node one unit above its right child; draw $L(T(v_{l+1}^j))$ on h_{l+2}, with any node one unit below its left child.

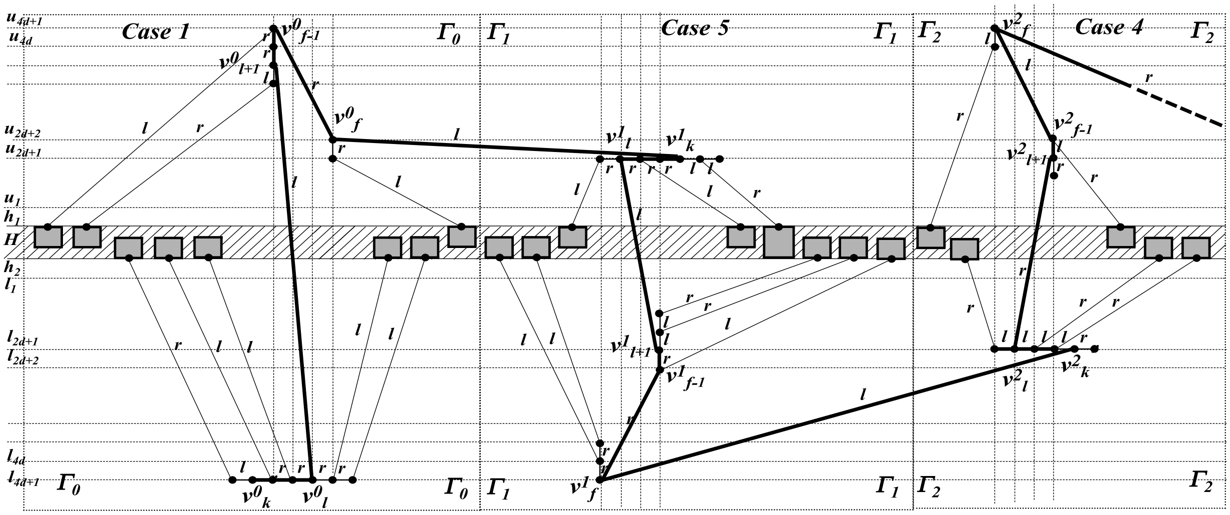

Fig. 4. Thick segments represent the edges of S. An edge between a node and its left child (resp. between a node and its right child) is labeled by l (resp. by r).

Case 6. Draw v^j_f at the intersection between h_{l+2} and l_{2d+2}; draw $L(T(v^j_f))$ on h_{l+2}, with any node one unit below its left child. Draw v^j_{f-1} at the intersection between h_{l-1} and l_{4d+1}. Draw $L(T(v^j_{l+1}))$ until v^j_{f-1} on h_{l-1}, with any node one unit above its left child; draw $R(T(v^j_{l+1}))$ on h_{l-1}, with any node one unit below its right child.

Case 7. Draw v^j_f at the intersection between h_{l-2} and l_{4d+1}; draw $R(T(v^j_f))$ on h_{l-2}, with any node one unit below its right child. Draw v^j_{f-1} at the intersection between h_{l+1} and l_{2d+2}. Draw $R(T(v^j_{l+1}))$ until v^j_{f-1} on h_{l+1}, with any node one unit above its right child; draw $L(T(v^j_{l+1}))$ on h_{l+1}, with any node one unit below its left child.

Case 8. Draw v^j_f at the intersection between h_{l+1} and l_{2d+2}; draw $L(T(v^j_f))$ on h_{l+1}, with any node one unit below its left child. Draw v^j_{f-1} at the intersection between h_{l-2} and l_{4d+1}. Draw $L(T(v^j_{l+1}))$ until v^j_{f-1} on h_{l-2}, with any node one unit above its left child; draw $R(T(v^j_{l+1}))$ on h_{l-2}, with any node one unit below its right child.

Fig. 4 shows the algorithm's construction. Paths S_0, S_1, and S_2 are drawn as in Cases 1, 5, and 4, respectively.

For each T_j, with $0 \leqslant j \leqslant q$, recursively construct a drawing of each subtree rooted at a node of T_j that has not been already drawn and that is child of a node of T_j that has been already drawn. Let $h_{\max}$ be the maximum among the heights of the drawings of the subtrees recursively drawn. Set the distance between h_1 and h_2 to be $h_{\max} - 1$, that is, H consists of $h_{\max}$ horizontal grid lines.

For each T_j, with $0 \leqslant j \leqslant q$ and j even, construct a drawing Γ_j starting from the already constructed drawing of S_j, as follows:

- Consider the drawings of the subtrees rooted at non-already drawn nodes of T_j that are children of nodes belonging to $L(T_j)$ or to $R(T_j)$, and whose parents are placed to the right of v^j_l. Place such drawings in the order induced by their parents on l_{4d+1} or on l_{2d+1}, at one unit of horizontal distance between them, with the leftmost vertical line intersecting the leftmost drawing one unit to the right of the last node of $R(T_j)$, and with their leftmost and rightmost paths on h_1.
- Consider the drawings of the subtrees rooted at non-already drawn nodes of T_j that are children of nodes belonging to $L(T_j)$ or to $R(T_j)$, and whose parents are placed to the left of v^j_l. Place such drawings in the order induced by their parents on l_{4d+1} or on l_{2d+1}, at one unit of horizontal distance between them, with the rightmost vertical line intersecting the rightmost drawing one unit to the left of the last node of $L(T_j)$, and with their leftmost and rightmost paths on h_1.
- Consider the drawings of the subtrees rooted at non-already drawn nodes of T_j that are children of nodes already drawn in the upper part of the drawing. Rotate such drawings of π radiants and place them so that their leftmost and rightmost paths lie on h_2, so that the drawings of the subtrees rooted at children of nodes drawn on h_{l+1} or on h_{l+2} (on h_{l-1} or on h_{l-2}) are placed to the right (resp. to the left) of the drawing constructed up to now, at one unit of horizontal distance in the order induced by their parents in $L(T(v^j_{l+1}))$ or in $R(T(v^j_{l+1}))$.

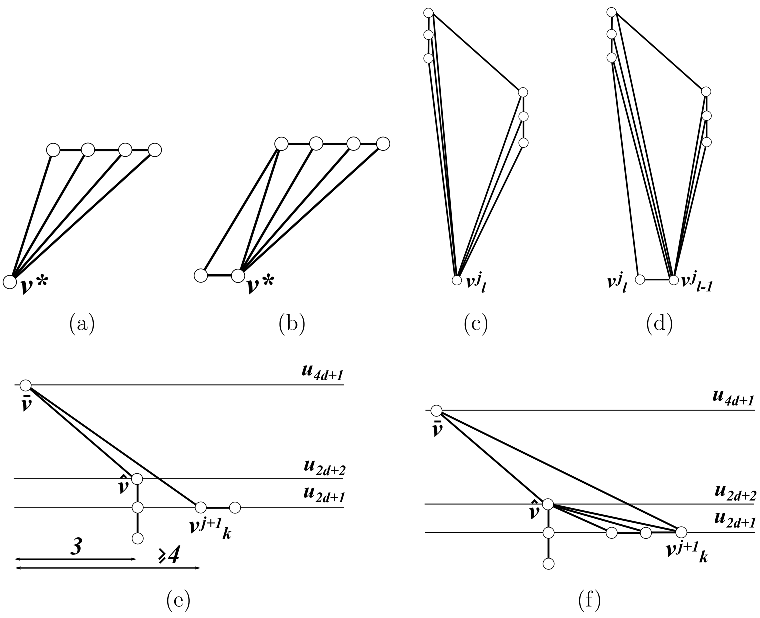

Fig. 5. Illustrations for the proof of Lemma 3.

If j is odd, a drawing Γ_j of T_j can be constructed analogously.

Now place all the Γ_j's together, starting from Γ_0, and iteratively adding Γ_j, for $j = 1, \ldots, m$, so that the leftmost vertical line intersecting Γ_j is one unit to the right of the rightmost vertical line intersecting Γ_{j-1}.

We prove that the resulting drawing Γ is a star-shaped drawing of T.

Lemma 3. *Γ is a star-shaped drawing of T.*

Proof. We show that, given an ordered rooted binary tree T, dual to a maximal outerplanar graph G, the above described algorithm constructs a star-shaped drawing Γ of T.

First, observe that the claim that all the subtrees recursively drawn are contained inside H holds, since the distance between h_1 and h_2 is set equal to the height of the highest subtree recursively drawn.

Second, all the nodes directly drawn are in the upper and lower part of Γ. Namely, we have that: (i) by construction such nodes are never placed above u_{4d+1} or below l_{4d+1} and (ii) if such nodes are placed in H, then it's easy to deduce by the algorithm's construction that there exists a leftmost or a rightmost path of a subtree of T whose length is greater than d; however, this would imply that there exists a vertex of G whose degree is greater than d, since all the nodes of a leftmost or rightmost path of a subtree of T are neighbors of the same node of G.

Drawing Γ is straight-line and order-preserving by construction. We prove that Γ is planar and satisfies conditions 1 and 2 of a star-shaped drawing. However, the planarity of Γ comes straightforwardly by the algorithm's construction, with the exception of the proof that each edge (v^j_f, v^{j+1}_k), with $0 \leqslant j \leqslant q-1$, does not cross other edges.

For each node v^* lying in the upper part of the drawing, except for v^j_{l-1} and v^j_l, all the nodes of $P_l(v^*)$ and of $P_r(v^*)$ are placed either on h_1 or in the upper part of Γ. Analogously, for each node v^* lying in the lower part of the drawing, except for v^j_{l-1} and v^j_l, all the nodes of $P_l(v^*)$ and of $P_r(v^*)$ are placed either on h_2 or in the lower part of Γ. Hence, all the edges incident to nodes different from v^j_{l-1} and v^j_l and lying in the upper part and in the lower part of the drawing do not cut H. It follows that, provided a proof that the edges connecting v^j_{l-1} and v^j_l to the nodes on $P_l(v^j_{l-1})$ and $P_r(v^j_{l-1})$, and to the nodes on $P_l(v^j_l)$ and $P_r(v^j_l)$, respectively, do not cause crossings, the proof that Γ is planar and satisfies conditions 1 and 2 of a star-shaped drawing can be done separately for each recursive step of the algorithm.

Consider a subtree T_j of T, as defined in the algorithm's description. For each node v^* of T_j different from v^j_{l-1}, v^j_l, v^j_{f-1}, and v^j_f, one between $P_l(v^*)$ and $P_r(v^*)$ has nodes placed either all on h_1 or all on h_2, except for v^* (see Fig. 5(a)); the other one between $P_l(v^*)$ and $P_r(v^*)$ has one node placed on the same horizontal or vertical line of v^* and the other nodes either all on h_1 or all on h_2 (see Fig. 5(b)). Hence, straight-lines can be drawn from v^* to the nodes of $P_l(v^*)$ and of $P_r(v^*)$ without creating crossings in Γ.

Concerning v^j_{l-1} (resp. v^j_l), one between $P_l(v^j_{l-1})$ and $P_r(v^j_{l-1})$ (resp. one between $P_l(v^j_l)$ and $P_r(v^j_l)$) has nodes all on h_1 or all on h_2, except for v^j_{l-1} (resp. except for v^j_l and its child that lies on the same horizontal line of v^j_l) and the other one between $P_l(v^j_{l-1})$ and $P_r(v^j_{l-1})$ (resp. between $P_l(v^j_l)$ and $P_r(v^j_l)$) is a convex polygon $\mathcal{P}$ (see Figs. 5(c) and 5(d)) cutting H. However, no recursively drawn subtree has intersection with the smallest vertical strip containing $\mathcal{P}$ (such a strip is delimited either by lines h_{l-2} and h_{l+1} or by lines h_{l-1} and h_{l+2}). Hence, straight-lines can be drawn from v^j_{l-1} (from v^j_l) to the nodes of $P_l(v^j_{l-1})$ and of $P_r(v^j_{l-1})$ (resp. of $P_l(v^j_l)$ and of $P_r(v^j_l)$) without creating crossings in Γ.

Finally, consider nodes v^j_{f-1} and v^j_f. One out of v^j_{f-1} and v^j_f, say $\bar{v}$, is placed on u_{4d+1} or on l_{4d+1}, while the other one, say $\hat{v}$, is placed on u_{2d+2} or on l_{2d+2}. One between $P_l(\bar{v})$ and $P_r(\bar{v})$ has nodes all on h_1 or all on h_2, except for $\bar{v}$, while the other one has nodes all on u_{2d+1} or on l_{2d+1}, except for $\bar{v}$ and, eventually, for $\hat{v}$. To prove that $\bar{v}$ is visible from the nodes of $P_l(\bar{v})$ and $P_r(\bar{v})$, we claim that the slope of the edge connecting $\bar{v}$ to v^{j+1}_k is less than the one of the edge connecting $\bar{v}$ to $\hat{v}$ (see Fig. 5(e)). Namely, the horizontal distance between $\bar{v}$ and v^{j+1}_k is at least 4 and the vertical distance between $\bar{v}$ and v^{j+1}_k is exactly $2d$. Further, the horizontal distance between $\bar{v}$ and $\hat{v}$ is exactly 3 and the vertical distance between $\bar{v}$ and $\hat{v}$ is exactly $2d-1$, and so the slope of $(\bar{v}, v^{j+1}_k)$ is less than or equal to $\frac{2d}{4}$, while the one of $(\bar{v}, \hat{v})$ is $\frac{2d-1}{3}$. We have that $\frac{2d}{4} < \frac{2d-1}{3}$ if and only if $d > 2$ that is always satisfied considering maximal outerplanar graphs with at least 3 vertices. Hence, straight-lines can be drawn from $\bar{v}$ to the nodes of $P_l(\bar{v})$ and of $P_r(\bar{v})$ without creating crossings in Γ. Concerning $\hat{v}$, the nodes of one between $P_l(\hat{v})$ and $P_r(\hat{v})$ lie all on h_1 or all on h_2, except for $\hat{v}$ and eventually for its child that lies on the same vertical line of $\hat{v}$. The nodes of the other one between $P_l(\hat{v})$ and $P_r(\hat{v})$ lie all on u_{2d+1} or all on l_{2d+1}, except for $\hat{v}$ and eventually for $\bar{v}$, that we have already proved to be visible from $\hat{v}$. Hence, straight-lines can be drawn from $\hat{v}$ to the nodes of $P_l(\hat{v})$ and of $P_r(\hat{v})$ without creating crossings in Γ (see Fig. 5(f)).

Concerning condition 3, $L(T)$ and $R(T)$ lie on the bottommost line l_{4d+1} intersecting Γ. Hence, placing the poles u_l and u_r of G one unit below l_{4d+1} allows to draw edges from u_l and u_r to the nodes of $L(T)$ and $R(T)$ without creating crossings with Γ. □

Let's analyze the area requirement of Γ. The width of Γ is clearly $O(n)$. The height of Γ is the sum of the heights of H, of the upper part, and of the lower part of Γ. The height of H is equal to the height of the highest subtree of T recursively drawn; by definition of S, each subtree recursively drawn has at most $n/2$ nodes. Denoting by $H(n)$ the maximum height of a drawing of an n-node tree T constructed by the algorithm, we get: $H(n) \leqslant (4d+1) + (4d+1) + H(n/2) = O(d) + H(n/2) = O(d \log n)$.

Place the poles of G' on the horizontal line one unit below v_0, at one horizontal unit distance from each other, and so that one pole is on the same vertical line of v_0. Notice that this placement doesn't asymptotically increase the area of Γ. Finally, the edges necessary to augment Γ in a drawing of G' can be inserted and the dummy edges inserted in step (i) of the algorithm can be removed obtaining a drawing of G. Fig. 6 shows an example of application of the above described algorithm.

Theorem 1. *Any n-vertex outerplanar graph of degree d has a straight-line outerplanar drawing in $O(dn \log n)$ area.*

Straightforwardly, we obtain the following:

Corollary 1. *Any n-vertex outerplanar graph with constant degree has a straight-line outerplanar drawing in $O(n \log n)$ area.*

4. Conclusions and open problems

In this paper we have shown that $O(dn \log n)$ area is always achievable for straight-line drawings of outerplanar graphs. The following problems, however, still remain widely open:

Problem 1. Which are the asymptotic bounds for the area requirements of straight-line and poly-line planar drawings of outerplanar graphs?

While the research efforts on the determination of the area requirements of straight-line and poly-line drawings of outerplanar graphs have produced many algorithms (and consequently some upper bounds), no lower bound better than the trivial $\Omega(n)$ is known. In [1] Biedl conjectured an $\Omega(n \log n)$ lower bound on the area requirement of straight-line and poly-line drawings of outerplanar graphs. More precisely, she exhibited a class of outerplanar graphs, the *snowflake graphs* shown in Fig. 7(a), for which she claimed:

Conjecture 1. *(See Biedl [1].) Any poly-line drawing of the snowflake graph has $\Omega(n \log n)$ area.*

It can be observed that an n-vertex snowflake graph G is composed of three identical $O(n)$-vertex *complete outerplanar graphs*, defined in [11,12] as the outerplanar graphs whose dual graphs are complete binary trees. For a complete outerplanar graph O with n vertices, an $O(\sqrt{n}) \times O(\sqrt{n})$ area algorithm has been presented in [11,12]. Define the *height* of O to be the

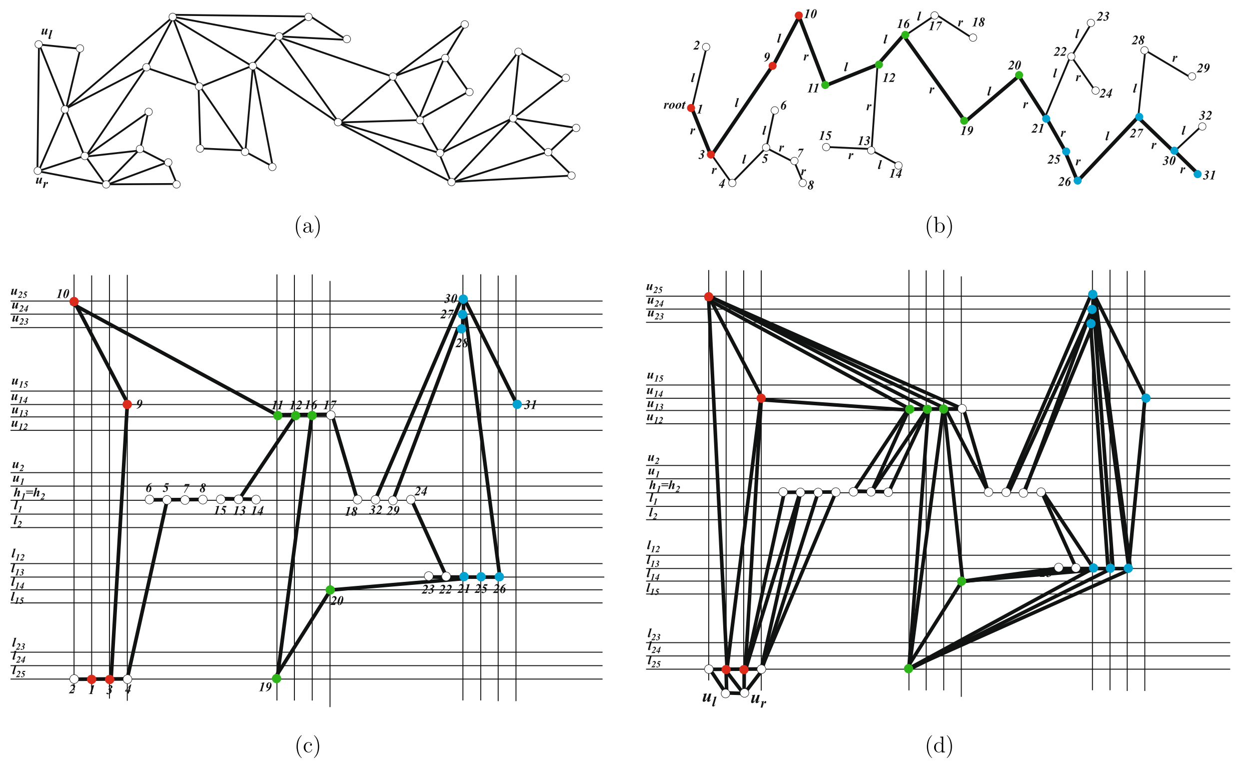

Fig. 6. (a) A maximal outerplanar graph G with $n = 34$ vertices and degree $d = 6$. The poles of G are labeled by u_l and u_r. (b) The dual binary tree T of G. Edges labeled by l (by r) are between a node and its left (resp. right) child. Thick edges show the spine S. Red, green, and blue vertices show subpaths S_0, S_1, and S_2 of S, respectively. Cases 2, 8, and 1 have to be applied to draw S_0, S_1, and S_2, respectively. (c) The star-shaped drawing Γ of T constructed by the algorithm shown in Section 3. (d) The outerplanar drawing of G obtained by augmenting Γ with the poles of G and with extra edges. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

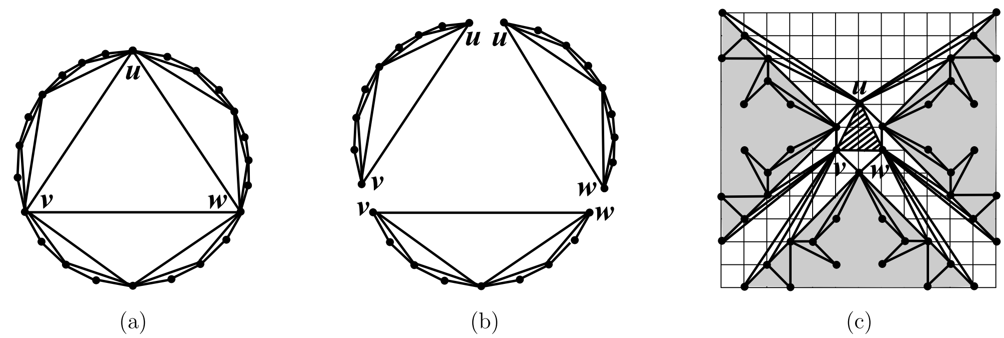

Fig. 7. (a) A snowflake graph. (b) A snowflake graph subdivided into three complete outerplanar graphs. (c) Drawing snowflake graphs in linear area. The shaded regions correspond to the three copies of the drawing of the internal subgraph of a complete outerplanar graph constructed by the algorithm in [11,12].

number of vertices in a path from the root to a leaf in the complete binary tree that is the dual of O. Then, the algorithm presented in [11,12], when the height of O is even, constructs a drawing of the internal subgraph I of O in which the vertices of I that are connected to the poles of O lie on two half-lines l_1 and l_2 starting at a common endpoint and having slopes $-\pi/4$ and $\pi/4$. The entire drawing of I lies in the wedge with angle $\pi/2$ delimited by l_1 and l_2. Hence, consider a complete outerplanar graph O with height h even and consider three copies of the drawing of I constructed by the algorithm in [11,12]. Rotate such copies of 0, π, and $3\pi/2$, respectively, and place the drawings together by inserting the three vertices that are poles for the three complete outerplanar graphs. This results into an $O(\sqrt{n}) \times O(\sqrt{n})$ area straight-line drawing of the snowflake graph G (see Fig. 7(c)). If the height of O is odd, then: (i) Augment G to a snowflake graph G' that is composed of three $O(n)$-vertex complete outerplanar graphs O' whose height is even, (ii) apply the previously described algorithm to draw G', and (iii) remove the inserted dummy vertices and edges. Observe that a snowflake graph G' exists with the required properties and with less than $2n$ vertices.

We would like to point up that all the known algorithms for constructing straight-line drawings of general outerplanar graphs try to minimize the extension of the drawing in only one coordinate dimension, while allowing the other dimension to be $O(n)$. However, we believe that $O(n \log n)$ area cannot be achieved by squeezing the drawing in only one dimension and that hence a compaction in both dimensions (or a proof that there exist n-vertex outerplanar graphs for which one dimension is $\Omega(n)$) should be pursued.

Conjecture 2. *There exist n-vertex outerplanar graphs for which, for any straight-line drawing in which the longest side of the bounding-box is $O(n)$, the smallest side of the bounding-box is $\omega(\log n)$.*

We notice that there exist outerplanar graphs such that both the width and the height of any grid drawing are $\Omega(\log n)$ (e.g., outerplanar graphs containing a complete ternary tree, that is known to require $\Omega(\log n)$ length in both directions [16,17], as a subgraph).

Acknowledgement

Thanks to Giuseppe Di Battista for very useful comments and suggestions.

References

[1] T.C. Biedl, Drawing outer-planar graphs in $O(n \log n)$ area, in: M.T. Goodrich, S.G. Kobourov (Eds.), Graph Drawing (GD'02), in: LNCS, vol. 2528, Springer, 2002, pp. 54–65.
[2] H. de Fraysseix, J. Pach, R. Pollack, How to draw a planar graph on a grid, Combinatorica 10 (1990) 41–51.
[3] W. Schnyder, Embedding planar graphs on the grid, in: Symposium on Discrete Algorithms (SODA'90), 1990, pp. 138–148.
[4] D. Dolev, F.T. Leighton, H. Trickey, Planar embedding of planar graphs, in: Advances in Computing Research, vol. 2: VLSI Theory, 1984.
[5] F. Frati, M. Patrignani, A note on minimum area straight-line drawings of planar graphs, in: S.H. Hong, T. Nishizeki (Eds.), Graph Drawing (GD'07), in: LNCS, vol. 4875, Springer, 2008, pp. 339–344.
[6] P. Crescenzi, G.D. Battista, A. Piperno, A note on optimal area algorithms for upward drawings of binary trees, Computational Geometry: Theory and Applications 2 (1992) 187–200.
[7] A. Garg, A. Rusu, Straight-line drawings of general trees with linear area and arbitrary aspect ratio, in: V. Kumar, M.L. Gavrilova, C.J.K. Tan, P. L'Ecuyer (Eds.), Computational Science and its Applications (ICCSA'03), in: LNCS, vol. 2669, Springer, 2003, pp. 876–885.
[8] D. Dolev, H.W. Trickey, On linear area embedding of planar graphs, Technical Report, Stanford University, Stanford, USA, 1981.
[9] A. Garg, A. Rusu, Area-efficient drawings of outerplanar graphs, in: G. Liotta (Ed.), Graph Drawing (GD'03), in: LNCS, vol. 2912, Springer, 2004, pp. 129–134.
[10] A. Garg, A. Rusu, Area-efficient planar straight-line drawings of outerplanar graphs, Discrete Applied Mathematics 155 (2007) 1116–1140.
[11] G. Di Battista, F. Frati, Small area drawings of outerplanar graphs, in: P. Healy, N.S. Nikolov (Eds.), Graph Drawing (GD'05), in: LNCS, vol. 3843, Springer, 2005, pp. 89–100.
[12] G. Di Battista, F. Frati, Small area drawings of outerplanar graphs, Algorithmica 54 (2009) 25–53.
[13] M.R. Karim, M.J. Alam, M.S. Rahman, Straight-line grid drawings of label-constrained outerplanar graphs with $O(n \log n)$ area (extended abstract), in: S. Das, R. Uehara (Eds.), Workshop on Algorithms and Computation (WALCOM'09), in: LNCS, vol. 5431, Springer, 2009, pp. 310–321.
[14] G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis, Graph Drawing, Prentice–Hall, Upper Saddle River, NJ, 1999.
[15] G. Kant, H.L. Bodlaender, Triangulating planar graphs while minimizing the maximum degree, Information and Computation 135 (1997) 1–14.
[16] S. Felsner, G. Liotta, S.K. Wismath, Straight-line drawings on restricted integer grids in two and three dimensions, Journal of Graph Algorithms & Applications 7 (2003) 363–398.
[17] M. Suderman, Pathwidth and layered drawings of trees, International Journal of Computational Geometry & Applications 14 (2004) 203–225.