Straight-line Drawings of Outerplanar Graphs in O(dn log n) Area

Fabrizio Frati*

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 [1] on the area requirement of outerplanar graphs by showing that snowflake graphs admit linear area drawings.

1 Introduction

Almost thirty years ago, Leiserson [7] and Dolev and Trickey [4] independently showed that every n-vertex outerplanar graph whose degree is bounded by 4 admits a poly-line drawing in $\Theta(n)$ area. The techniques presented in [7, 4] can be modified in order to obtain poly-line drawings of outerplanar graphs with degree d in $O(d^2 n)$ area, as pointed out in [1]. More recently, the problem of obtaining minimum area drawings of outerplanar graphs has been tackled by Biedl, who in [1] provided an $O(n \log n)$ area upper bound for poly-line drawings of general outerplanar graphs. 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 [5] Garg and Rusu have shown that every n-vertex outerplanar graph has a straight-line drawing with $O(dn^{1.48})$ area. However, a sub-quadratic area bound has been proved two years ago in [3], where Di Battista and Frati showed that for straight-line drawings of general outerplanar graphs $O(n^{1.48})$ area always suffices.

In Section 3 we provide an algorithm for obtaining straight-line drawings of outerplanar graphs in $O(dn \log n)$ area. Further, in Section 4 we show that snowflake graphs admit linear area drawings, settling in the negative the above cited conjecture appeared in [1]. 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 [2].

A *planar straight-line grid drawing* of a graph is such that each vertex is mapped to a point with integer coordinates, each edge is represented by a segment between its endpoints and no two edges cross, but, possibly, at common endpoints. In the following, unless otherwise specified, for *drawing* we always mean planar straight-line grid drawing. Clearly, a straight-line drawing of a graph G is fully determined by the placement of the vertices of G. 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 .

An *embedding* of a graph G is an ordering of the edges incident to each vertex. An embedding of 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 of a graph in which all vertices are incident to the same face, say the *external 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 external face. A *maximal outerplanar graph* is a graph that admits an outerplanar embedding in which all faces, but for the external 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.

Let T be a binary tree rooted at node r. Let $T(v)$ denote the subtree of T rooted at node v. The *leftmost path* $L(T)$ (the *rightmost path* $R(T)$) of T is 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, $\forall i$ such that $0 \leq i \leq k-1$, and v_k doesn't have a left child (resp. a right child). The *left-right* (*right-left*) *path* of a node $v \in T$ is 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 $T(v_1)$. Consider any drawing Γ of T. The *left polygon of the neighbors* $P_l(v)$ (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 (resp. right-left) path plus a segment connecting v_k and v_0.

A planar straight-line drawing Γ of a binary tree T is *star-shaped* if (1) it's order-preserving, i.e. the order of the children is the same of one fixed in advance; (2) for each node $v \in T$, $P_l(v) = (v, v_1, \dots, v_k)$ and $P_r(v) = (v, u_1, \dots, u_p)$ are simple polygons and each segment (v, v_i), with $2 \leq i \leq k-1$ (resp. (v, u_j), with $2 \leq j \leq p-1$), belongs to the interior of $P_l(v)$ (resp. $P_r(v)$), but for its endpoints; (3) for each pair of nodes $v_a, v_b \in T$, each of $P_l(v_a)$ and $P_r(v_a)$ doesn't intersect $P_l(v_b)$ and $P_r(v_b)$, but, possibly, at common endpoints or at common edges; and (4) there exist points p_l and p_r from which it's possible to draw edges to each node of

*Dipartimento di Informatica e Automazione, Università di Roma Tre, frati@dia.uniroma3.it Work partially supported by MUR under Project MAINSTREAM Algorithms for Massive Information Structures and Data Streams.

$L(T)$ and to each node of $R(T)$, respectively, and to draw edge (p_l, p_r), without creating crossings with the edges of Γ.

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 external 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 [3] 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 *[3] 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 its internal subgraph is $f(n)$.*

The proof of the previous lemma is based on a bijection that can be established between the nodes of T and the vertices of G, but for its poles. Such a bijection allows to draw G by augmenting the star-shaped drawing of T with two extra points (representing the poles of G) and with some extra straight-line edges. Hence, 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.

3 The algorithm

The outline of the algorithm is as follows: (i) augment the input outerplanar graph G to a maximal outerplanar graph G'; (ii) select any edge (u_l, u_r) on the external 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 in the first step to obtain a drawing of G.

Step (i) can be performed by means of the algorithm described in [6], where it is shown that an outerplanar graph can be augmented to maximal by inserting dummy edges that do not asymptotically alter the degree of the graph.

Now 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 an 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 in subpaths, and each subpath "cuts" H, that is, 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 (and of any subtree recursively drawn) 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 $u_1, u_2, \ldots, u_{4d+1}$ (resp. $l_1, l_2, \ldots, l_{4d+1}$) from the lowest to the highest (resp. from the highest to the lowest). All nodes of $L(T)$ and of $R(T)$ lie on l_{4d+1}.

Assume T is rooted at any node r of degree at most 2. Select a path $S = (v_0, v_1, \ldots, v_m)$ in T, called *spine*, such that (a) $v_0 = r$, (b) for $1 \leq i \leq k$, v_i is root of the subtree with the greatest number of nodes among the subtrees rooted at children of v_{i-1}, and (c) v_m is a leaf. The nodes of T belonging (not belonging) to S are *spine nodes* (*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 (right) child of v_{i-1} in S.

Path S is subdivided into subpaths, based on the alternance between left and right edges that compose S. Iteratively subdivide S in paths $S_0, S_1, \ldots, S_q$, so that, for $0 \leq j \leq q$, $S_j = (v^j_k, v^j_{k+1}, \ldots, v^j_l, v^j_{l+1}, \ldots, v^j_{f-1}, v^j_f)$ is defined as follows: $v^0_k = v_0$, and, for $1 \leq j \leq q$, v^j_k is the node after v^{j-1}_f in S. Suppose that (v^j_k, v^j_{k+1}) is a right edge. Let v^j_l be the first node after v^j_k in S such that (v^j_l, v^j_{l+1}) is a left edge. If (v^j_{l+1}, v^j_{l+2}) is a left edge (right edge) then let v^j_f be the first node after v^j_l in S such that (v^j_f, v^j_{f+1}) is a right (resp. left) edge. Now suppose that (v^j_k, v^j_{k+1}) is a left edge. Let v^j_l be the first node after v^j_k in S such that (v^j_l, v^j_{l+1}) is a right edge. If (v^j_{l+1}, v^j_{l+2}) is a right edge (left edge) then let v^j_f be the first node after v^j_l in S such that (v^j_f, v^j_{f+1}) is a left (resp. right) edge. Notice that S^q_j could have no vertex v^q_l or v^q_f if the spine ends before a left edge or a right edge is encountered. T is also subdivided in subtrees: For $0 \leq j \leq q$, 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 $0 \leq j \leq q$. We distinguish eight cases, based on whether (i) j is even (Cases 1-2-3-4) or odd (Cases 5-6-7-8); (ii) (v^j_k, v^j_{k+1}) is a right edge (Cases 1-2-5-6) or a left edge (Cases 3-4-7-8); and (iii) (v^j_{l+1}, v^j_{l+2}) 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, draw $L(T_j)$ and $R(T_j)$ on l_{2d+1} so that the each node of $L(T_j)$ (of $R(T_j)$) is one unit to the right (to the left) of its left (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 $L(T_j)$ (of $R(T_j)$) is one unit to the left (to the right) of its left (right) child. Denote by h_l the vertical grid line passing through v_l, 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

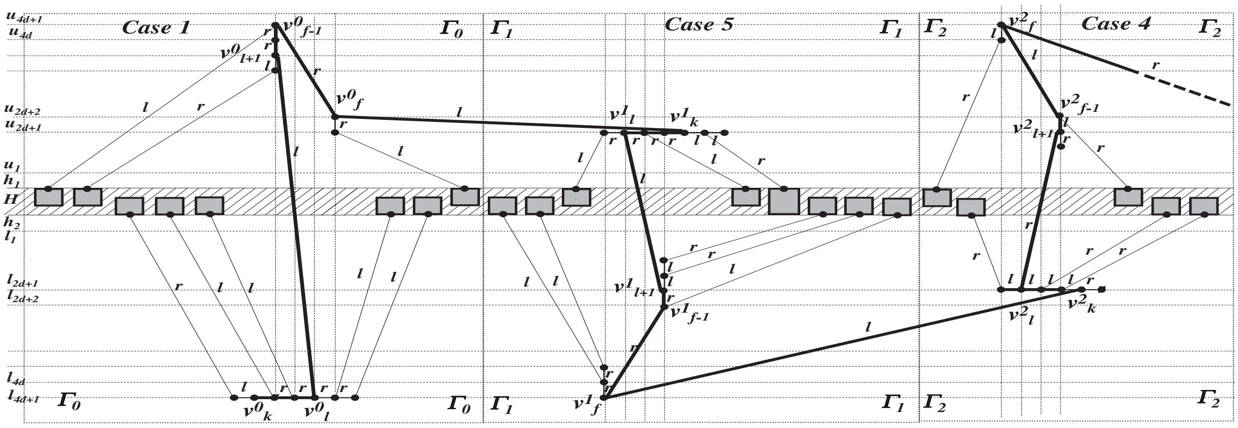

Figure 1: Thick segments represent the edges of S. Left (right) edges are labeled by l (resp. by r).

the left, and two units to the left of h_l, respectively.

Case 1. (see Fig. 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 $R(T(v_{l+1}^j))$ till 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 $L(T(v_{l+1}^j))$ till 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 $R(T(v_{l+1}^j))$ till 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. (see Fig. 1) 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 $L(T(v_{l+1}^j))$ till 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. (see Fig. 1) 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 $R(T(v_{l+1}^j))$ 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.

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

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

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

In Γ_0 shift the nodes of $L(T_0)$ and $R(T_0)$ vertically, so that they keep the same x-coordinates and lie on line l_{4d+1}.

For each T_j, with $0 \le j \le q$, 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 between 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 $0 \le j \le q$ and j even, 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 (to the left) of v_l^j. Place such drawings in the left-right order induced by the order of their parents on l_{4d+1} or on l_{2d+1}, at one unit of horizontal distance between them, with the leftmost (resp. rightmost) vertical line intersecting the leftmost (resp. rightmost) drawing one unit to the right (resp. to the left) of the last node of $R(T_j)$ (resp. of $L(T_j)$), and with their leftmost and rightmost paths on h_1. Then, consider the drawings of

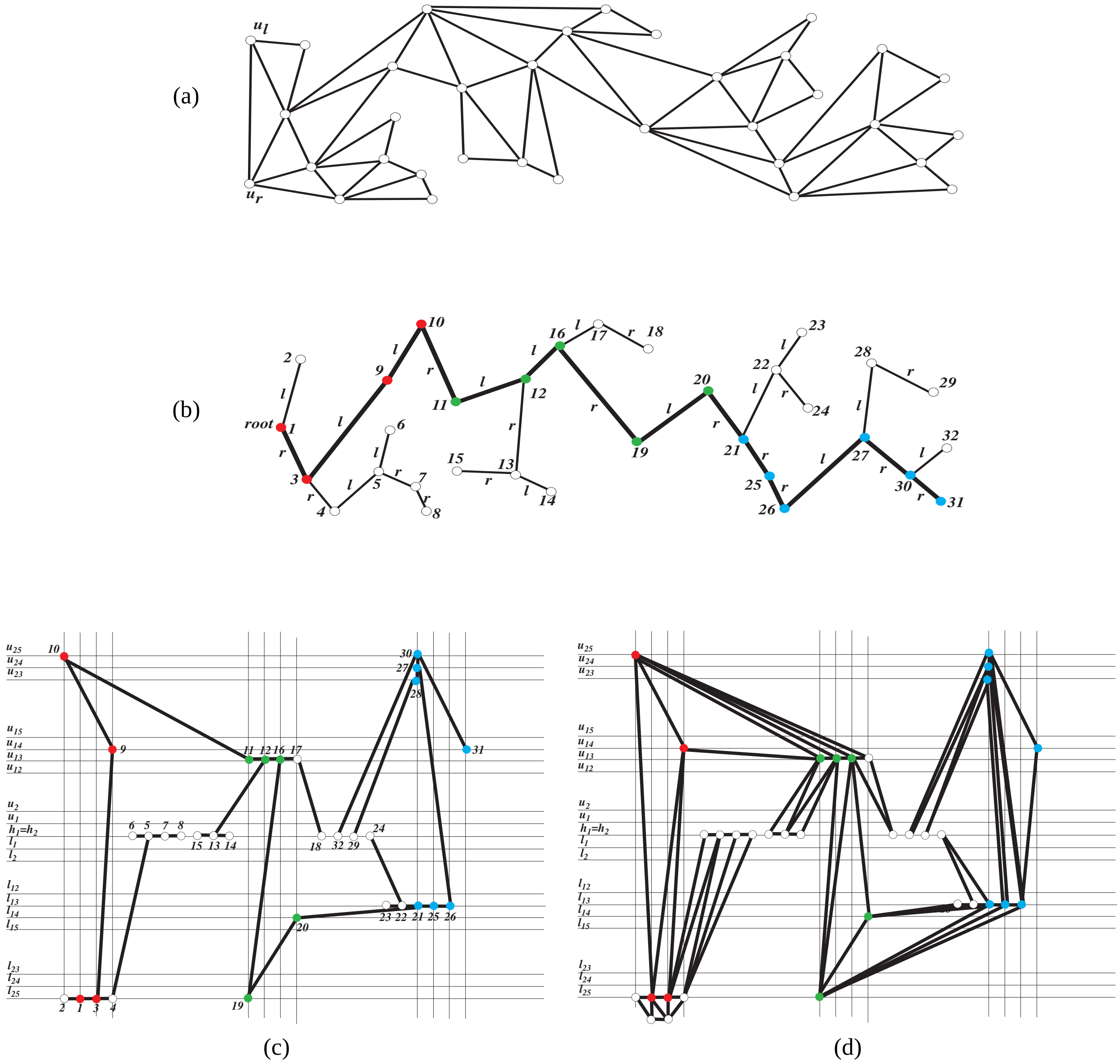

Figure 2: (a) A maximal outerplanar graph G with $n = 34$ vertices and degree $d = 6$. The poles of G are labeled by u_l and u_r. (b) The dual binary tree T of G. Edges labeled l (r) are between a node and its left (right) child. Thick edges show the spine S selected by the algorithm described in *Section 3*. 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.

the subtrees rooted at non-already drawn nodes of T_j that are children of nodes already drawn. Rotate such drawings of π radiants and place them so that $L(T_j)$ and $R(T_j)$ 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 $R(T(v^j_{l+1}))$. 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 $j = 1, \dots, m$, so that the leftmost vertical line intersecting Γ_j is one unit to the right of the rightmost vertical line intersecting Γ_{j-1}.

The constructed drawing Γ of T is star-shaped (see the appendix). Let's analyze the area requirement of Γ. The width of Γ is trivially $O(n)$. 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-nodes tree T constructed by the algorithm, we get: $H(n) = (4d+1) + (4d+1) + H(n/2) = O(d) + H(n/2) = O(d \log n)$.

Place the poles of G' both on the horizontal line one unit below v_0, so that they are at a distance of one horizontal unit, 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 the first step can be removed obtaining a drawing of G. Figure 2 shows an example of application of the algorithm described in this section.

Theorem 2 *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 3 *Any n-vertex outerplanar graph with constant degree has a straight-line outerplanar drawing in $O(n \log n)$ area.*

4 Conclusions and Conjectures

In this paper we have shown that $O(dn \log n)$ area is always achievable for straight-line drawings of outerplanar graphs.

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. 3.a, for which she claimed:

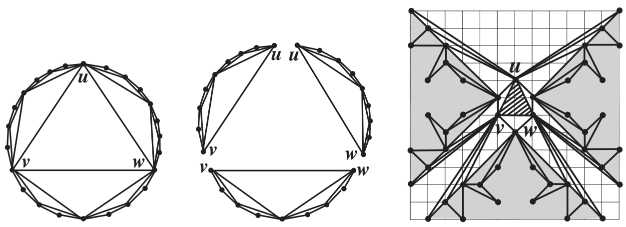

Figure 3: (a) A snowflake graph. (b) A snowflake graph subdivided in three complete outerplanar graphs. (c) Drawing snowflake graphs in linear area. The shaded regions correspond to the three copies of the drawing of the internal subgraph of a complete outerplanar graph constructed by the algorithm in [3].

Conjecture A [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 is composed of three identical $O(n)$-vertex complete outerplanar graphs (see Fig. 3.b), defined in [3] as the outerplanar graphs whose dual graphs are complete binary trees. For complete outerplanar graphs an $O(\sqrt{n}) \times O(\sqrt{n})$ area straight-line drawing algorithm has been presented in [3]. Such an algorithm constructs drawings in which the vertices of the internal subgraph that are connected to the poles lie on two half-lines l_1 and l_2 with a common endpoint and with slopes $-\pi/4$ and $\pi/4$. The entire drawing of the internal subgraph lies in the wedge with angle $\pi/2$ delimited by l_1 and l_2. Hence, considering three copies of the drawing of the internal subgraph of a complete outerplanar graph constructed by the algorithm in [3], rotating them of 0, π, and $3\pi/2$, respectively, placing the drawings together, and inserting the three vertices that are poles for the three complete outerplanar graphs, an $O(\sqrt{n}) \times O(\sqrt{n})$ area straight-line drawing of a snowflake graph can be obtained (see Fig. 3.c).

We would like to point up that all 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 one dimension is $\Omega(n)$) should be pursued.

Conjecture 1 *There exist n-vertex outerplanar graphs that in any straight-line drawing in which one dimension is $O(n)$ require $\omega(\log n)$ in the other dimension.*

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 binary tree as a subgraph).

Acknowledgements

Thanks to Giuseppe Di Battista for useful comments and suggestions.

References

[1] T. C. Biedl. Drawing outer-planar graphs in O(n log n) area. In *Graph Drawing*, pages 54–65, 2002.

[2] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, 1999.

[3] G. Di Battista and F. Frati. Small area drawings of outerplanar graphs. In *Graph Drawing*, pages 89–100, 2005.

[4] D. Dolev and H. W. Trickey. On linear area embedding of planar graphs. Technical report, Stanford, USA, 1981.

[5] A. Garg and A. Rusu. Area-efficient drawings of outerplanar graphs. In *Graph Drawing*, pages 129–134, 2003.

[6] G. Kant and H. L. Bodlaender. Triangulating planar graphs while minimizing the maximum degree. *Inf. Comput.*, 135(1):1–14, 1997.

[7] C. E. Leiserson. Area-efficient graph layouts (for VLSI). In *FOCS*, pages 270–281, 1980.

Appendix: Proof of Star-Shaped Visibility

We show that, given a rooted binary tree T, dual of a maximal outerplanar graph G, the algorithm described in Section 3 constructs a star-shaped drawing Γ of T.

First, observe that the claim that all 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.

Further, all 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 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 (see also [3])

Γ satisfies property (i) of a star-shaped drawing by construction. Concerning property (iv), $L(T)$ 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 and u_r to the nodes of $L(T)$ and $R(T)$ without creating crossings with Γ. Now we analyze properties (ii) and (iii) of a star-shaped drawing. For every node v lying in the upper or in the lower part of Γ, all nodes of $P_l(v)$ and of $P_r(v)$ are placed either on h_1, or on h_2, or in the upper, or in the lower part of Γ. This allows to verify that properties (ii) and (iii) of a star-shaped drawing are satisfied by Γ separately for each recursive step of the algorithm. Consider a subtree T_j of T, as defined in Section 3. 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, but for v (see Fig. 4.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. 4.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} (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, but for v^j_{l-1} (resp. but for v^j_{l-1} and its child that lies on the same horizontal line of v^j_{l-1}) 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 polygonal line (see Figs. 4.c and 4.d). 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 $\overline{v}$, 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 $P_l(\overline{v})$ and $P_r(\overline{v})$ has nodes all on h_1 or all on h_2, but for $\overline{v}$, while the other one has nodes all on u_{2d+1} or on l_{2d+1}. To prove that $\overline{v}$ is visible from the nodes of $P_l(\overline{v})$ and $P_r(\overline{v})$, observe that the slope of the edge connecting $\overline{v}$ to v^{j+1}_k is less than the one of the edge connecting $\overline{v}$ to $\hat{v}$ (see Fig. 4.e). Namely, the horizontal (vertical) distance between $\overline{v}$ and v^{j+1}_k is at least 4 (is exactly $2d$), while the horizontal (vertical) distance between $\overline{v}$ and $\hat{v}$ is exactly 3 (is exactly $2d-1$), and so the slope of $(\overline{v}, v^{j+1}_k)$ is less or equal than $\frac{2d}{4}$, while the one of $(\overline{v}, \hat{v})$ 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 more than 3 vertices. Hence, straight-lines can be drawn from $\overline{v}$ to the nodes of $P_l(\overline{v})$ and of $P_r(\overline{v})$ 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, but 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}, but for $\hat{v}$ and eventually for $\overline{v}$, 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. 4.f).

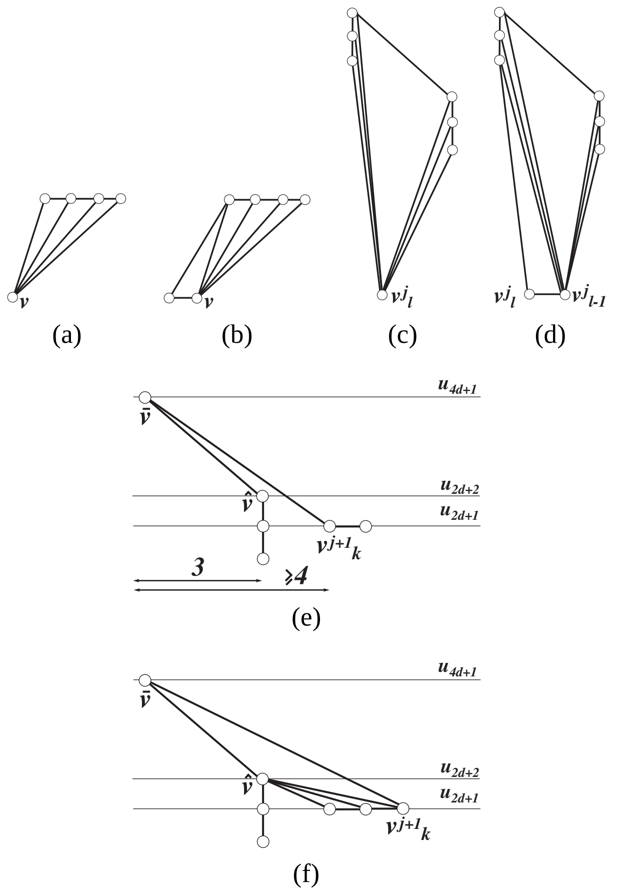

Figure 4: Illustrations for the proof of star-shaped visibility of the drawings constructed by the algorithm in *Section 3*.