Area Requirements for Drawing Hierarchically Planar Graphs

Xuemin Lin[1] and Peter Eades[2]

[1] Department of Computer Science, The University of Western Australia
Nedlands, WA 6907, Australia. e-mail: lxue@cs.uwa.oz.au.
[2] Department of Computer Science, The University of Newcastle
Callaghan, NSW 2308, Australia. e-mail: eades@cs.newcastle.edu.au.

(Extended Abstract)

Abstract. In this paper, we investigate area requirements for drawing s-t hierarchically planar graphs by straight-lines. Two drawing standards will be discussed: 1) each vertex is represented by a point and 2) grid visibility representation (that is, a line segment is allowed to represent a vertex). For the first drawing standard, we show an exponential area lower bound needed for drawing hierarchically planar graphs. The lower bound holds even for hierarchical graphs without transitive arcs, in contrast to the results for upward planar drawing. Applications of some existing algorithms from upward drawing can guarantee the quadratic drawing area for grid visibility representation but do not necessarily guarantee the minimum drawing area. Motivated by this, we will present another grid visibility drawing algorithm which is efficient and guarantees the minimum drawing area.

Keywords: Graph Drawing, Hierarchically Planar Graph, Straight Line Drawing, Visibility Representation, Drawing Area.

1 Introduction

Automatic graph drawing plays an important role in many computer-based applications such as CASE tools, software and information visualization, and VLSI design. The *upward* drawing convention for drawing acyclic directed graphs has received a great deal of attention since last decade. Many results [2, 4, 6, 9, 11] for drawing upward planar graphs have been published. However, acyclic directed graphs are not powerful enough to model all applications. Hierarchical graphs are then introduced, where layering information is also specified in an acyclic directed graph. Naturally, the "hierarchical" drawing convention (to be defined in Section 2) is proposed to display the specified layering information.

Due to the additional layering constraint, hierarchical drawing is different to upward drawing. Results in upward drawing and hierarchical drawing are not always the same. Issues, such as planar, straight-line, convex, and symmetric representations, have been revisited [7, 8, 10, 11] for drawing "hierarchically planar graphs" (to be defined in Section 2). In this paper, we investigate the

problem of area requirements for drawing "s-t hierarchically planar graphs" (to be defined in Section 2) with respect to a given *resolution requirement*.

In [6], it has shown that exponential areas are generally necessary for drawing upward planar graphs by the drawing standard of using points only to represent vertices and straight-lines to represent arcs. However, only quadratic drawing areas are required when upward planar graphs are *reduced*, meaning that no "transitive" arcs exist.

In this paper, we show that the results in [6] do not hold for hierarchically planar graphs. Specifically, we show that by the same drawing standard, exponential drawing areas are necessary even for hierarchically planar graphs without transitive arcs. This is the first contribution of the paper.

Secondly, we study the drawing area problem by allowing line segments to represent vertices. Particularly, we study the drawing area problem for "grid visibility representations" (to be defined in Section 2). An application of the algorithm VISIBILITY_DRAW in [4] gives the quadratic area for the grid visibility representation of hierarchically planar graphs. However, this algorithm does not necessary guarantee the minimal drawing area - an example will be given in Section 4. Motivated by this, we present an efficient algorithm for grid visibility representations of s-t hierachically planar graphs which guarantees the minimum drawing area.

The rest of the paper is organized as follows. Section 2 gives the basic terminology and background knowledge. Section 3 shows an exponential lower bound of drawing area. In Section 4, we present a drawing algorithm for producing a grid visibility representation which minimizes the drawing area. This is followed by the conclusions and remarks.

2 Preliminaries

The basic graph theoretic definitions can be found in [1].

A *hierarchical graph* $H = (V, A, \lambda, k)$ consists of a simple and acyclic directed graph (V, A), a positive integer k, and for each vertex u, an integer $\lambda(u) \in \{1, 2, ..., k\}$ with the property that if $u \rightarrow v \in A$, then $\lambda(u) > \lambda(v)$. For $1 \leq i \leq k$ the set $\{u : \lambda(u) = i\}$ of vertices is the *ith layer* of H and is denoted by L_i. An arc $u \rightarrow v$ in $H = (V, A, \lambda, k)$ is a *transitive* arc if there exists another directed path from u to v with length at least 2. An arc $u \rightarrow v$ is *long* if it spans more than two layers, that is, $\lambda(u) - \lambda(v) \geq 2$.

A *sink* of a hierarchical graph H is a vertex which does not have outgoing arcs, and a *source* of H is a vertex which does not have incoming edges. H is *s-t* if it has only one sink and one source.

A hierarchical graph is *proper* if it has no long arcs. Clearly, adding $\lambda(u) - \lambda(v) - 1$ dummy vertices to each long arc $u \rightarrow v$ in an unproper hierarchical graph H results in a proper hierarchical graph, denoted by H_p. H_p is called the *proper image* of H. Note that $H_p = H$ if H is proper.

To display the specified hierarchical information in a hierarchical graph, the *hierarchical drawing convention* is proposed, where a vertex in each layer L_i is

separately allocated on the horizontal line $y = i$ and arcs are represented as curves monotonic in y direction; see Figures 1 (a)-(c). A hierarchical drawing is *planar* if no pair of noincident arcs intersect. A hierarchical graph is *hierarchically planar* if it has a planar drawing admitting the hierarchical drawing convention. In this paper, we will discuss only hierarchical drawing convention.

An *embedding* of a proper hierarchical graph H consists of an ordering of the vertices in each layer, and is denoted by E_H. An *embedding* of an unproper hierarchical graph H means an embedding of the proper image H_p of H, and is also denoted by E_H.

A hierarchical drawing α of H *respects* E_H if α gives the same vertex ordering in each layer in H_p as E_H does. An embedding E_H is *planar* if any straight-line drawing of H_p respecting E_H is planar.

Various drawing standards exist for drawing hierarchically planar graphs by retaining planarity. In a *straight* line drawing α, each vertex v is represented as a point $\alpha(u)$ and each arc $u \rightarrow v$ is represented as a line segment connecting $\alpha(u)$ and $\alpha(v)$; see Figure 1 (a). In a *polyline* drawing, each long arc is allowed to be represented as a polygonal chain with bends allocated on some of the k horizontal lines $y = i$ for $1 \leq i \leq k$; see Figure 1 (c). In a *visibility representation* β, each vertex u is represented as a horizontal line segment $\beta(u)$ on $y = \lambda(u)$ and each arc $u \rightarrow v$ as a vertical line segment connecting $\beta(u)$ and $\beta(v)$, such that:

- $\beta(u)$ and $\beta(v)$ are disjoint if $u \neq v$, and
- a vertical line segment and a horizontal line segment do not intersect if the correponding arc and vertex are not incident.

See Figure 1(b), for example. Note that in a visibility representation, a line segment used to represent a vertex may be degenerated into a point.

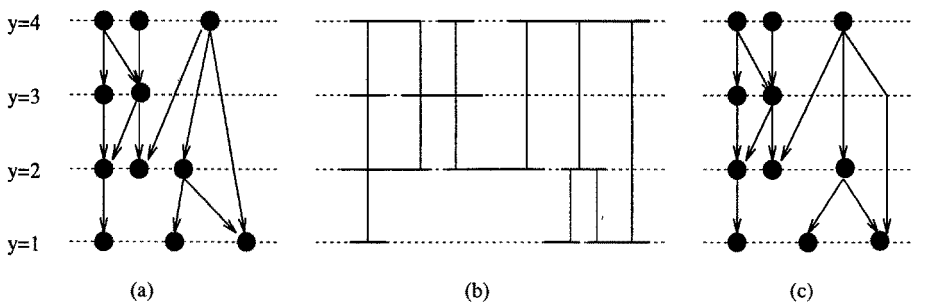

Fig. 1. Various Representations

A straight line drawing is a *grid* drawing if each vertex is at a grid position; and a polyline drawing is a *grid* drawing if vertices and bends are at grid positions. Similarly, in a *grid* visibility representation each horizontal line segment and vertical line segments must use grid points as their ends.

Drawing a hierarchically planar graph H consists of two phases: 1) finding a planar embedding E_H, and 2) finding a hierarchical drawing of H respecting E_H.

A linear time algorithm [10] was proposed for phase 1. In this paper, we concentrated on phase 2 and assume that a planar embedding is given. More restrictly, we study how to draw a planar embedding of an s-t hierarchically planar graph. This does not destroy the generality of the investigation of phase 2, since each planar embedding can be easily extended to a planar embedding of an s-t hierarchically planar graph [7, 8, 11].

3 An Exponential Area Lower Bound

The drawing *area* of a hierarchical drawing α is the minimal rectangle R which contains α and is composed of horizontal and vertical lines.

The *width* of a hierarchical drawing is the horizontal distance between the leftmost vertex and the rightmost vertex, while the *height* is the vertical distance between the top layer and the bottom layer. For a given hierarchical graph H, any hierarchical drawing of H has the fixed height. Consequently, the investigation of drawing area problem is reduced to that of drawing width problem.

In this section, we define a hierarchically planar graph $H_n = (V_n, A_n, \lambda_n, 4n-1)$ with $|V_n| = 10n-6$ for $n \geq 1$, such that 1) H_n has no transitive arcs, and 2) any planar straight-line drawing of H_n requires exponential drawing area with respect to a given vertex resolution requirement. More specifically, we define H_n by extending H_{n-1} for $n \geq 2$. The extension follows the same topology.

The graph H_1 consists of 4 vertices $\{t_1, c_{1,1}, c_{2,1}, s_1\}$ and three layers $L_3 = \{s_1\}$, $L_2 = \{c_{1,1}, c_{2,1}\}$, and $L_1 = \{t_1\}$. Four arcs connect H_1 in a diamond shape (see Figure 2(a)). To extend H_1 to H_2, ten vertices are added as depicted in Figure 2(b).

Generally, we construct H_{i+1} from H_i by adding the following ten vertices in a way depicted in Figure 2(c):

$$V_{i+1} = V_i \cup \{s_{i+1}, t_{i+1}, a_{1,i+1}, a_{2,i+1}, b_{1,i+1}, b_{2,i+1}, c_{1,i+1}, c_{2,i+1}, d_{1,i+1}, d_{2,i+1}\}.$$

The layering of H_n is described below:

$L_1 = \{t_n\}$, $L_{4n-1} = \{s_n\}$, $L_{2n} = \{c_{1,1}, c_{2,1}\}$; and for $1 \leq i \leq n-1$, $L_{2n+2i} = \{b_{j,i+1} : 1 \leq j \leq 2\}$, $L_{2n+2i-1} = \{s_i\} \cup \{a_{j,i+1} : 1 \leq j \leq 2\}$, $L_{2i+1} = \{t_{n-i}\} \cup \{c_{j,n-i+1} : 1 \leq j \leq 2\}$, and $L_{2i} = \{d_{j,n-i+1} : 1 \leq j \leq 2\}$.

The arc set A_n of H_n consists of:

$\{s_1 \rightarrow c_{j,1},\ c_{j,1} \rightarrow t_1 :\ 1 \leq j \leq 2\}$, $\{s_i \rightarrow s_{i-1},\ s_i \rightarrow b_{j,i},\ s_i \rightarrow c_{j,i} :\ 1 \leq j \leq 2,\ 2 \leq i \leq n\}$, $\{b_{j,i} \rightarrow a_{j,i},\ b_{j,i} \rightarrow c_{j,i-1},\ a_{j,i} \rightarrow t_{i-1},\ a_{j,i} \rightarrow d_{j,i} :\ 1 \leq j \leq 2,\ 2 \leq i \leq n\}$, and $\{c_{j,i} \rightarrow d_{j,i},\ d_{j,i} \rightarrow t_i,\ t_{i-1} \rightarrow t_i :\ 1 \leq j \leq 2,\ 2 \leq i \leq n\}$.

The following two lemmas can be immediately verified [12] based on the structure of H_n.

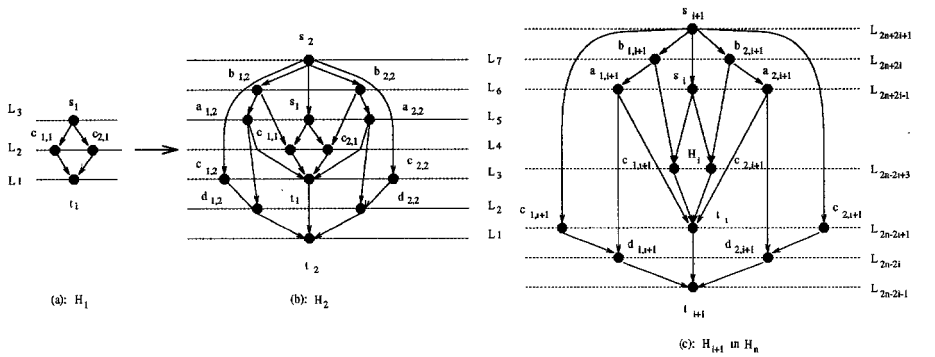

Fig. 2. Construct H_n

Lemma 1. *For $n \geq 1$, H_n is a hierarchically planar graph with no transitive arcs.*

Lemma 2. *For $n \geq 1$, the planar embedding E_{H_n} of H_n is unique up to a complete reversal.*

Here is the main result in this section.

Theorem 3. *For each H_n, suppose that α is a hierarchically planar straight line drawing of H_n, where each pair of vertices in the same layer are at least distance 1 apart. Then α has width at least $\Omega((2n-1)!)$.*

Proof: With respect to α, suppose that for $2 \leq i \leq n$, the distance between $\alpha(c_{1,i})$ and $\alpha(c_{2,i})$ is l_i.

Lemma 2 tells us that the planar embedding given by Figure 2(c) is unique to any hierarchically planar drawing of H_n up to a complete reversal.

Without loss of generality, we may assume that α gives the planar embedding as depicted in Figure 2(c).

Thus, in α, the relationship of the vertices orderings between α restricted to H_{i+1} and α restricted to H_i must be as the one illustrated in Figure 2(c). Consider the two triangles in Figure 3 with respect to α. Since the drawing α is a planar straight line drawing, elementary geometry implies $\frac{l_{i+1}}{l_i} \geq 2i(2i-1)$. Hence, $l_n \geq (2(n-1))!$. Therefore the Theorem holds.

□

Note that any hierarchical drawing of H_n has height $4n-2$. Thus each hierarchically planar straight-line drawing of H_n, in which each pair of vertices in the same layer are at least distance 1 apart, has area at least $\Omega(n(2n-2)!)$.

Note that H_n can be drawn upward planar in quadratic area (with respect to the number of vertices in H_n) by the algorithm in [6], but the layering of H_n is not preserved.

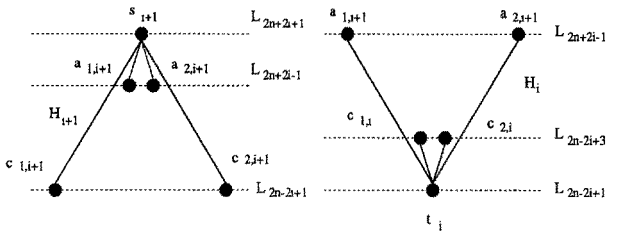

Fig. 3. Relationship among widths

4 Visibility Representation

The algorithm VISIBILITY_DRAW in [4] was developed for grid visibility representations of *s-t* upward planar graphs by using the *dual graph* technique; and it can be immediately applied to s-t hierarchically planar graphs for grid visibility representations. By applying the algorithm VISIBILITY_DRAW to an s-t hierarchically planar graph H, the output visibility representation of H has the width $w-1$ where w is the length of the longest path from the source to the sink of the dual graph of H. However, the minimum width of a visibility representation of H may be much smaller than $w-1$; and this is shown by the following examples.

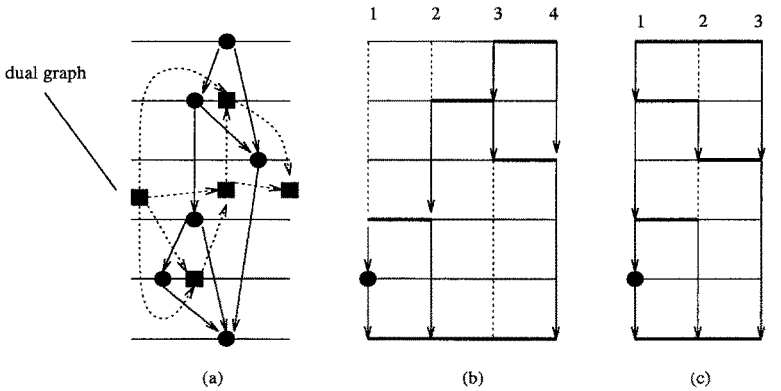

Fig. 4. Example 1

A hierarchical graph H_1 and its dual graph are illustrated in Figure 4(b), where the dual graph is depicted by rectangles and dotted arcs. An application of the algorithm VISIBILITY_DRAW produces the grid visibility representation of H_1 with width 3 as shown in Figure 4(b). However, the minimum width of a grid visibility representation of H is 2 as shown in Figure 4(c). Actually, the drawing in 4(c) is output by our algorithm.

We can generalize the example in Figure 4 to the graph H_2 as shown in Figure 5, where H_1 in Figure 4 is duplicated n times in H_2. It can be immediately verified that the length of the longest path from the source to the sink of the dual graph of H_2 is $4n$. Consequently, the width of the grid visibility representation of H_2 produced by the algorithm VISIBILITY_DRAW is $4n - 1$. However, it easy to show that the minimum width of a grid visibility representation of H_2 is $3n - 1$.

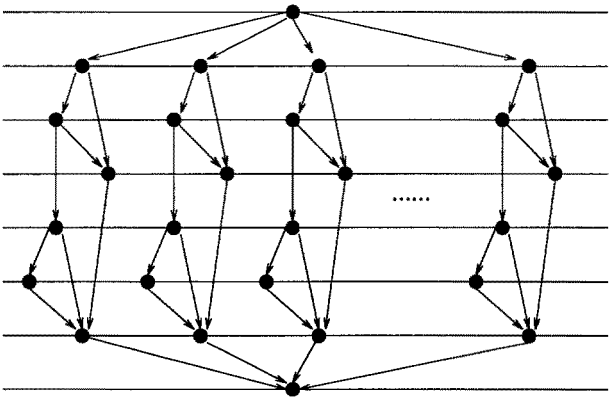

Fig. 5. Example 2

Inspired by the work in [4], in this section we present a new drawing algorithm GVP for the grid visibility representation of an s-t hierarchically planar graph with respect to a given planar embedding. The algorithm guarantees the minimal drawing area; that is, the width is minimized.

Like the algorithm VISIBILITY_DRAW, the algorithm GVP consists of two steps: 1) label each arc a by an integer $l(a)$ and 2) allocate an arc a on the vertical line $x = i$ if $l(a) = i$. However, the labeling technique in our algorithm GVP is different than that in the algorithm VISIBILITY_DRAW and therefore can guarantee the minimum drawing area.

The basic idea of our labeling technique is to push each individual vertical line segment as left as possible. This can be done by the following procedure ARC-LABELING which labels a given arc according to the previous labeling information.

Suppose that an s-t hierarchically planar graph H and its proper image H_p are given. Recall that H_p is a proper hierarchical graph. The label of a long arc $u \rightarrow v$ in H is inherited by the short arcs in H_p of which $u \rightarrow v$ is composed. In our algorithm, the current labeling information of H is kept with respect to H_p to simplify descriptions. For each layer L_i in H_p,

- I_i denotes the maximal integer label used among labeled arcs incoming to L_i,
- O_i denotes the maximal label used among labeled arcs outgoing from L_i, and

- IO_i denotes the maximal label used among arcs incident to the vertices in L_i all of whose outgoing and incoming arcs are labeled.

We also need the following notion for the description of our algorithm. Let E_H be a given planar embedding of H. For two arcs $u \to v$ and $x \to y$ between L_{i+1} and L_i, $u \to v$ is at the *left* of $x \to y$ with respect to E_H if $u \leq x$ and $v \leq y$ according to E_H.

Procedure ARC-LABELING

INPUT: an arc $u \to v$ to be labeled, a given planar embedding E_H, and the current O_i, I_i and IO_i for each layer L_i in H.
OUTPUT: a label $l(u \to v)$ of $u \to v$ and the updated O_i, I_i and IO_i for each layer L_i.

Suppose that $u \in L_{j+m}$ and $v \in L_j$. Let $L = \max_{1 \leq \eta \leq m-1}\{IO_{j+\eta}\}$ if $m \geq 2$, and $L = 0$ if $m = 1$. The arc $a = u \to v$ is labeled according to the four different cases:

1. If a is the leftmost outgoing arc from u and the leftmost incoming arc to v, $l(a) = \max\{IO_{j+m}, IO_j, L\} + 1$.
2. If a is the leftmost outgoing arc from u but not the leftmost incoming arc to v, $l(a) = \max\{IO_{j+m}, I_j, L\} + 1$.
3. If a is the leftmost incoming arc to v but not the leftmost outgoing arc from u, $l(a) = \max\{O_{j+m}, IO_j, L\} + 1$.
4. If a is neither leftmost outgoing from u nor leftmost incoming to v, $l(a) = \max\{O_{j+m}, I_j, L\} + 1$.

After labeling a, update $IO_{j+\eta}$ to $l(a)$ for $1 \leq \eta \leq m-1$. Meanwhile, we modify O_{j+m}, and IO_{j+m}, I_j, and IO_j as follows. $O_{j+m} = l(a)$ and $I_j = l(a)$. If all arcs incident to u are labeled, then $IO_{j+m} = \max\{l(a), I_{j+m}\}$. Similarly, if all arcs incident to v are labeled, then $IO_j = \max\{l(a), O_j\}$. □

To preserve the given planar embedding E_H in the algorithm GVP and then to guarantee the planarity of the drawing, we successively label each arc in H according to the trajectory of a leftmost depth-first search (LDFS) on H_p with respect to E_H. LDFS is a variation of the depth-first search technique [1].

LDFS

Start from the source of H_p. While going down by the depth-first search from vertex u, always first visit the leftmost unvisited outgoing arc $u \rightarrow v$ from u. If all outgoing arcs from v are already visited or v is a sink, then the LDFS procedure continues as follows.

S1: Terminate the current LDFS path. Goto S2.
S2: Along the reverse direction of the current LDFS path, trace back (till reach the source) to the bottom-most vertex v_0 which has unvisited outgoing arcs. If no such v_0 exists then stop the LDFS procedure, otherwise goto S3.
S3 Start the next LDFS path from v_0. □

For example, by applying LDFS to H_2 in Figure 2(b) we successively visit the arcs: $s_2 \rightarrow c_{1,2}$, $c_{1,2} \rightarrow d_{1,2}$, $d_{1,2} \rightarrow t_2$, $s_2 \rightarrow b_{1,2}$, $b_{1,2} \rightarrow a_{1,2}$, $a_{1,2} \rightarrow d_{1,2}$, $a_{1,2} \rightarrow t_1$, $t_1 \rightarrow t_2$, ...

Algorithm GVP

INPUT: an s-t hierarchically planar graph H and its planar embedding E_H.
OUTPUT: a grid visibility representation of H respecting E_H.

Step 1: Labeling. Initially, I_i, O_i and IO_i are set to zero. Label arcs successively as follows according to the ordering given by LDFS till all arcs are labeled. Note that each arc in H_p is a short arc. While an arc $u \rightarrow v$ in H_p is visited in LDFS, there are two cases:

case1: $u \rightarrow v$ is an arc of H. Then call the procedure ARC-LABELING for labeling $u \rightarrow v$.
case2: At least one of u, v is a dummy vertex to H; that is, $u \rightarrow v$ is one part of a long arc $x \rightarrow y$ in H. In this case, we do not label $u \rightarrow v$ separately but give a label to the whole arc $x \rightarrow y$. Thus, call the procedure ARC-LABELING for labeling $x \rightarrow y$. (In LDFS, we should be able to notice this long arc b immediately after x, and be able to reach y by the LDFS path from x.)

Step 2: Drawing. This step follows immediately Step 1 and draws H based on the output of Step 1. It consists the following two phases: drawing vertices and drawing arcs of H.
Drawing vertices. For each vertex $u \in H$, let A_u represent the set of arcs in H which are incident to u. Assume $u \in L_i$. Represent u by the horizontal line segment from $(\min_{a \in A_u}\{l(a)\}, i)$ to $(\max_{a \in A_u}\{l(a)\}, i)$.
Drawing arcs. Represent an arc $a = u \rightarrow v$ with $u \in L_i$ and $v \in L_j$ by the vertical line segment from $(l(a), i)$ to $(l(a), j)$. □

For instance, Figure 6(a) shows the result after applying Step 1 to H_2 in Figure 2(b), and Figure 6(b) illustrates the result after applying Step 2 in the algorithm GVP to the output (Figure 6(a)) of Step 1.

It can be verified [12] that the drawing given by Step 2 respects the given planar embedding; and thus,

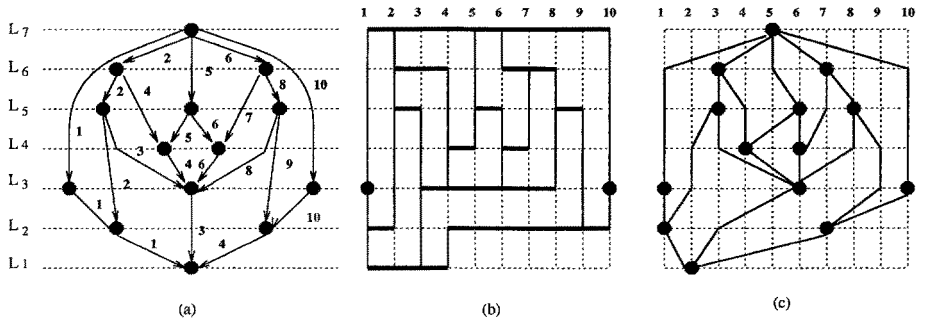

Fig. 6. Apply Algorithm GVP and Algorithm GRID_DRAW to H_2

Lemma 4. *The algorithm GVP gives a grid visibility representation of H respecting a given planar embedding E_H.*

Applying similar arguments as used in [4], we can immediately show that the grid visibility representation given by the algorithm GVP occupies drawing area $O(n^2)$. Furthermore, we can show:

Theorem 5. *Respecting a planar embedding E_H of a hierarchically planar graph H, the grid visibility representation of H produced by the algorithm GVP has the minimum drawing area.*

Sketch of the proof: It can be shown in [12], based on induction, that every arc has been allocated on the "most left-possible" vertical line. The theorem immediately follows. □

It is easy [12] to implement the algorithm GVP in time $O(n + L)$ if the ordering of outgoing arcs from each vertex is pre-specified in a given E_H, where L is the total lengths of long arcs in H. If such ordering has not been provided for each vertex, then the algorithm runs in time $O(n \log n + L)$.

5 Conclusions and Remarks

In this paper, we have shown an exponential area lower bound for planar straight-line drawings of hierarchically planar graphs without transitive arcs in contrast to the result [6] for upward planar drawing. An efficient algorithm has been presented for producing a grid visibility representation with the minimal drawing area.

Finally, we should note that if the algorithm GRID_DRAW is applied to the output of the algorithm GVP, then a grid polyline drawing is obtained, which guarantees the following properties:

- each long arc is represented by a poly line with at most two bends;
- the drawing area is $O(n^2)$.

Figure 4(c) shows the result after applying the algorithm GRID_DRAW to the drawing in Figure 4(b).

Note that our drawing algorithms do not necessarily produce a symmetric drawing when a graph is symmetric. This will be our future study.

References

1. J. A. Bondy and U. S. R. Murty, *Graph Theory with Application*, The Macmilan Press LTD, 1977.
2. G. Di Battista, P. D. Eades, R. Tamassia, and I. Tollis, Algorithms for Automatic Graph Drawing: An Annotated Bibliography, *Computational Geometry: Theory and Application*, 4, 235-282, 1994.
3. G. Di Battista and E. Nardelli, Hierarchies and Planarity Theory, *IEEE Tran. Sys. Man Cybern.* 18(6), 1035-1046, 1988.
4. G. Di Battista and R. Tamassia, Algorithms for Plane Representations of Acyclic Diagraphs, *Theoretical Computer Science*, 61, 175-198, 1988.
5. G. Di Battista and R. Tamassia, On-Line Planarity Testing, *SIAM Journal on Computing*, 25(5), 1996.
6. G. Di Battista, R. Tamassia, and I. G. Tollis, Area Requirement and Symmetry Display of Planar Upward Drawings, *Discrete & Computational Geometry*, 7(381-401), 1992.
7. P. Eades, Q. Feng and X. Lin, Straight-Line Drawing Algorithms for Hierarchical Graphs and Clustered Graphs, *Graph Drawing'96*, LNCS 1190, Springer-Verlag, 113-128, 1996.
8. P. Eades, X. Lin and R. Tamassia, An Algorithm for Drawing a Hierarchical Graph, *International Journal of Computational Geometry and Applications*, 6(2), 145-155, 1996.
9. A. Garg and R. Tamassia, On the Computational Complexity of Upward and Rectilinear Planarity Testing, *Graph Drawing'94*, LNCS, Springer-Verlag, 286-297, 1995.
10. L. S. Heath and S. V. Pemmaraju, Recognizing Leveled-Planar Dags in Linear Time, *Draph Drawing'95*, LNCS 1027, Springer-Verlag, 1995.
11. X. Lin, *Analysis of Algorithms for Drawing Graphs*, PHD thesis, University of Queensland, 1992.
12. X. Lin and P. Eades, Area Requirement for Drawing Hierarchically Planar Graphs, Tec. Rep., The University of Western Australia, 1997.