Area Minimization for Grid Visibility Representation of Hierarchically Planar Graphs

Xuemin Lin[1] and Peter Eades[2]

[1] School of Computer Science & Engineering
The University of New South Wales
Sydney, NSW 2052, Australia. lxue@cse.unsw.edu.au.
[2] Department of Computer Science & Software Engineering
The University of Newcastle
Callaghan, NSW 2308, Australia.

Abstract. Hierarchical graphs are an important class of graphs for modelling many real applications in software and information visualization. In this paper, we shall investigate the computational complexity of constructing minimum area grid visibility representations of hierarchically planar graphs. Firstly, we provide a quadratic algorithm that minimizes the drawing area with respect to a fixed planar embedding. This implies that the area minimization problem is polynomial time solvable restricted to the class of graphs whose planar embeddings are unique. Secondly, we show that the area minimization problem is generally NP-hard.

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

1 Introduction

Automatic graph drawing plays an important role in many modern computer-based applications, such as CASE tools, software and information visualization, VLSI design, visual data mining, and internet navigation. Directed acyclic graphs are an important class [2] of graphs to be investigated in this area. The *upward* drawing convention for drawing directed acyclic graphs has received a great deal of attention since last decade; and a number of results for drawing *upward planar graphs* have been published [2,4,6,10,15,16].

Consider [7,8] that directed acyclic graphs are not powerful enough to model every real-life application. "Hierarchical" graphs are then introduced, where layering information is added to a directed acyclic graph. Consequently, the "hierarchical" drawing convention is proposed to display the specified layering information.

Due to the additional layering constraint, most problems in hierarchical drawing are inherently different to those in upward drawing. For example, testing for "upward planarity" of directed acyclic graphs is NP-Complete [10], while it can be done in linear time [3,11] for "hierarchically planarity". Therefore, issues, such as "planar", "straight-line", "convex", and "symmetric" representations,

T. Asano et al. (Eds.): COCOON'99, LNCS 1627, pp. 92–102, 1999.

have been revisited [7,8,11,12,13] with respect to hierarchically planar graphs. In this paper, we shall investigate the problem of minimizing the drawing area for hierarchically planar graphs, where drawings are restricted to the 2-dimensional space.

Drawing a hierarchically planar graph involves two phases: 1) computing a "planar embedding", and 2) finding a good drawing "respecting" the embedding. A linear time algorithm [11] was proposed for phase 1. In [8], a simple and *force-directed* algorithm was developed that integrates the two phases and can deliver a convex and symmetric drawing. However, the results in [8] are applicable only to a special class of graphs - *well connected* graphs [8]. In [7], an efficient polynomial algorithm was provided for drawing hierarchically planar graph by the *straight-line* drawing standard (that is, using points to represent vertices and straight-line segments to represent arcs). Considering that these three algorithms [7,8,11] may produce drawings with exponential areas for a given *resolution requirement*, in our earlier work [13] we proved that exponential area is generally necessary for straight-line drawings. To resolve the exponential drawing area problem, a relaxation of the drawing standard such as allowing line segments to represent vertices was made, as with the upward planar graphs [2,6,16]. Particularly, in [4,13] it is shown that the drawing area can be always made within a quadratic area if the "grid visibility representation" is employed for hierarchically planar graphs. Moreover, an efficient grid visibility representation algorithm was presented that can achieve the minimum drawing area for a hierarchically planar graph with only one "source", only one "sink", and a fixed "planar embedding".

This paper presents a more general investigation than that in [13]. Firstly, we present an efficient (quadratic time) grid visibility representation algorithm that guarantees the minimum drawing area for a hierarchically planar graph with arbitrary number of sources and sinks, and a fixed planar embedding. This implies that for a hierarchical graph with the unique planar embedding, the area minimization problem for grid visibility representation is polynomial time solvable. This result is more general than that in [13]. The second contribution of the paper is to prove that the problem of area minimization is NP-hard for the grid visibility representation if a planar embedding is not fixed.

The rest of the paper is organized as follows. Section 2 gives the basic terminology and background, as well as the precise definition of our problem. Section 3 presents the first contribution and Section 4 presents the second contribution. 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 directed acyclic 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 $u \to v \in A$, then $\lambda(u) > \lambda(v)$. For $1 \le i \le k$

the set $\{u : \lambda(u) = i\}$ of vertices is the *ith layer* of H and is denoted by L_i. An arc $a = u \rightarrow v$ in H is *long* if it spans more than two layers, that is, $\lambda(u) - \lambda(v) \geq 2$.

For each vertex u in H, we use A_u to denote the set of arcs incident to u, A_u^+ to denote the set of arcs outgoing from u, and A_u^- to denote the set of arcs incoming to u. A *sink* u of a hierarchical graph H is a vertex that does not have outgoing arcs; that is, $A_u^+ = \emptyset$. A *source* of H is a vertex that does not have incoming arcs; that is, $A_u^- = \emptyset$.

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 improper 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). In this paper, we will discuss only hierarchical drawing convention.

A drawing is *planar* if no pair of no-incident arcs intersect. A hierarchical graph is *hierarchically planar* if it has a planar drawing admitting the hierarchical drawing convention.

An *embedding* E_H of a proper hierarchical graph H gives an ordered vertex set $\mathcal{L}_i$ for each layer L_i in H. For a pair of vertices $u, v \in \mathcal{L}_i$, u is on the *left side* of v if $u < v$. An *embedding* of an improper hierarchical graph H means an embedding of the proper image H_p of H, and is also denoted by E_H. Note that for an improper hierarchical graph H, $\mathcal{L}_i$ may contain more vertices than L_i due to additional dummy vertices.

A hierarchical drawing α of H *respects* E_H if for each pair of vertices u, v in a $\mathcal{L}_i$, the x-coordinate value $\alpha(u)$ is smaller than that of $\alpha(v)$ if and only if $u < v$. An embedding E_H is *planar* if a straight-line drawing of H_p respecting E_H is planar.

Besides straight line drawing, various drawing standards are available for drawing hierarchically planar graphs; see Figures 1(a) - 1(c) for example. In this paper we are focused only on a “visibility representation” of a hierarchically planar graph. In a *visibility representation* β, each vertex u is represented as a horizontal line segment $\beta(u)$ on $y = \lambda(u)$ and each arc $u \rightarrow v$ is represented 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 corresponding 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 degenerate to a point.

A visibility representation is a *grid drawing* if horizontal line segments and vertical line segments use only grid points as their ends. The drawing *area* of a grid visibility representation β is the area of the minimum isothetic rectangle that contains β. The *width* and the *height* of β are the width and height respectively

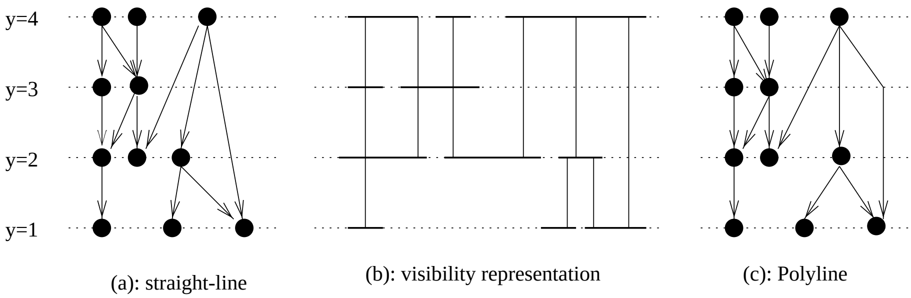

Fig. 1. Various Representations

of this rectangle. In [13], we showed that a hierarchical graph is hierarchically planar if and only if it admits a grid visibility representation. In this paper, we shall study the following optimization problem.

Minimum Area of Grid Visibility Drawing (MAGVD)

INSTANCE: A hierarchical planar graph H is given.

QUESTION: Find a grid visibility representation of H such that the drawing area is minimized.

Without loss of generality, we assume that in H, there is no *isolated* vertex - a vertex without any incident arcs. Note that all grid visibility representations of a given hierarchically planar graph have the fixed height according to the drawing convention. Consequently, MAGVD is reduced to the width drawing minimization problem. In section 4, we will prove that MAGVD is NP-hard. Firstly, however, we show that it is polynomially solvable if the planar embedding is given as part of the input.

3 Area Minimization for a Fixed Planar Embedding

Di Battista and Tamassia [4] proposed an efficient grid visibility representation algorithm, VISIBILITY_DRAW, for drawing upward planar graphs. In fact the algorithm can be immediately applied to hierarchically planar graphs [13] with a fixed planar embedding. Below is the version for hierarchical graphs.

Algorithm VISIBILITY_DRAW

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

Step 1: Labelling. Give each arc a an integer $l(a)$.

Step 2: Drawing. This step follows immediately Step 1 and draws H based on the output of Step 1. It consists of 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 $a = u \to v$ with $u \in L_i$ and $v \in L_j$ by the vertical line segment from $(l(a), i)$ to $(l(a), j)$. □

Suppose that the largest x-coordinate value assigned to a grid visibility representation β of H is N, and the smallest one is 1. Then the width of β is $N - 1$. Clearly, the key in applying the Algorithm VISIBILITY_DRAW to minimizing drawing width is to optimize Step 1. Note that neither the original labelling technique (*dual graph* technique) in [4] nor the labelling technique in [13] can guarantee the minimality of the drawing width for hierarchically planar graphs with arbitrary number of sources and sinks, and with a fixed planar embedding.

In this section, we provide a new algorithm OPTIMAL_LABELLING to Step 1, which guarantees the minimum drawing area for a hierarchically planar graph with a fixed planar embedding. The basic idea is simple - labelling each arc with the minimal possible integer.

To describe OPTIMAL_LABELING, the following notation is needed. For two different arcs $a_1 = u_1 \to v_1, a_2 = u_2 \to v_2 \in H$, a_1 is on the *left* side of a_2 with respect to E_H if and only if in E_H there are a vertex u on a_1 and a vertex v on a_2 such that u and v are in the same layer and u is on the left side of v. Note that E_H is a planar embedding of H_p, and u and v may be dummy vertices on the long arcs. The four possible cases are depicted in Figures 2 (a) - (d), where dotted lines indicate possible extensions to long arcs.

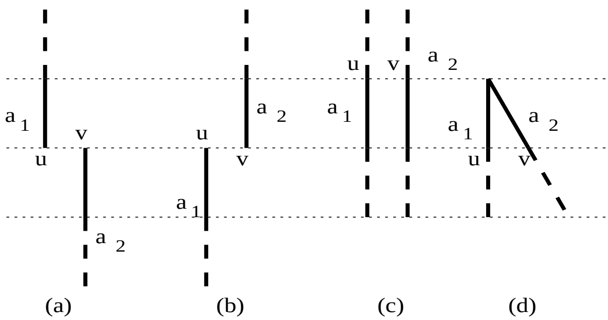

Fig. 2. 4 possible cases where a_1 is on the left side of a_2

An arc a in H is *left-most* with respect to E_H if there is no arc in H that is on the left side of a.

The algorithm OPTIMAL_LABELLING iteratively finds the left-most arcs (with respect to E_H) in H to label. In each iteration i:

S1: OPTIMAL_LABELLING scans the hierarchical graph H from the top layer to the bottom layer to label the left-most arcs in the current H with the integer i. Go to S2.

S2: OPTIMAL_LABELLING deletes all arcs labelled in this iteration; and deletes the isolated vertices resulted after arcs deletion in H. Go to $(i+1)$th iteration.

The algorithm terminates if all arcs in H are labelled.

For instance, Figure 3(b) shows the result after applying the algorithm OPTIMAL_LABELLING to the graph with respect to the planar embedding depicted in Figure 3(a). Meanwhile, Figure 3(c) illustrates the result after applying Step 2 in VISIBILITY_DRAW to the output (Figure 3(b)) of OPTIMAL_LABELLING.

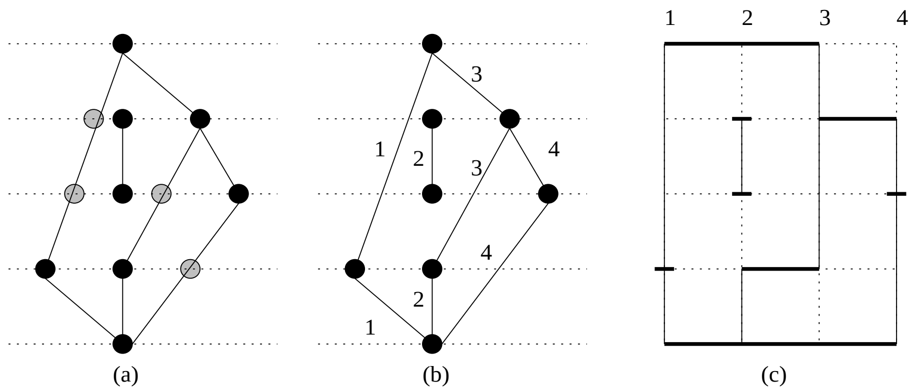

Fig. 3. OPTIMAL_LABELLING

It can be immediately verified that the drawing, given by a combination of OPTIMAL_LABELLING and Step 2 in VISIBILITY_DRAW, respects the given planar embedding E_H; that is,

Lemma 1. *The combination of OPTIMAL_LABELLING and Step 2 in VISIBILITY_DRAW gives a grid visibility representation of H respecting a given planar embedding E_H.*

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

Theorem 2. *Respecting a given planar embedding E_H of a hierarchically planar graph H, the grid visibility representation of H, produced by the combination of OPTIMAL_LABELLING and Step 2 in VISIBILITY_DRAW, has the minimum drawing width.*

Proof. **Sketch:** Basically, every arc has been allocated on the "left-most possible" vertical line by OPTIMAL_LABELLING. This can be immediately verified based on a mathematic induction, following the ordering of arc labelling. The full proof can be found in the full paper [14]. □

Suppose that vertices in each layer L_i in H are stored from left to right according to their ordering given by E_H, as well as the vertices in $\mathcal{L}_i$ do. Assume that for each vertex u, arcs in A_u^+ are also stored from left to right according their ordering. To execute OPTIMAL_LABELLING efficiently, S1 and S2 can be integrated together in each iteration. In each iteration i, start with the left most vertex u in the top layer of the remaining H, and search down along the leftmost arc $a = u \to v$ in A_u^+ to see if a is the leftmost arc in the current H:

case 1: If a is also the leftmost arc in the current H, then label a with i and delete a from H. Consequently, delete any resultant isolated vertices from H. Continue the iteration from the layer one level below the layer of v if A_v^+ is empty; otherwise continue the iteration from the layer of v.

case 2: If a is not the left most arc in the current H, in the remaining H there must be a vertex w such that the leftmost arc $b = u_1 \to v_1$ of A_w^+ is on the left side of a and u_1 is the leftmost vertex in the layer of u_1. Choose such a vertex u_1 that its layer number is maximized. Then continue the iteration i from the layer of u_1.

Clearly, the computation involved in the above two cases is proportional to a scan of the first vertices in the layers spanned by a. Consequently, for a $H = (V, A, \lambda, k)$ each iteration takes $O(k)$ time. Note that the number of iteration must be less than the number of arcs, because each iteration labels at least one arc. Therefore, the algorithm OPTIMAL_LABELLING runs in time $O(k|A|)$. As H is planar, $|A| = O(|V|)$; and thus the algorithm runs in $O(k|V|)$.

4 The Complexity of MAGVD

In this section, we prove the NP-hardness of MAGVD by showing the NP-completeness of the corresponding decision problem. As mentioned earlier, in MAGVD we need only to consider the drawing width minimization problem.

Decision Problem for MAGVD (DPMAGVD)

INSTANCE: A hierarchical planar graph H, and an integer K.

QUESTION: Find a grid visibility representation of H such that its width is not greater than K.

It is well known [9] that the 3-PARTITION problem is NP-complete. In our proof, we will transform 3-PARTITION to a special case of DPMAGVD.

3-PARTITION

INSTANCE: A finite set S of $3n$ elements, an integer B, and an integer weight $s(e)$ for each element $e \in S$ are given such that each $s(e)$ satisfies $\frac{B}{4} < s(e) < \frac{B}{2}$ and $\sum_{e \in S} s(a) = nB$

QUESTION: Can S be partitioned into n disjoint sets S_1, S_2, ... , S_n such that for $1 \leq i \leq n$, $\sum_{e \in S_i} s(e) = B$?

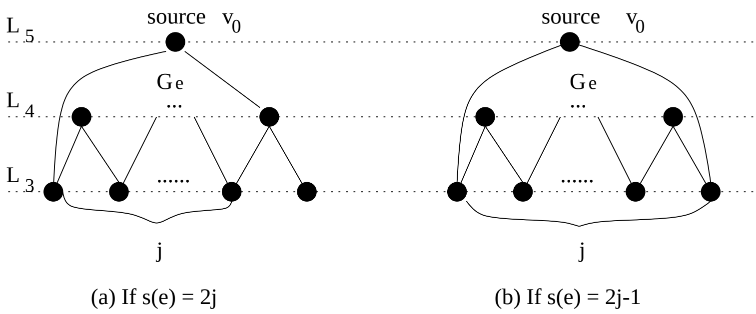

Fig. 4. Constructing G_e

Now, we transform an instance I_{3P} of 3-PARTITION to an instance $D_{I_{3P}} = (H_{I_{3P}}, K_{I_{3P}})$ of DPMAGVD. The hierarchically planar graph $H_{I_{3P}}$ has five layers and the top layer has only one source v_0. $H_{I_{3P}}$ is constructed as follows:

> Each element $e \in S$ corresponds to a three-layered graph G_e that hangs over the source v_0 of $H_{I_{3P}}$, such that G_e takes one of the two possible graphs as depicted in Figures 4(a) and 4(b) depending on the odevity of $s(e)$. Further, in $H_{I_{3P}}$ we also duplicate n times a graph G_B. Here, G_B also takes one of the two shapes, depicted in Figures 5(a) and 5(b), subject to the odevity of B.

We assign $K_{I_{3P}}$ as $n(B + 2)$.

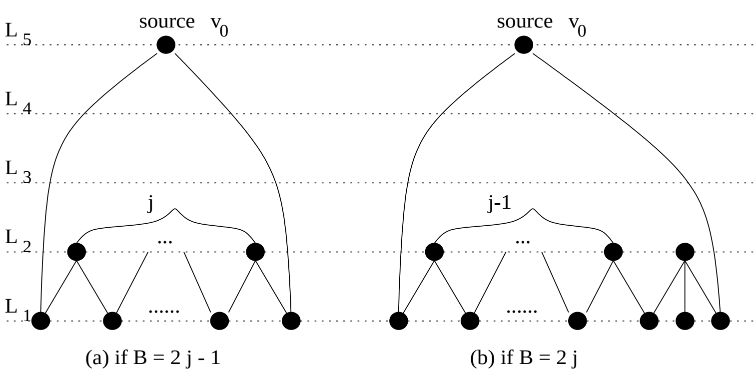

Fig. 5. Constructing G_B

Note that a G_e has a unique planar embedding up to a complete reversal, as well as G_B does. Consequently an application of OPTIMAL_LABELLING to a G_e (or G_B) can guarantee the minimality of the drawing width of G_e (or G_B). More specifically, the following Lemma can be immediately verified by the structures of G_e and G_B.

Lemma 3. *The minimum drawing width of a grid visibility representation of G_e is $s(e)$, and the minimum drawing width of G_B is $B+2$.*

Corollary 4. *Let wid_β denote the width of a grid visibility representation β of $H_{I_{3P}}$. Then $wid_\beta \geq n(B+2)$.*

Proof. The Corollary immediately follows Lemma 3 and the fact that $H_{I_{3P}}$ consists of n G_Bs. □

The following fact is the key to the proof of NP-Completeness of DPMAGVD.

Theorem 5. *3-PARTITION has a solution to I_{3P} if and only if DPMAGVD has a solution to $D_{I_{3P}} = (H_{I_{3P}}, K_{I_{3P}})$.*

Proof. Note that a pair of G_{e_1} and G_{e_2} cannot have any intervening apart from v_0 in a grid visibility representation α of $H_{I_{3P}}$. Further, in α a G_e must be either drawn inside a G_B or outside all G_Bs.

The above facts together with Lemma 3 and Corollary 4 imply that DPMAGVD has a solution if and only if the set Π of subgraphs G_e of $H_{I_{3P}}$ can be divided into n disjoint sets Π_1, Π_2, Π_3, ... , Π_n, and there exists a grid visibility representation β of $H_{I_{3P}}$, such that

- each Π_i consists of 3 different graphs $G_{e_{i_1}}$, $G_{e_{i_2}}$, and $G_{e_{i_3}}$;
- $s(e_{i_1}) + s(e_{i_2}) + s(e_{i_3}) = B$;
- in β, each Π_i is drawn inside the drawing of a G_B; and
- in β, the drawing of each G_B contains only one Π_i.

The theorem immediately follows. □

Furthermore, the following Lemma is immediate based on the construction of D_{I_3P} from I_{3P}.

Lemma 6. *The transformation between each I_{3P} to $D_{I_{3P}}$ takes polynomial time with respect to $n+B$.*

Note that Lemma 6 does not necessary imply that there is a polynomial time transformation, with respect to the input size n of 3-PARTITION, between I_{3P} and $D_{I_{3P}}$, because B may be arbitrarily larger. However, it has been shown [9] that 3-PARTITION is strongly NP-complete; that is, it is NP-complete even if B is bounded by a polynomial of n. This, together with Theorem 5 and Lemma 6, imply that DPMAGVD is NP-Complete. Consequently [9]:

Theorem 7. *MAGVD is NP-hard.*

5 Conclusions

In this paper, we studied the drawing area minimization problem for hierarchically planar graph, restricted to the grid visibility representation. An efficient algorithm has been presented for producing a grid visibility representation with the minimal drawing area if a planar embedding is given and fixed. This implies that for a class of hierarchically planar graphs whose planar embeddings are unique, such as the well connected graphs [8], MAGVD is polynomial time solvable. However, we showed that in general, MAGVD is NP-hard. Note that a slight modification of the proof of NP-hardness can lead to a stronger result: MAGVD is NP-hard even restricted to hierarchically planar graphs with only one source [14]. For a possible future study, we are interested in investigating:

– whether or not similar results exist for upward planar graphs;
– a good approximation algorithm for solving MAGVD; and
– symmetric drawing issues.

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 Digraphs, *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, X. Lin, and H. Nagamocha, Straight-Line Drawing Algorithms for Hierarchical Graphs and Clustered Graphs, to appear in *Algorithmic*.
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. M. R. Garey and D. S. Johnson, *Computers and Intractability - a guide to the theory of NP-Completeness*, Freeman, 1979.
10. A. Garg and R. Tamassia, On the Computational Complexity of Upward and Rectilinear Planarity Testing, *Graph Drawing'94*, LNCS, Springer-Verlag, 286-297, 1995.
11. L. S. Heath and S. V. Pemmaraju, Recognizing Leveled-Planar Dags in Linear Time, *Graph Drawing'95*, LNCS 1027, Springer-Verlag, 1995.
12. X. Lin, *Analysis of Algorithms for Drawing Graphs*, PHD thesis, University of Queensland, 1992.
13. X. Lin and P. Eades, Area Requirements for Drawing Hierarchically Planar Graphs, *Graph Drawing'97*, LNCS, Springer-Verlag, 1997.

14. X. Lin and P. Eades, Towards Area Requirements for Drawing Hierarchically Planar Graphs, *Manuscript*, UNSW, 1998.
15. R. Tamassia, On Embedding a Graph in the Grid with the Minimum Number of Bends, *SIAM J. Computing* 16(3), 421-444, 1987.
16. R. Tamassia and I. G. Tollis, Tesselation Representations of Planar Graphs, *Proc. 27th Annual Allerton Conf*, 1989.