Drawing Algorithms for Planar st-Graphs*

Roberto Tamassia
Department of Computer Science
Brown University
Providence, RI 02912-1910
rt@cs.brown.edu

Abstract

We present a survey on algorithms for drawing planar digraphs such that the arcs do not intersect and are monotonically increasing in the vertical direction (upward planar drawings). Various graphic standards (e.g., straight-line, polyline) and quality measures (e.g., area, bends) are considered.

1 Introduction

The problem of constructing aesthetically pleasing drawings of graphs has received increasing attention in the last years [6,18]. Various graphic standards exist for the representation of graphs in the plane. Usually, the vertices are represented by points and the edges by simple open curves. In a *polyline* drawing the edges are represented by polygonal chains. In a *straight-line* drawing the edges are represented by straight-line segments. A drawing is *planar* if no two edges intersect. A polyline drawing is a *grid* drawing if the vertices and the bends of the edges have integer coordinates. A drawing of a digraph is *upward* if each arc is a curve monotonically increasing in the vertical direction. Upward drawings are widely used to display hierarchic structures. Examples include PERT diagrams, ISA hierarchies, and subroutine-call graphs.

Typical quality measures for the readability of a drawing are the minimization of the number of crossings and the display of symmetries. In polyline drawings it is desirable to have a low number of bends. The minimization of the area of the drawing (defined as the area of the smallest convex polygon covering the drawing) is also important. Here, we assume that a *resolution rule* is given which implies a finite

*Research partially supported by Cadre Technologies, Inc.

minimum area for the drawing. For example, we may require integer coordinates for the vertices (grid drawing), or a minimum distance between any two vertices.

This paper surveys algorithms for constructing upward planar drawings of digraphs and related representations in the plane. We discuss the performance of the algorithms in terms of the aforementioned quality measures and of their time complexity. The digraphs which admit upward planar drawings are exactly the subgraphs of *planar st-graphs*, which are planar acyclic digraphs with exactly one source and exactly one sink, both on the external face.

Since planar undirected graphs can be augmented and oriented into planar *st*-graphs, and nonplanar graphs can be transformed into planar graphs by replacing crossings with fictitious vertices, the surveyed algorithms can be extended and modified to draw general graphs.

2 Planar st-graphs

Let G be an acyclic digraph. An arc (u,v) of G is *transitive* if there is another directed path in G from u to v. An acyclic digraph is said to be *reduced* if it has no transitive arcs. Notice that by removing all transitive arcs from G we obtain a reduced digraph with the same transitive closure as G. A *topological numbering* ξ of G maps every vertex v of G to a number $\xi(v)$ such that $\xi(u) < \xi(v)$ for every arc (u,v). Number $\xi(v)$ is often referred to as the *rank* of vertex v. A digraph G admits a topological numbering if and only if it is acyclic.

A *planar st-graph* is an acyclic planar digraph G with exactly one source s and exactly one sink t, embedded in the plane with vertices s and t on the boundary of the external face (see Figure 1). Planar *st*-graphs were first introduced in [11] in connection with a planarity testing algorithm. They have subsequently been used in a host of applications, dealing with partial orders [8,10], planar graph embedding [1,4, 17], graph planarization [13], floor planning [22], planar point location [7,14], visibility [12,16,19,20,21,23], motion planning [15], and VLSI layout compaction [22].

Lemma 1 *[19] Let G be a planar st-graph. The incoming arcs of each vertex v appear consecutively around v, and so do the outgoing arcs. Also, the boundary of each face f consists of two directed paths enclosing f, with common origin and destination.*

Let V, A, and F denote the set of vertices, arcs, and faces of G, respectively. We assume that set F contains two representatives for the external face of G, denoted s^* ("left external face") and t^* ("right external face"). For each arc a, we define $low(a)$ and $high(a)$ as the tail and head vertices of a, respectively. Also, we denote by $left(a)$

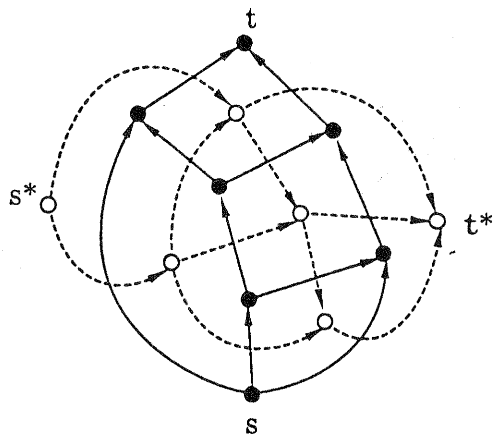

Figure 1: A planar *st*-graph G (solid lines) and its dual G^* (dashed lines).

the face to the left of a, and by $right(a)$ the face to the right of a when a is traversed from $low(a)$ to $high(a)$. If arc a has the external face to its left, we define $left(a) = s^*$, and if a has the external face to its right, we define $right(a) = t^*$. The *dual* graph G^* of a planar *st*-graph G is the embedded planar digraph with vertex set F and having an arc a^* from $left(a)$ to $right(a)$, for each $a \in A$ (see Figure 1).

Lemma 2 *[12] Let G be a planar st-graph. The dual graph G^* is a planar st-graph with source s^* and sink t^*.*

Notice the duality of the two separation properties expressed by Lemma 1. The face separating the incoming and outgoing arcs of a vertex v are called $left(v)$ and $right(v)$, respectively (see Figure 2.a). Also, the source and sink of the boundary of a face f are called $low(f)$ and $high(f)$, respectively (see Figure 2.b). The terminology can be extended by defining vertices $low(c)$ and $high(c)$ and faces $left(c)$ and $right(c)$ for each element c in $V \cup A \cup F$. For each vertex v, we define $low(v) = high(v) = v$ and $left(v)$ and $right(v)$ as above. For each face f, we define $low(f)$ and $high(f)$ as above and $left(f) = right(f) = f$.

Lemma 3 *[11] Let G be a planar st-graph. Every vertex v of G is on a simple path from s to t. Also, the digraph obtained from G by adding the arc (s,t) is planar and 2-connected.*

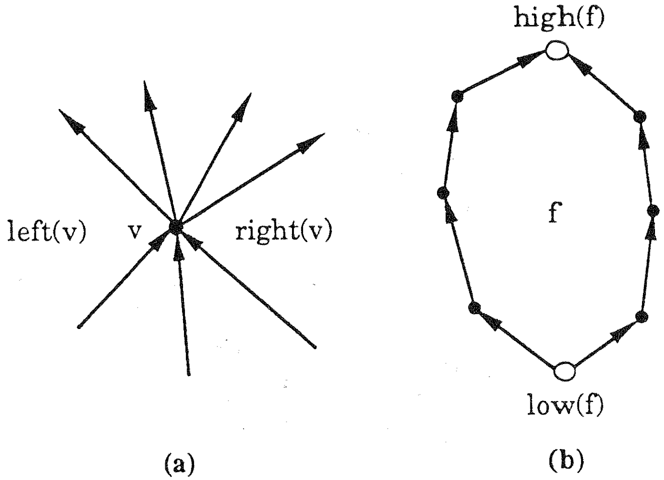

Figure 2: Separation of primal and dual incidences in a planar *st*-graph.

3 Upward planarity

Let G be a digraph. We say that G is *upward planar* if G admits an upward planar drawing. Clearly, an upward planar graph must be acyclic and planar. However, acyclicity and planarity are not sufficient to guarantee upward planarity, as shown in the example of Figure 3. In this section we present a characterization of upward planarity as subgraph containment in a planar *st*-graph.

Let Γ be a straight-line upward planar drawing of a digraph G. We denote with $\theta(a)$ the slope of arc a of Γ with respect to the x-axis. Since Γ is upward, $0 < \theta(a) < \pi$ for every arc a. Let $\theta_{\min}$ and $\theta_{\max}$ be the minimum and maximum slopes of the arcs on the external face of Γ. We say that Γ has *tolerance angle* α if the maximum deviation of the slope of any arc of Γ from the interval $[\theta_{\min}, \theta_{\max}]$ is upper bounded by α, i.e., $\max\{\theta(a) - \theta_{\min} , \theta_{\max} - \theta(a)\} \leq \alpha$, for each arc a.

Lemma 4 *[3] Let G be a maximal planar st-graph. Given a straight-line upward planar drawing Δ for the external face of G and a constant $\alpha > 0$, there exists a straight-line upward planar drawing Γ of G with external face Δ and tolerance angle α.*

Sketch of Proof: The proof is by induction on the number n of vertices of G. The basis of the induction, $n = 3$, is immediate. Now, assume that the theorem holds for

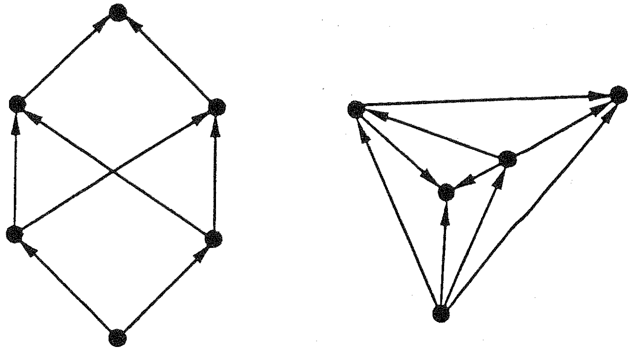

Figure 3: Planar acyclic digraphs that are not upward planar.

graphs with less than n vertices. Let v be a vertex of G that is not on the external face, and $v_0, \ldots, v_{k-1}$ be the neighbors of v, in circular order around v. Since G is maximal, these vertices form an undirected cycle χ. We distinguish two cases:

Case 1: χ has a chord, i.e., an arc (v_i, v_j) between two nonconsecutive vertices.
The undirected cycle $\lambda = (v_i, v, v_j)$ delimits two subgraphs of G, each a planar *st*-graph, where we denote the external one with G_1 and the internal one with G_2. By the inductive hypothesis we construct a straight-line upward planar drawing Γ_1 for G_1, with external face Δ and tolerance angle $\alpha/2$. Let Λ be the triangle in Γ_1 corresponding to the cycle λ. We use again the inductive hypothesis to construct a straight-line upward planar drawing Γ_2 for G_2, with external face Λ and tolerance angle $\alpha/2$. The union of Γ_1 and Γ_2 is a straight-line upward planar drawing Γ for G with the required properties.

Case 2: Otherwise.
Let v_i be a predecessor of v such that there is no directed path from v_i to any other predecessor of v. We contract arc (v_i, v) into vertex v_i. The resulting graph G' is a maximal planar *st*-graph. Now, we apply the inductive hypothesis to construct a straight-line upward planar drawing Γ' for G' with external face Δ and tolerance angle $\alpha/2$. Finally, we obtain a drawing Γ of G by suitable re-expanding vertex v_i into arc (v_i, v). □

Theorem 1 *[3,9] A digraph is upward planar if and only if it is a subgraph of a planar st-graph. Also, every upward planar digraph admits a straight-line upward planar drawing.*

4 Straight-line drawings

4.1 General planar st-graphs

We begin by showing how to construct a straight-line upward planar drawing of a maximal planar *st*-graph G.

Algorithm *Straight-Line-Draw [3]*

Input: maximal planar *st*-graph G;

Output: straight-line upward planar drawing Γ of G.

1. **Preprocessing:** Compute a topological numbering ξ of G and construct for each vertex v a balanced search tree $\mathcal{T}(v)$, called *rank tree* of v, which stores the arcs incident upon v sorted according to the rank of their other endpoint vertex. Notice that we can perform insertions, deletions, and searches in $\mathcal{T}(v)$ in $O(\log n)$ time, n being the number of vertices of G. Also, construct a doubly connected list $\mathcal{L}$, called the *candidate list*, storing the internal vertices of G with degree at most five.

2. **Drawing:** Call a recursive procedure that follows the steps of the proof of Lemma 4. We efficiently discriminate between Case 1 and Case 2 of Lemma 4 by choosing v from the candidate list $\mathcal{L}$, and then searching for chords using the rank trees. Since v has degree at most five, we need to perform only a constant number of such tree searches, with total time $O(\log n)$. In Case 1, in order to call the procedure recursively, we have to create the candidate lists $\mathcal{L}_1$ and $\mathcal{L}_2$ for the subgraphs G_1 and G_2. This is done by visiting in parallel G_1 and G_2, where at each visit step we perform a constant amount of work. Whenever we find a vertex that is in the original candidate list $\mathcal{L}$ of G we move it to the appropriate list $\mathcal{L}_1$ or $\mathcal{L}_2$. We terminate this parallel visit as soon as one of the two subgraphs is completely visited. At this point, we move to the candidate list of the other subgraph the remaining vertices in $\mathcal{L}$.

Theorem 2 *[3] Let G be a maximal planar st-graph with n vertices. Algorithm* Straight-Line-Draw *constructs a straight-line upward planar drawing Γ for G in $O(n \log n)$ time using $O(n)$ space.*

Since a planar *st*-graph with n vertices can be augmented into a maximal planar *st*-graph in $O(n)$ time, we have:

Corollary 1 *[3] Let G be a planar st-graph with n vertices. A straight-line upward planar drawing Γ for G can be constructed in $O(n \log n)$ time using $O(n)$ space.*

It should be noticed that the above algorithm uses arithmetic computations with real numbers, so that the drawing might have exponential area if a resolution rule is given. Indeed, as shown in Section 7, there is and exponential lower bound on the worst-case area requirement of straight-line upward planar drawings.

4.2 Reduced planar st-graphs

In this section we sketch an algorithm for constructing straight-line grid upward planar drawings of reduced planar *st*-graphs with the following remarkable properties: linear time complexity, quadratic area, detection and display of symmetries, and geometric characterization of the transitive closure by means of the dominance relation between the points associated with the vertices.

Algorithm *Dominance-Draw [5]*

Input: reduced planar *st*-graph G;

Output: straight-line upward planar grid drawing Γ of G.

1. **Preliminary Layout:** Assign to each vertex v distinct preliminary X- and Y-coordinates in the range $[0, n-1]$ by computing two topological numberings of the vertices of G. Such topological numberings are essentially obtained by scanning the outgoing arcs of each vertex from left to right and from right to left, respectively.

2. **Compaction:** Compute the final x-coordinates by scanning the vertices according to the order given by the preliminary X-coordinates. Let u and v be a pair of vertices with consecutive X-coordinates. In general, the final x-coordinate is not incremented if (u, v) is an arc, and is incremented otherwise. However, in the special case when (u, v) is the only outgoing arc of u and the only incoming arc of v, the x-coordinate is incremented. This is done to prevent the possibility that u and v be assigned the same pair of coordinates. The final y-coordinates are similarly computed from the preliminary Y-coordinates.

A run of algorithm *Dominance-Draw* is illustrated in Figure 4. Perhaps the best aesthetic result is obtained by a $\pi/4$ rotation of the axes. We say that a straight-line drawing of a digraph is a *dominance drawing* if for any two vertices u and v there is a directed path from u to v if and only if $x(u) \leq x(v)$ and $y(u) \leq y(v)$. Notice that these two conditions cannot be simultaneously satisfied with equality since distinct vertices must be placed at distinct points.

Theorem 3 *[5] Let G be a reduced planar st-graph with n vertices. Algorithm* Dominance-Draw *has $O(n)$ time complexity and constructs a planar dominance grid drawing Γ of G with $O(n^2)$ area.*

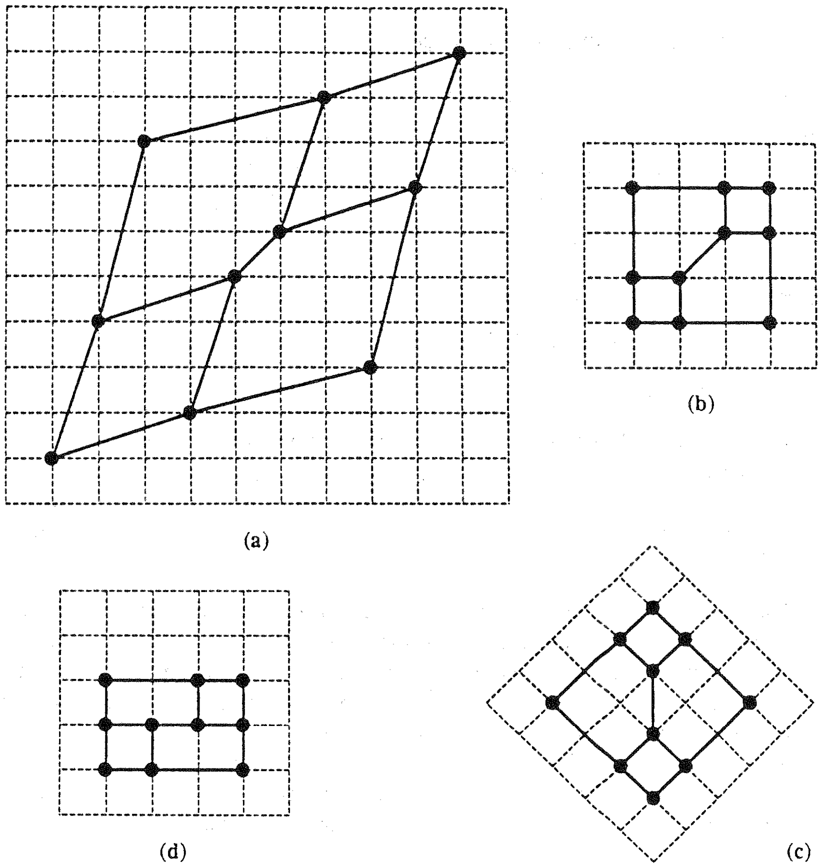

Figure 4: A run of Algorithm *Dominance-Draw*: (a) preliminary drawing; (b) final drawing; (c) final drawing rotated by a $\pi/4$ angle; (d) minimum area drawing.

A *pq-component* of a planar *st*-graph G is an induced subgraph G' of G such that (see Figure 5.a):

1. G' is a planar *pq*-graph, where the pair $\{p, q\}$ is a separation pair of G;
2. G' contains every vertex of G that is on some path from p to q;
3. G' contains every outgoing arc of p and every incoming arc of q.

Notice that G is a *pq*-component of itself, namely, its *st*-component.

Theorem 4 *[5] Let G be a reduced planar st-graph. The drawing* Γ *of G constructed by Algorithm* Dominance-Draw *displays all the geometric isomorphisms and coisomorphisms (i.e., translations, reflections, and rotations) of the pq-components of G.*

Figures 5.b-c show the drawing produced by algorithm *Dominance-Draw* for the planar *st*-graph of Figure 5.a. If the display of symmetries is not important, Algorithm *Dominance-Draw* can be modified so that it produces a minimum area drawing among all dominance drawings of G (see Figure 4.d).

5 Tessellation and visibility representations

In this section we present two drawing standards for planar *st*-graphs, called tessellation representation and visibility representation, where vertices and arcs are associated with isothetic rectangles/segments in the plane. These representations are interesting in their own and they are useful to generate polyline drawings.

5.1 Tessellation representations

A *tile* is a rectangle with sides parallel to the coordinate axes. A tile can be unbounded or degenerate to a segment or a point. Two tiles are *horizontally (vertically) adjacent* if they share a portion of a vertical (horizontal) side. The coordinates of a tile θ will be denoted by $x_{\min}(\theta)$, $x_{\max}(\theta)$, $y_{\min}(\theta)$, and $y_{\max}(\theta)$.

Let G be a planar *st*-graph. As usual, we denote by V, A, and F the sets of vertices, arcs, and faces of G, respectively. (Recall that F has two "external faces", s^* and t^*.) An element $c \in V \cup A \cup F$ is called a *constituent* of G. A *tessellation representation* Θ for G maps each constituent (vertex, arc, or face) c of G into a tile $\Theta(c)$ such that (see Figure 6.b):

1. The interiors of tiles $\Theta(c)$ and $\Theta(d)$ are disjoint whenever $c \neq d$.

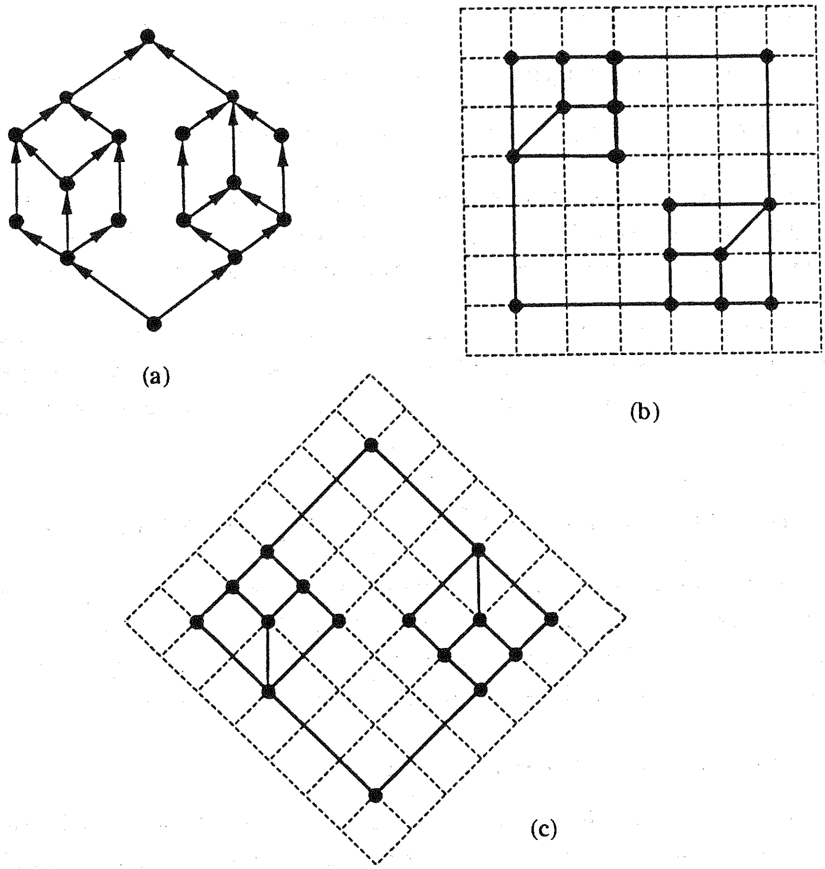

Figure 5: (a) A planar *st*-graph G with two rotationally isomorphic *pq*-components. (b) Drawing of G constructed by algorithm *Dominance-Draw*. (c) Rotated drawing.

2. The union of all tiles $\Theta(c)$, $c \in V \cup A \cup F$, is a tile.

3. Tiles $\Theta(c)$ and $\Theta(d)$ are horizontally adjacent if and only if:

$$c = \mathit{left}(d) \text{ or } c = \mathit{right}(d) \text{ or } d = \mathit{left}(c) \text{ or } d = \mathit{right}(c).$$

4. Tiles $\Theta(c)$ and $\Theta(d)$ are vertically adjacent if and only if:

$$c = \mathit{low}(d) \text{ or } c = \mathit{high}(d) \text{ or } d = \mathit{low}(c) \text{ or } d = \mathit{high}(c).$$

The following algorithm constructs a tessellation representation Θ for a planar *st*-graph G such that the tiles associated with vertices and faces degenerate to segments.

Algorithm *Tessellation-Draw [21]*

Input: planar *st*-graph G;

Output: tessellation representation Θ for G.

1. Compute a topological numbering Y of G.
2. Compute a topological numbering X of G^*.
3. For each constituent $c \in V \cup A \cup F$, let the coordinates of tile $\Theta(c)$ be:

$$x_{\min}(c) = X(\mathit{left}(c)), \quad x_{\max}(c) = X(\mathit{right}(c)),$$
$$y_{\min}(c) = Y(\mathit{low}(c)), \quad y_{\max}(c) = Y(\mathit{high}(c)).$$

An example of a run of algorithm *Tessellation-Draw* is shown in Figure 6.

Theorem 5 *[21] Let G be a planar st-graph with n vertices. Algorithm* Tessellation-Draw *correctly constructs a tessellation representation Θ of G in $O(n)$ time.*

Theorem 6 *Given a planar st-graph G with n vertices and nonnegative numbers $h(c)$ and $w(c)$, for each constituent c of G, a minimum area tessellation representation Θ for G such that each tile $\Theta(c)$ has height at least $h(c)$ and width at least $w(c)$ can be constructed in time $O(n)$.*

Corollary 2 *Given a planar st-graph G with n vertices, a minimum area grid tessellation representation Θ of G such that no tile is degenerate can be computed in $O(n)$ time. Also, the area of Θ is $O(n^2)$.*

Corollary 3 *Given a planar st-graph G with n vertices, a minimum area grid tessellation representation Θ of G such that the vertex- and face-tiles are degenerate and the arc-tiles are nondegenerate can be computed in $O(n)$ time. Also, the area of Θ is $O(n^2)$.*

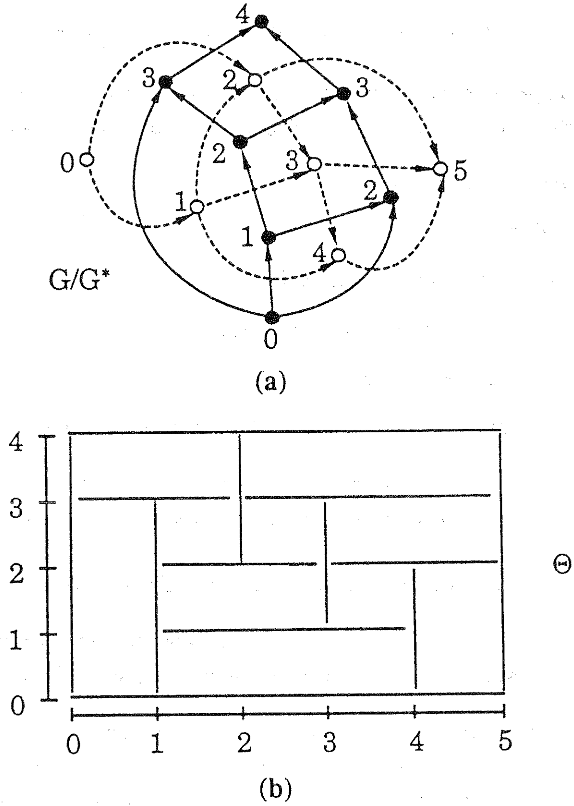

Figure 6: Example of a run of algorithm *Tessellation-Draw*: (a) graphs G and G^* labeled by Y and X; (b) tessellation representation Θ constructed by the algorithm.

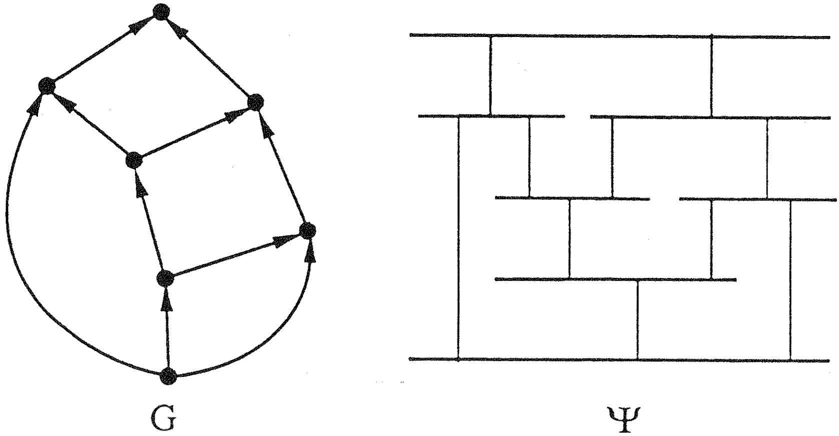

Figure 7: Example of a visibility representation.

5.2 Visibility representations

A *visibility representation* Ψ for a planar *st*-graph G maps each vertex v into a horizontal segment $\Psi(v)$, and each arc a into a vertical segment $\Psi(a)$ such that (see Figure 7):

1. segments $\Psi(u)$ and $\Psi(v)$ are disjoint for distinct vertices u and v;
2. segment $\Psi(a)$ has its lower endpoint on $\Psi(low(a))$, its upper endpoint on $\Psi(high(a))$ and does not intersect any other segment.

A visibility representation can be viewed as a straight-line drawing where each arc is a vertical segment, and each vertex has been stretched into a horizontal segment. Algorithms for constructing visibility representations are given in [3,12,16, 19]. The following algorithm constructs a visibility representation from a tessellation representation.

Algorithm *Visibility-Draw*

Input: planar *st*-graph G;

Output: visibility representation Ψ for G.

1. Construct a tessellation representation Θ for G.

2. For each vertex v, let $\Psi(v)$ be equal to a horizontal segment extending from the left side to the right side of tile $\Theta(v)$.

3. For each arc a, let $\Psi(a)$ be equal to any vertical segment intersecting the interior of tile $\Theta(a)$ and extending from $\Psi(low(a))$ to $\Psi(high(a))$.

Theorem 7 *Let G be a planar st-graph with n vertices. Algorithm* Visibility-Draw *correctly constructs a visibility representation Ψ for G in $O(n)$ time.*

Corollary 4 *Let G be a planar st-graph with n vertices. A minimum area grid visibility representation Ψ for G can be computed in $O(n)$ time. Also, the area of Ψ is $O(n^2)$.*

An interesting feature of visibility representations is the possibility of aligning arcs. Let G be a planar *st*-graph. Two (directed) paths π_1 and π_2 of G are said to be *nonintersecting* if they are arc-disjoint and they do not cross at common vertices. Let Π be a collection of nonintersecting paths of a planar *st*-graph G. It is possible to construct a visibility representation of G such that the arcs of every path in Π are vertically aligned.

Theorem 8 *[2] Let G be a planar st-graph with n vertices, and Π a set of nonintersecting paths covering the arcs of G. A grid visibility representation for G with $O(n^2)$ area such that the arcs of every path π in Π are vertically aligned can be computed in $O(n)$ time.*

6 Polyline drawings

In this section we present two algorithms for constructing polyline upward planar drawings of planar *st*-graphs. The first algorithm transforms a visibility representation into a polyline drawing. The second algorithm uses the technique for domination drawings of reduced planar *st*-graphs.

Algorithm *Polyline-Draw-1 [3]*

Input: planar *st*-graph G;

Output: polyline upward planar drawing Γ of G.

1. Construct a minimum area grid visibility representation Ψ for G according to Corollary 4. We denote with $Y(v)$ the y-coordinate of vertex-segment $\Psi(v)$, and with $X(a)$ the x-coordinate of arc-segment $\Psi(a)$. An arc a will be called *short* if the arc-segment $\Psi(a)$ has length 1, and *long* otherwise.

2. **foreach** vertex v **do begin**
   let a be a long arc incident upon v, if one exists, or otherwise any arc incident upon v;
   let $\Gamma(v)$ be the intersection of $\Psi(v)$ and $\Psi(a)$ **end**

3. **foreach** arc a **do begin**
   if a is short
   then let $\Gamma(a)$ be the segment $\Gamma(low(a)) \mapsto \Gamma(high(a))$
   else let $\Gamma(a)$ be the polygonal line:
   $\Gamma(low(a)) \mapsto (X(a), Y(low(a))+1) \mapsto$
   $\mapsto (X(a), Y(high(a))-1) \mapsto \Gamma(high(a))$ **end**

Theorem 9 *[3] Let G be a planar st-graph with n vertices. Algorithm* Polyline-Draw-1 *constructs in $O(n)$ time a polyline grid upward planar drawing Γ for G such that:*

1. $\Gamma(a)$ has at most two bends, for each arc a;

2. Γ has a total of no more than $(10n-31)/3$ bends;

3. Γ has $O(n^2)$ area.

Theorem 10 *[2] Let G be a planar st-graph with n vertices, and Π a set of nonintersecting paths of G. A polyline grid upward planar drawing Γ for G such that*

1. Γ has a total of no more than $4n-10$ bends;

2. Γ has $O(n^2)$ area;

3. the arcs of every path π in Π are vertically aligned

can be computed in $O(n)$ time.

The following algorithm uses the technique for straight-line drawings of reduced planar *st*-graphs.

Algorithm *Polyline-Draw-2 [5]*

Input: planar *st*-graph G;

Output: polyline upward planar drawing Γ of G.

1. Construct a reduced planar *st*-graph G' by replacing each transitive arc a with the chain $(low(a), v_a, high(a))$, where v_a is a new vertex, called a *reduction vertex*.

2. Compute a dominance drawing Γ' of G' using algorithm *Dominance-Draw*.

3. Construct from Γ' a polyline drawing Γ of G by replacing each reduction vertex $\Gamma'(v_a)$ with a bend.

Theorem 11 *[5] Let G be a planar st-graph with n vertices. Algorithm* Polyline-Draw-2 *constructs in $O(n)$ time a polyline grid upward planar drawing Γ for G such that:*

1. $\Gamma(a)$ has at most one bend if a is a transitive arc, and no bends otherwise;

2. Γ has a total of no more than $2n - 5$ bends;

3. Γ has $O(n^2)$ area.

7 Area requirement of planar st-graphs

In this section we present a class of planar *st*-graphs which require exponential area in any straight-line upward planar drawing. This class is recursively defined as follows (see Figure 8). G_0, shown in Figure 8.a, consists of two vertices, s_0 and t_0, and a single arc, (s_0, t_0), directed from s_0 to t_0. G_1, shown in Figure 8.b, consists of G_0 plus vertices s_1 and t_1 and arcs (s_1, s_0), (t_0, t_1), (s_1, t_0) and (s_0, t_1). In general, G_n is constructed from G_{n-1} by adding vertices s_n and t_n, and arcs (s_n, s_{n-1}), (t_{n-1}, t_n), (s_n, t_{n-1}) and (s_{n-1}, t_n), with the embedding shown in Figure 8.c.

It is easy to verify that G_n is a planar $s_n t_n$-graph with $2n + 2$ vertices and $4n + 1$ arcs. Notice that the arcs (s_i, t_{i-1}) are always embedded "on the right" and the arcs (s_{i-1}, t_i) are always embedded "on the left".

Theorem 12 *[5] Given a resolution rule, let A_n be the minimum area of a straight-line upward planar drawing of G_n that preserves the embedding. Then $A_n = \Omega(2^n)$.*

Sketch of Proof: We show by induction that $A_n \geq 4A_{n-2}$. Since $A_2 \geq c$, for some constant c depending on the resolution rule, this implies the claimed result. □

The restriction that the drawing of G_n preserves the given embedding can be removed since the addition of arcs (s_{i-2}, t_i) and (s_i, t_{i-2}) $(2 \leq i \leq n)$ ensures that G_n is 3-connected for $n \geq 2$, and 3-connected graphs have a unique embedding. Hence, we have:

Theorem 13 *[5] A straight-line upward planar drawing of G_n has area $\Omega(2^n)$ under any resolution rule.*

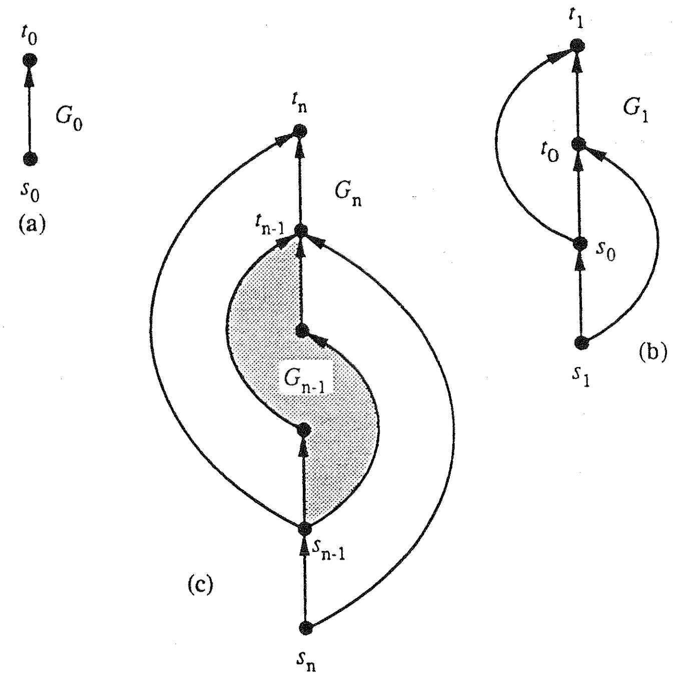

Figure 8: Recursive definition of planar *st*-graph G_n.

Notice that the graph G_n achieving exponential area requirement has $2n - 2$ transitive arcs. Indeed, as shown in Theorem 3, a reduced planar *st*-graph admits a straight-line drawing with quadratic area. If we allow bends, we have an $\Omega(n^2)$ worst-case lower bound on the area of planar polyline drawings, even if we drop the upward requirement [24].

8 Open problems

We conclude the paper with the following open problems:

1. Find an $O(n)$-time algorithm for constructing a straight-line upward planar drawing of an n-vertex planar *st*-graph, or provide an $\Omega(n \log n)$ lower bound.

2. Find a polynomial-time algorithm for testing whether a digraph G is upward planar (i.e., whether G is a subgraph of a planar *st*-graph), or show that the problem is NP-complete.

3. Study the tradeoff between area and number of bends in polyline upward planar drawings.

Acknowledgments

I would like to thank Carlo Batini, Giuseppe Di Battista, Peter Eades, and Ioannis Tollis for useful discussions.

References

[1] N. Chiba, T. Nishizeki, S. Abe & T. Ozawa, "A Linear Algorithm for Embedding Planar Graphs Using PQ-Trees," *J. of Computer and System Sciences* 30 (1985), 54–76.

[2] G. Di Battista, E. Pietrosanti, R. Tamassia & I.G. Tollis, "Automatic Layout of PERT Diagrams with XPERT," *Proc. IEEE Workshop on Visual Languages (VL'89)* (1989), 171–176.

[3] G. Di Battista & R. Tamassia, "Algorithms for Plane Representations of Acyclic Digraphs," *Theoretical Computer Science* 61 (1988), 175–198.

[4] G. Di Battista & R. Tamassia, "Incremental Planarity Testing," *Proc. 30th IEEE Symp. on Foundations of Computer Science* (1989), 436–441.

[5] G. Di Battista, R. Tamassia & I.G. Tollis, "Area Requirement and Symmetry Display in Drawing Graphs," *Proc. ACM Symp. on Computational Geometry* (1989), 51–60.

[6] P. Eades & R. Tamassia, "Algorithms for Automatic Graph Drawing: An Annotated Bibliography," Dept. of Computer Science, Brown Univ., Technical Report CS-89-09, 1989.

[7] H. Edelsbrunner, L.J. Guibas & J. Stolfi, "Optimal Point Location in a Monotone Subdivision," *SIAM J. Computing* 15 (1986), 317–340.

[8] T. Kameda, "On the Vector Representation of the Reachability in Planar Directed Graphs," *Information Processing Letters* 3 (1975), 75–77.

[9] D. Kelly, "Fundamentals of Planar Ordered Sets," *Discrete Mathematics* 63 (1987), 197–216.

[10] D. Kelly & I. Rival, "Planar Lattices," *Canadian J. Mathematics* 27 (1975), 636–665.

[11] A. Lempel, S. Even & I. Cederbaum, "An Algorithm for Planarity Testing of Graphs," *Theory of Graphs, Int. Symposium* (1966), 215–232.

[12] R.H.J.M. Otten & J.G. van Wijk, "Graph Representations in Interactive Layout Design," *Proc. IEEE Int. Symp. on Circuits and Systems* (1978), 914–918.

[13] T. Ozawa & H. Takahashi, "A Graph-planarization Algorithm and its Applications to Random Graphs," *Graph Theory and Algorithms, Lecture Notes in Computer Science* 108 (1981), 95–107.

[14] F.P. Preparata & R. Tamassia, "Fully Dynamic Point Location in a Monotone Subdivision," *SIAM J. Computing* 18 (1989), 811–830.

[15] I. Rival & J. Urrutia, "Representing Orders by Translating Convex Figures in the Plane," *Order* 4 (1988), 319–339.

[16] P. Rosenstiehl & R.E. Tarjan, "Rectilinear Planar Layouts of Planar Graphs and Bipolar Orientations," *Discrete & Computational Geometry* 1 (1986), 343–353.

[17] R. Tamassia, "A Dynamic Data Structure for Planar Graph Embedding," *Automata, Languages and Programming (Proc. 15th ICALP, Tampere, Finland)* (1988), 576–590.

[18] R. Tamassia, G. Di Battista & C. Batini, "Automatic Graph Drawing and Readability of Diagrams," *IEEE Transactions on Systems, Man and Cybernetics* SMC-18 (1988), 61–79.

[19] R. Tamassia & I.G. Tollis, "A Unified Approach to Visibility Representations of Planar Graphs," *Discrete & Computational Geometry* 1 (1986), 321–341.

[20] R. Tamassia & I.G. Tollis, "Centipede Graphs and Visibility on a Cylinder," in *Graph-Theoretic Concepts in Computer Science, (Proc. Int. Workshop WG '86, Bernierd, June 1986)*, G. Tinhofer and G. Schmidt, ed., Lecture Notes in Computer Science, vol. 246, Springer-Verlag, 1987, 252–263.

[21] R. Tamassia & I.G. Tollis, "Tessellation Representations of Planar Graphs," *Proc. 27th Annual Allerton Conf.* (1989).

[22] S. Wimer, I. Koren & I. Cederbaum, "Floorplans, Planar Graphs, and Layouts," *IEEE Trans. on Circuits and Systems* 35 (1988), 267–278.

[23] S.K. Wismath, "Characterizing Bar Line-of-Sight Graphs," *Proc. ACM Symp. on Computational Geometry* (1985), 147–152.

[24] D. Woods, "Drawing Planar Graphs," Computer Science Dept., Stanford Univ., Ph.D. dissertation (Technical Report STAN-CS-82-943), 1982.