On Drawing Angle Graphs*

(Extended Abstract)

Ashim Garg
Department of Computer Science
Brown University
Providence, RI 02912–1910, USA

Abstract. An *Angle graph* is a graph with a fixed cyclic order of edges around each vertex and an angle specified for every pair of consecutive edges incident on each vertex. We study the problem of constructing a drawing of an angle graph that preserves its angles, and present several new results.

1 Introduction

An *angle graph* is a graph with a fixed cyclic order of edges around each vertex and an angle (between 0° and 360°) specified for every pair of consecutive edges incident on each vertex such that the sum of angles around every vertex is 360°. A *Rectilinear* angle graph is an angle graph in which each angle is a multiple of 90°.

A graph drawing algorithm takes a graph as its input and constructs a drawing of the graph (see [3] for an extensive survey). Some well known algorithms such as Tamassia's bend-minimization algorithm [16] and the visibility representation algorithm by Tamassia and Tollis [17] use rectilinear angle graphs as intermediate stage products. Infact the bend-minimization algorithm has been found to give aesthetically pleasing drawings in practice [11]. Our hope is that a good characterization of planar angle graphs may allow us to use the bend minimization algorithm to construct aesthetically pleasing drawings of any planar graph, not just orthogonal planar graphs.

A recent trend in the area of graph drawing is towards developing systems which allow user to specify arbitrary constraints on the positioning of vertices and edges in the drawings. See for example, [2, 6, 12]. Since a user may specify angle constraints, a study of angle graphs is important for such systems.

A very important aesthetic criteria for a straight line drawing (edges drawn as straight lines) is that the angles between the consecutive edges incident on each vertex be large. This is formalized by defining *angular resolution* of a drawing [7, 15]. *Angular resolution* of a straight line drawing of a graph is the minimum angle in the drawing between any two consecutive edges incident on the same vertex.

Studying angle graphs can provide important clues in constructing drawings with large angular resolution. For example we use our result, that testing an angle graph for planarity is NP-hard, to show that, given a triconnected planar graph G and an angle α, determining whether G admits a planar straight line drawing with angular resolution at least α is NP-hard.

Angles have been found to be useful in characterizing planar graphs (see [5], for example). The study of angle graphs is also of theoretical interest in itself

* Research supported in part by the National Science Foundation, by the U.S. Army Research Office, by the Office of Naval Research and the Defense Advanced Research Projects Agency.

because they require an extensive use of plane geometry in their analysis. Most of the known graph drawing algorithms use graph theoretic concepts such as orientation, coloring, flow etc. Recent results [9] however shows that plane geometry is a useful tool by successfully applying geometric techniques in deriving many results on planar drawings with large angular resolution. A study of angle graphs can provide an invaluable insight into the use of plane geometry for graph drawing algorithms and stimulate research in this relatively unexplored area.

We now give some definitions. Let A be an angle graph. A *drawing* of A is a mapping of its vertices to points in the plane and its edges to straight lines joining their endpoints so that the angles of A are preserved in the drawing. A *planar* drawing of A has no edge crossings. Unlike general graphs, an angle graph may not have a drawing at all. Fig 1(a) shows such an angle graph (the angles between edges are as given in the figure). Fig 1(b) shows an angle graph that has a drawing. A is called *consistent* if it has a drawing. A is called *planar* if it has a planar drawing.The graph isomorphic to A if we drop the angle constraints is called the *underlying graph* of A. An *angle-cycle* (*outerplanar* angle graph, *series-parallel* angle graph, resp.) is an angle graph whose underlying graph is a cycle (outerplanar graph, series-parallel graph, resp.).

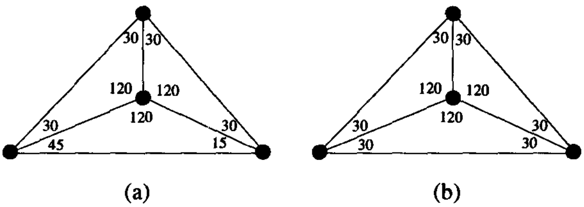


Figure 1: An angle graph (a) that is not consistent; (b) that is consistent.

The pioneering work in the area of angle graphs was done by Vijayan in [19]. Vijayan gave a linear program for testing an angle graph for consistency. Some necessary conditions for an angle graph to be planar, as well as necessary and sufficient conditions for some special angle graphs such as angle cycles, angle graphs with convex faces, outer planar angle graphs and rectilinear angle graphs have also been described in [19].Vijayan also posed some conjectures about the planarity of the angle graphs. A characterization of planar angle graphs with triangular faces by a set of non-linear equalities is given in [5]. [20] gives a linear time algorithm for testing a rectilinear angle graph for planarity and quadratic time algorithm for constructing a planar drawing if it is planar.

1.1 Our Results

We give several new results concerning drawing of angle graphs. We first study the planarity of angle graphs. We disprove the conjectures of [19] by providing counter examples to them. We then show that testing a consistent angle graph for planarity is NP-hard. We also show that given a *triconnected* planar graph and an angle α, determining whether it admits a planar straight line drawing with angular resolution at least α is NP-hard. Our result therefore strengthens a similar result given by Kant [13] that holds for *biconnected* planar graphs.

We study the problem of testing a series-parallel angle graph for consistency and provide a linear time algorithm for solving the problem.

The study of area requirements of drawings of graphs has received a lot of attention (see, e.g., [1, 4, 8, 9]) and is motivated by the finite resolution of the graph drawing technologies and circuit-area optimization criteria of VLSI layouts [14, 18]. We study the area requirement of angle graphs and show that there exists a family of angle graphs requiring exponential area in drawing.

Finally we consider the *multiplanarity* problem of angle graphs. A *multilayered* angle graph is an angle graph in which each edge is assigned to one of several *layers*. A *multiplanar* angle graph is a multilayered angle graph if it has a drawing in which edges assigned to the same layer do not cross. We show that very surprisingly testing even a bilayered rectilinear angle graph (edges assigned to only two layers) for biplanarity is NP-hard. This is in sharp contrast to the linear time complexity of testing a monolayer rectilinear graph for planarity [20].

2 Planarity Testing of Angle Graphs

Vijayan [19] made following conjectures (see [19] for definitions of the terms used in the conjecture. We do not describe them here for the lack of space):

Conjecture 1 (Vijayan [19]) *A consistent angle graph A that does not contain any angle-cycle which is a subdivision of a triangle is planar if and only if (a) the faces of each biconnected component are consistent angle-cycles, (b) no two biconnected components interlace at an articulation vertex of A, and (c) the force inside relation among the biconnected components is a partial order.*

Conjecture 2 (Vijayan [19]) *The conditions stated in the Conjecture 1 are necessary and sufficient for consistent angle graphs that do not contain subdivisions of triangles and whose biconnected components have convex interior faces.*

We give a counter example to Conjecture 2, which in turns disproves Conjecture 1. Consider the angle graph A shown in Fig. 2(a). u is an articulation vertex of A. A has two biconnected components B_1 and B_2. Both B_1 and B_2 have convex interior faces and angle graph A does not contain subdivisions of triangles. By simple geometric considerations we can show that A is not planar.

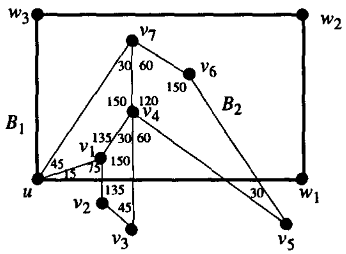


Figure 2: Counter example to Conjecture 2: Angle graph A.

2.1 NP-Hardness of Planarity Testing

We now show that testing a consistent angle graph for planarity is NP-hard by reducing the following version of the $3SAT$ problem to it:

given a set $X = \{x_1, x_2, \ldots, x_n\}$ of variables and a set $C = \{c_1, c_2, \ldots, c_m\}$ of clauses over X such that every clause has three literals, each variable occurs in at most five clauses either negated or un-negated [10] and in none of them both as negated and un-negated, is there a satisfying truth assignment for C?

We construct an angle graph A such that there is a satisfying truth assignment of C if and only if A is planar. We need the following gadgets (which are angle graphs) for our construction. We denote the distance between two vertices u and v in a drawing of A by $d(u, v)$.

- **Horse shoe:** We use the horse shoe to represent the variables. A *horse shoe* H is shown in Fig. 3(a). The vertices $\{x_0, x_1, \ldots, x_9\}$ and $\{y_0, y_1, \ldots, y_9\}$ are called the *output* vertices of the faces F_x and F_y respectively. Vertices x_i and x_{i+1} (y_i and y_{i+1}) for even values of i are *siblings* of each other. Edges e_x, e_y and e_t are called the *left attachment*, *right attachment* and *top edge* respectively of the horse shoe. In any planar drawing Δ of H.
  Observation 1: the attachments of H have same length as its top edge.
  Observation 2: if the top edge of H has unit length then at most one of $d(x_0, x_9)$ and $d(y_0, y_9)$ is at least one unit. H is *left heavy* in Δ if $d(x_0, x_9) \geq 1$ (Fig. 3(b)) and is *right heavy* in Δ if $d(y_0, y_9) \geq 1$ (Fig. 3(c)).

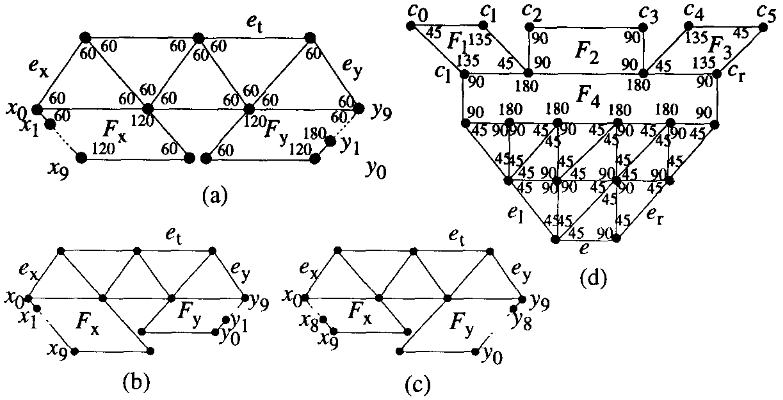
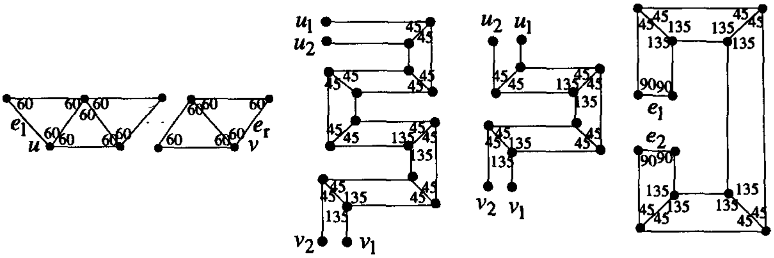


Figure 3: (a) A horse shoe H; (b) Left heavy horse shoe; (c) Right heavy horse shoe; (d) A Crown.

- **Crown:** We use the crown to represent the clauses. Fig. 3(d) shows a *crown*. Edges e_l and e_r are called the *left attachment* and *right attachment* respectively of the crown. Edge e is the *base* of the crown.
  Observation 3: If the base has unit length then $\sum_{0 \leq i \leq 2} d(c_{2i}, c_{2i+1}) = 5$.
- **Beam:** A *Beam* B is shown in Fig. 4(a). A beam consists of 123 equilateral triangles. Edges e_l and e_r are called the *left attachment* and *right attachment* respectively of the beam. In any planar drawing of B,
  Observation 4: All its edges are of equal length.
  Observation 5: $d(u, v) = 61$ if the length of an attachment is one unit.
- **Wiggle:** A *wiggle* W is shown in Fig. 4(b). Vertices u_1 and u_2 (v_1 and v_2) are called the *input* (*output*) vertices of the wiggle. In any planar drawing of W,

Observation 6: The horizontal distance between its output vertices equals the vertical distance between its input vertices.
Observation 7: The structure of this gadget does not place any restrictions on the horizontal distance between its input and output vertices.



(a) (b) (c) (d)

Figure 4: (*a*) A Beam; (*b*) A Wiggle; (*c*) A Connector; (*d*) A Holder.

- **Connector:** A *connector* C is a subdivision of the angle graph shown in Fig. 4(*c*). Vertices u_1 and u_2 (v_1 and v_2) are called the *input* (*output*) vertices of C. In any planar drawing of C,
  Observation 8: The horizontal distance between its output vertices equals the horizontal distance between its input vertices.
  Observation 9: Same as Observation 7.
- **Holder:** A holder H is shown in Fig. 4(*d*). Edges e_1 and e_2 are called its *upper clamp* and *lower clamp* respectively. In any planar drawing of H,
  Observation 10: The upper and lower clamps of H are of same length.

Fig 5(*a*) gives a high level view of A. We can identify three important subgraphs of A, namely *variable* subgraph, *connection* subgraph and *clause* subgraph. The variable and connection subgraphs are "connected" through some wiggles, the connection and clause subgraphs are "connected" through some wiggles. The variable and clause subgraphs are "connected" through a single holder R.

The *variable* subgraph V_n of A can be described recursively: V_1 is a horse shoe X_1; V_n is constructed from V_{n-1} by "connecting" the horse shoe X_{n-1} (present in V_{n-1}) with a horse shoe X_n using a beam B as shown in Fig 6(*a*). We say that horse shoe X_i *represents* variable x_i.

The *clause* subgraph S_m of A can be described recursively: S_1 is a crown C_1; S_m is constructed from S_{m-1} by "connecting" the crown C_{m-1} (present in S_{m-1}) with a crown C_m using a beam B and two edges e' and e'' as shown in Fig 6(*b*). We say that crown C_j *represents* clause c_j.

The *connection* subgraph D is constructed in following steps:

1. Let G be a bipartite graph whose vertices can be partitioned into two sets P and Q. There is a vertex $x_{i,j}$ in P if literal x_i occurs in clause c_j. There is a vertex $y_{i,j}$ in P if literal $\bar{x}_i$ occurs in clause c_j. $x_{i,j}$ ($y_{i,j}$) is called an *image* of the literal x_i ($\bar{x}_i$). Q has a vertex $c_{j,i}$ if either literal x_i or $\bar{x}_i$ occurs in clause c_j (notice that both of them can not occur simultaneously in c_j). $c_{j,i}$ is called an *image* of clause c_j. There is an edge in G between $x_{i,j}$ (or $y_{i,j}$) and $c_{j,i}$. Construct a drawing Δ as shown in Fig. 7(*a*) in which the vertices

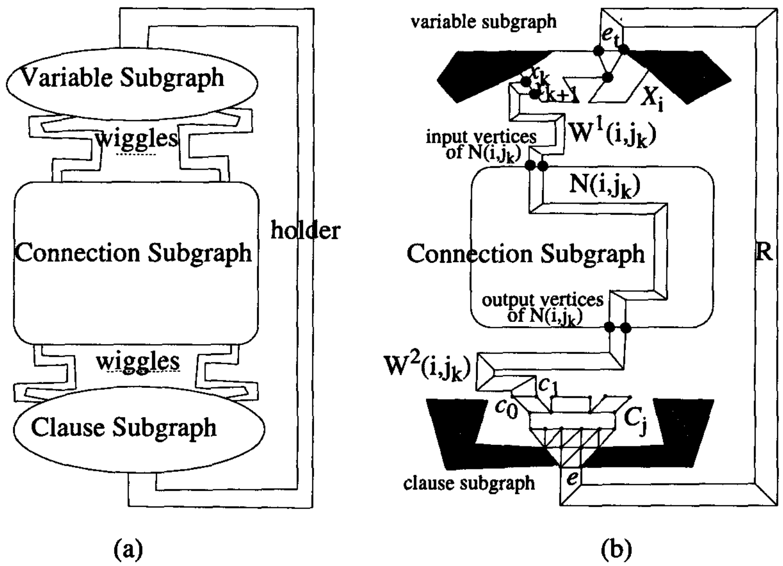


Figure 5: (*a*) A High Level View of the Angle Graph A; (*b*) Constructing angle graph A: "connecting" siblings x_k and x_{k+1} in horse shoe X_i with $N(i, j_k)$ by wiggle $W^1(i, j_k)$, "connecting" c_0 and c_1 in crown C_j with $N(i, j_k)$ by wiggle $W^2(i, j_k)$ and 'connecting" Horse shoe X_i with Crown C_j with a Holder R.

of P are placed on the same horizontal level and so are the vertices of Q.

2. Replace each line-segment $(x_{i,j}, c_{j,i})$ or $(y_{i,j}, c_{j,i})$ in Δ by the drawing of a connector $N(i, j)$ (this also means replacing $x_{i,j}$ and $c_{j,i}$ by the input and output vertices resp. of $N(i, j)$). Fig. 7(*b*) shows the resultant drawing Δ'.
3. Construct a planar angle graph D by replacing the crossings in Δ' by vertices.

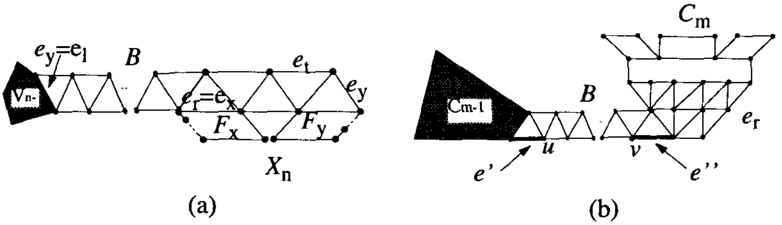


Figure 6: (*a*) Attaching V_{n-1} and a Horse shoe X_n to Construct Variable Subgraph V_n; (*b*) Attaching S_{m-1} and a Crown C_m to Construct Clause Subgraph S_m.

Angle Graph A We now complete the description of angle graph A.

- Suppose literal l_i where $l = x$ or $l = y = \bar{x}$ occurs in clauses $c_{j_0}, c_{j_1}, \ldots, c_{j_r}$, where $j_k < j_{k+1}$ and $r < 5$. "Connect" siblings l_k and l_{k+1} in horse shoe X_i (recall that X_i represents variable x_i) with connector $N(i, j_k)$ using a wiggle $W^1(i, j_k)$ (see Fig. 5(*b*) which shows connection of x_k and x_{k+1}). Edge (l_k, l_{k+1}) is called the *image* of clause c_{j_k} in X_i.
- Suppose variables x_{i_0}, x_{i_1}, and x_{i_2} where $i_0 < i_1 < i_2$ occur in a clause c_j either negated or un-negated. "Connect" c_{2k} and c_{2k+1} in crown C_j with

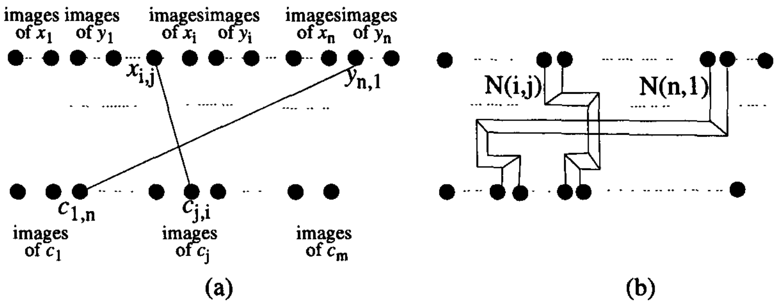


Figure 7: Bipartite graph G: (a) A drawing Δ of G; (b) Drawing Δ'.

connector $N(i, j_k)$ using a wiggle $W^2(i, j_k)$ (Fig. 5(b) shows connection with c_0 and c_1). Edge (c_k, c_{k+1}) is called the *image* of variable x_{i_k} in C_j.

- Now choose any horse shoe X_i and crown C_j, and "connect" X_i with C_j by identifying the upper clamp of a holder R with the top edge of X_i and the lower clamp of R with the base of C_j (see Fig. 5(b)).

Theorem 1. *There is a satisfying truth assignment of C if and only if A is planar.*

Sketch of Proof:

If: Suppose A is planar. Let Δ be a planar drawing of A. Let the clamps of the holder have unit length in Δ. Every X_i is either left heavy or right heavy in Δ (Observation 2). Set variable x_i to `true` if X_i is left heavy and to `false` otherwise. Suppose variable x_i occurs in clause c_j. From Observations 6 and 8, the length of the image of x_i in C_j is equal to $\sqrt{3}/2$ times the length of the image of c_j in X_i. Therefore the image of x_i in C_j has at least $\sqrt{3}/2$ units length only if literal x_i or $\bar{x}_i$ that occurs in c_j is `true`. From Observation 3 at least one of the images of the variables occurring in C_j has length at least $\sqrt{3}/2$ units. Consequently C_j is satisfiable.

Only If: It is easy to see that because the length of a beam is at least 61 units in any drawing of A, (the length of the clamps of the holder is taken to be one unit), the greater of $d(x_0, x_9)$ and $d(y_0, y_9)$ in any X_i can be $5(2/\sqrt{3})$ units without creating any crossings. Wiggles (Observation 7) provide enough flexibility to construct a planar drawing for A. □

Theorem 2. *The problem of testing whether a consistent angle graph is planar is NP-hard.*

All the angles used in our reduction were multiples of 15°. Therefore,

Corollary 3. *The problem of testing whether a consistent angle graph is planar is NP-hard even if the angles specified for every pair of consecutive edges incident on a vertex in the angle graph is a multiple of some integer $\alpha \neq 0$.*

3 Testing Triconnected Planar Graphs for High Angular Resolution

We now consider the problem of constructing planar straight line drawings of planar graphs with high angular resolution and show that the following problem is NP-hard:

Given a triconnected planar graph G and an angle α, determine whether G has a planar straight line drawing with angular resolution at least α.

We use the consistent angle graph A described in Section 2.1 to show that this problem is NP-hard. Let H be the underlying graph of A. Our approach is to convert H into a triconnected planar graph G in polynomial time by adding some special gadgets called *fans* that are described by Kant in [13], such that G has a planar straight line drawing with angular resolution at least 5° if and only if A is planar. Details are provided in the full paper.

Theorem 4. *Given a triconnected planar graph and an angle α, determining whether it admits a planar straight line drawing with angular resolution at least α is NP-hard.*

4 Drawing a Series-Parallel Angle Graph

A *series-parallel* directed graph G with a source and a sink [1] is defined recursively as follows: A series-parallel directed graph is either a single directed edge or a *series composition* or a *parallel composition* of two series-parallel directed graphs G_1 and G_2. G_1 and G_2 are called the *series-parallel components* of G.

We show that a series-parallel can be tested for consistency in linear time. Our approach is bottom up: We assume a cartesian coordinate system in which left, right, above and below have their usual meanings. Hence assuming that the source of the series-parallel graph is placed at the origin, we compute the range over all the possible drawings of the graph, of the angles made with the x-axis by an imaginary line-segment joining its source and sink. This information is encoded in form of two tuples (α, β) and (γ, δ). α and β (γ and δ) are the lower and upper bounds on the angles made by this imaginary line-segment in all the drawings in which the sink is placed to the left (right) of the source. These tuples are computed for a series-parallel graph G in constant time from the tuples of its series-parallel components, giving a linear time testing algorithm. This approach can be easily modified to give a linear time algorithm for drawing a series-parallel angle graph. The details of this approach are provided in the full paper.

Theorem 5. *Given a series-parallel angle graph with n vertices, we can test if it admits a drawing or not and construct a drawing if it does in $O(n)$ time.*

5 Area Requirement of Angle Graphs

We now investigate the area requirement of angle graphs. Our main theorem is:

Theorem 6. *For every $m \geq 1$, there exists an angle graph A_m with $3m$ vertices that requires area $\Omega(4^m)$ for any drawing.*

Sketch of Proof: Angle graph A_m can be described recursively: Angle graph A_1 is shown in Fig. 8(a). Angle graph A_m is constructed from A_{m-1} (see Fig 8(b)) by introducing vertices u_m, v_m and w_m and edges (u_{m-1}, u_m), (v_{m-1}, u_m), (v_{m-1}, v_m), (w_{m-1}, v_m), (w_{m-1}, w_m) and (u_{m-1}, w_m) in the exterior of the angle graph A_{m-1}.

Using plane geometry we show that $Area(A_m) = 4\,Area(A_{m-1})$ and consequently $Area(A_m) = \Omega(4^m)$. □

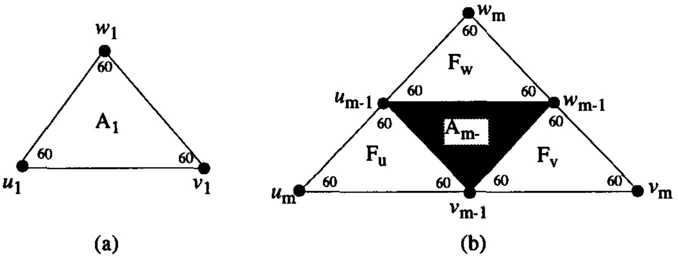


Figure 8: Angle graph A_m: (*a*) A_1; (*b*) Constructing angle graph A_m from A_{m-1}.

6 Multilayered Angle Graphs

Recall the definition of *multilayered* angle graphs and *multiplanar* angle graphs from Sec. 1.1. We now show that even if we consider only rectilinear graphs and restrict the number of layers to two, the problem of testing them for biplanarity is NP-hard.

We reduce the 3SAT problem with restriction that a variable can not occur both negated and un-negated in the same clause, to an instance of this problem. Suppose we are given a set $X = \{x_1, x_2, \ldots, x_n\}$ of variables and a set $C = \{c_1, c_2, \ldots, c_m\}$ of clauses. We construct a bilayered rectilinear angle graph A with two layers called the *red* and *blue* layers such that there is a satisfiability assignment of C if and only if A is biplanar. We call the edges of A belonging to the red (blue) layer as *red* (*blue*) edges. We use the following gadgets (which are bilayered angle graphs) for constructing A:

- **Subcourse:** A subcourse S is shown in Fig 9(a). Vertex u_j, where $1 \le j \le 2m$ is called the j^{th} *input* vertex of S. Vertex v_j, where $1 \le j \le 2m$ is called the j^{th} *output* vertex of S. Edge (t_j, t_{j+1}), where $1 \le j \le m$ is called the j^{th} *reference edge* of S. Edge (s_j, s_{j+1}), where $1 \le j \le m$ is called the j^{th} *value edge* of S. Each reference as well as value edge of S can either be blue or red (as will be described later). Similarly edges (d_1, d_2) and (d_3, d_4), called the *top clamping* and *bottom clamping* edges of S respectively, can be either blue or red. Edge (a_{2j-1}, a_{2j}) for $1 \le j \le m$ is called the j^{th} *inner* edge of S and is blue. Edge (v_{2j-1}, v_{2j}) for $1 \le j \le m$ is called the j^{th} *outer* edge of S and is red. Each edge (u_{2j}, a_{2j}) as well as each edge (a_{2j}, v_{2j}), where $1 \le j \le m$ is red. Other edges of the subcourse are blue.
  Let Δ be a drawing of S. The vertical line segment (s_1, s_{m+1}) is called the *value line segment* of S in Δ. The vertical line segment (t_1, t_{m+1}) in Δ is called the *reference line segment* of S in Δ. The value line segment is either to the right of the reference line segment (Fig. 9(b)) or it is to the left of the reference line segment(see Fig. 9(c)) or both overlap. If the value line segment is to the left (right) then S is said to be *left heavy* (*right heavy*) in Δ. If value and reference line segments overlap then S is *balanced* in Δ. As we will see later, each subcourse corresponds to a variable and the left heavy (right heavy) "state" corresponds to the variable being `true` (`false`).
- **Starting and Ending Location:** Fig. 10(a) shows a *starting* location L and Fig. 10(b) shows an *ending* location L'. Vertex c_j (c'_j), where $1 \le j \le m$ is called the j^{th} *starting* (*ending*) vertex of L (L'). Vertex w_j, where $1 \le$

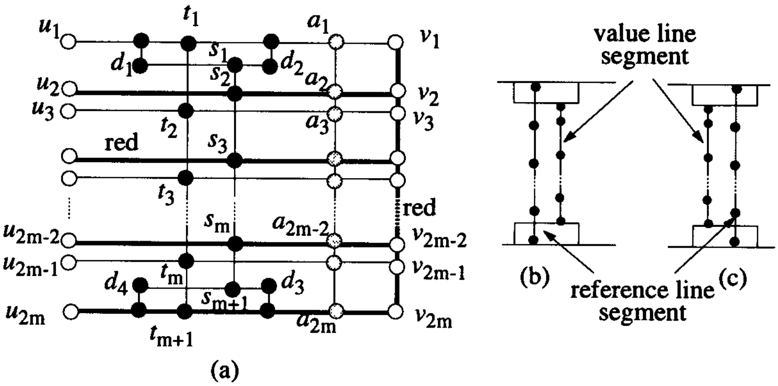


Figure 9: (*a*) A subcourse S; (*b*) A left heavy subcourse; (*c*) A right heavy subcourse.

$j \leq 2m$ is called the j^{th} *separation* vertex of L (L'). Vertices $c_1, c_2, \ldots, c_m$ ($c'_1, c'_2, \ldots, c'_m$) are called the *starting* (*ending*) vertices of the starting (ending) location. All the edges of both starting and ending locations are blue.

- **Hurdle:** A hurdle H is shown in Fig. 10(*c*). Vertex g_j where $1 \leq j \leq 2m$, is the j^{th} *input* vertex of H. Vertex h_j, where $1 \leq j \leq 2m$ is the j^{th} *output* vertex of H. Each edge (h_j, h_{j+1}) is red. Other edges of the hurdle are blue.
- **Runner:** A runner R(Fig. 10(*d*)) is a connected sequence of 14 alternating red and blue edges. The angle between consecutive edges is 180°. Vertex a is the *first end point* of R and vertex a' is its *last end point.* The edge incident on a and a' are called the *first* and *last* edges of R respectively. The first edge of R is blue. As we will see later each runner corresponds to a clause.

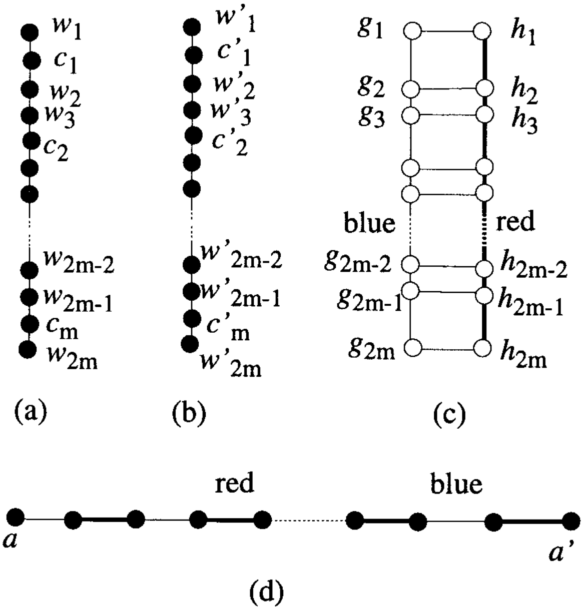


Figure 10: (*a*) A Starting Location; (*b*) An Ending Location; (*c*) A Hurdle; (*d*) A Runner.

6.1 Completed Course

Let S_i be a subcourse whose edges are assigned to the layers as follows: If variable x_i does not occur in clause c_j then both the j^{th} reference and value edges are red and both the clamping edges of S_i are blue. If variable x_i occurs negated in c_j then the j^{th} reference edge is red, j^{th} value edge is blue and both the clamping edges are blue. If variable x_i occurs un-negated in c_j then the j^{th} reference edge is blue, j^{th} value edge is red and both the clamping edges are red.

A *course* K_i is constructed recursively as follows:

1. K_1 is the subcourse S_1. S_1 is called the *first* subcourse of K_1.
2. K_i is constructed from K_{i-1} by identifying the k^{th} output vertex of the $i-1^{th}$ subcourse S_{i-1} of K_{i-1} with the k^{th} input vertex of subcourse S_i. S_i is called the i^{th} subcourse of K_i. For every j such that $1 \leq j < i$, the j^{th} subcourse of K_i is the j^{th} subcourse of K_{i-1}

A *completed course* G is constructed from a course K_n, a hurdle H, a starting location L and an ending location L' as follows: Join the j^{th} separation vertex of L with the j^{th} input vertex of the first subcourse of K_n by a blue edge. Join the j^{th} output vertex of the n^{th} subcourse of K_n with the j^{th} input vertex of H by a blue edge. Join the j^{th} output vertex of H with the j^{th} separation vertex of L' by a blue edge. Fig. 11(a) shows a high level view of G.

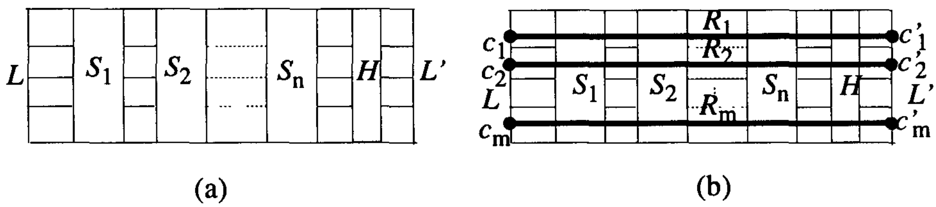


Figure 11: A high level view of (a) Completed Course G; (b) angle graph A.

6.2 Angle graph A

Angle graph A has a completed course G as its (angle) subgraph. Let K, H, L and L' be the course, hurdle, starting location and ending location respectively in G. In addition to G, for each clause c_j, graph A has a runner R_j with the j^{th} starting and ending vertices of L and L' respectively as its first and last end points. R_j is called the *representative* of clause c_j in A. This completes the construction of A. Fig. 11(b) shows a high level view of A.

Following correspondence exists between a biplanar drawing Δ of A and a satisfiability assignment ψ of C: A variable x_i is **true** in ψ if and only if the subcourse S_i is left heavy.

Theorem 7. *C has a satisfiability assignment if and only if the bilayered angle graph A is biplanar.*

Theorem 8. *The problem of testing whether a bilayered angle graph is biplanar is NP-hard.*

Corollary 9. *The problem of testing whether a multilayered angle graph is multiplanar is NP-hard.*

References

1. R.F. Cohen, G. Di Battista, R. Tamassia, I.G. Tollis, and P. Bertolazzi, "A Framework for Dynamic Graph Drawing," *Proc. ACM Symp. on Computational Geometry*, 1992, pp. 261–270.
2. I.F. Cruz, A. Garg, "Drawing Graphs by Example Efficiently: Trees and Planar Acyclic Digraphs," *Graph Drawing '94 (Proc. DIMACS Workshop on Graph Drawing)*, to appear.
3. G. Di Battista, P. Eades, R. Tamassia and I.G. Tollis, "Algorithms for Drawing Graphs: An Annotated Bibliography," *Comp. Geom. Theory Appl.*, to appear
4. G. Di Battista, R. Tamassia, and I.G. Tollis, "Area Requirement and Symmetry Display of Planar Upward Drawings," *Disc. & Comp. Geom.*, vol. 7, 1992.
5. G. Di Battista and L. Vismara, "Angles of Planar Triangular Graphs," *Proc. ACM Symp. on Theory of Computing*, 1993.
6. P. Eades and T. Lin. "Algorithmic and declarative approaches to aesthetic layout," *Graph Drawing '93 (Proc. ALCOM Workshop on Graph Drawing).*
7. M. Formann, T. Hagerup, J. Haralambides, M. Kaufmann, F.T. Leighton, A. Simvonis, E. Welzl, and G. Woeginger, "Drawing Graphs in the Plane with High Resolution," *Proc. FOCS*, 1990, pp. 86–95.
8. A. Garg, M.T. Goodrich, and R. Tamassia, "Area-efficient Upward Tree Drawings", *Proc. ACM Symp. on Computational Geometry*, 1993, pp. 359–368.
9. A. Garg and R. Tamassia, "Planar Drawings and Angular Resolution: Algorithms and Bounds", *European Symposium on Algorithms, 1994.*
10. M.R. Gary and D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," W.H. Freeman, New York, NY 1979.
11. M. Himsolt, "A View to Graph Drawing Algorithms Through GraphEd", *Graph Drawing '93 (Proc. ALCOM Workshop on Graph Drawing)*, Paris, 1993.
12. T. Kamada. *On Visualization of Abstract Objects and Relations.* PhD thesis, Department of Information Science, University of Tokyo, 1988.
13. G. Kant. Drawing planar graphs using the *lmc*-ordering. In *Proc. 33th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 101–110, 1992.
14. C.E. Leiserson, "Area-Efficient Graph Layouts (for VLSI)," *Proc. IEEE Symp. on Foundations of Computer Science*, 1980, pp. 270–281.
15. S.M. Malitz and A. Papakostas, "On the Angular Resolution of Planar Graphs," *Proc. ACM Symp. on Theory of Computing*, 1992, pp. 527–538.
16. R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.
17. R. Tamassia and I. G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete Comput. Geom.*, 1(4):321–341, 1986.
18. L. Valiant, "Universality Considerations in VLSI Circuits," *IEEE Trans. on Computers*, vol. C-30, 1981, pp. 135-140.
19. G. Vijayan, "Geometry of Planar Graphs with Angles", *Proc. ACM Symp. on Computational Geometry*, 1986, pp. 116–124.
20. G. Vijayan and A. Wigderson, "Rectiliear Graphs and their Embeddings," *SIAM Journal of Computing*, Vol. 14, No. 2, May 1985.