Planarity of Edge Ordered Graphs

Gopalakrishnan Vijayan & Avi Wigderson

Department of Electrical Engineering and Computer Science
Princeton University
Princeton, New Jersey 08544

Technical Report #307
December 1982

ABSTRACT

An edge ordered graph is an undirected graph together with cyclic orderings of the edges at each vertex. An edge ordered graph is said to be planar, if there is a planar embedding of the graph in which the cyclic orderings of the edges at the vertices are preserved. These graphs have applications in VLSI layout problems. In this paper, we describe a linear time algorithm for recognizing planar edge ordered graphs.

Planarity of Edge Ordered Graphs

Gopalakrishnan Vijayan
Avi Wigderson

Department of Electrical Engineering and Computer Science
Princeton University
Princeton, New Jersey 08544

1. Introduction

Suppose we are given a graph G and in addition we are given a cyclic ordering of the edges at each vertex. We call such graphs edge ordered graphs, or in short ordered graphs. We are interested in the following question: Does G have a planar embedding in which at each vertex the edges appear counterclockwise in the given cyclic order? If an ordered graph has such a planar embedding we call it embeddable, otherwise it is non-embeddable.

An efficient algorithm to answer the above question has important applications in VLSI. Consider each vertex in the graph to be a cell in a VLSI layout and the edges coming out in order to be the wires connecting this cell to the outside world. The ordering of the wires at each cell is a property of the cell and cannot be changed. We are now interested in finding a placement of the cells and the wires so that there are no crossings between the wires. Of course, in a real application we have to deal with straight line horizontal or vertical wires. This problem will be dealt with in a different paper [5].

In this paper we present a linear time algorithm for recognizing edge ordered graphs. We first deal with biconnected ordered graphs and then proceed to discuss articulation vertices in ordered graphs. For a definition of biconnected graphs and articulation vertices the reader is referred to [1]. As we shall see later, unlike in ordinary planar embedding [3], joining together the embeddings of the biconnected components at the articulation vertices may not always be possible. Unless otherwise mentioned all graphs in this paper are simple graphs.

2. Some Comments and Definitions

We first mention some obvious but important properties of ordered graphs. Clearly no ordered graph whose underlying graph (obtained by ignoring the cyclic orderings) is nonplanar can be embeddable. But there are ordered graphs which are not embeddable even though the underlying graph is planar. In figure 2.1 we have an example of one such graph. If an ordered graph is embeddable then it has a unique embedding in the sense that the faces of the planar embedding are completely determined by the cyclic orderings. We now need some definitions including that of an edge ordered graph.

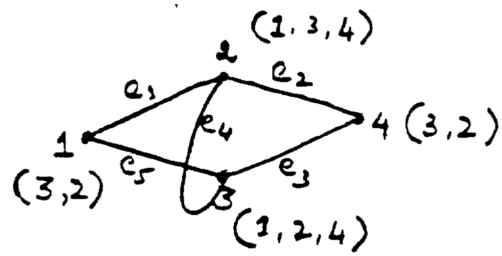

Figure 2.1
A non-embeddable edge ordered graph

Definition 2.1: An *edge ordered graph* G is given by $G = (V,L)$, where V is the set of vertices, and $L = \{L_G(v) \mid v \in V\}$, where $L_G(v)$ is a cyclic list of vertices adjacent to v , satisfying $v \in L_G(u) \Leftrightarrow u \in L_G(v)$. The edge set is given by $E = \{(u,v) \mid u \in L_G(v); u,v \in V\}$.

For convenience in a later definition we assume that the vertices are denoted by integers, i.e. V is a set of integers. Subgraphs of G are defined in the usual fashion but they now inherit the edge orderings at each vertex from G.

Definition 2.2: An ordered graph $G = (V,L)$ is said to be ***embeddable*** if there exists a planar embedding of G in which at each vertex v the edges appear counterclockwise in the order defined by the cyclic list $L_G(v)$.

3. Biconnected Ordered Graphs

In this section we discuss the algorithm for biconnected ordered graphs. We first define what we call *candidate faces* for a biconnected ordered graph.

Definition 3.1: Let $G = (V,L)$ be a biconnected ordered graph. With each edge $e = (v_1,v_2)$, $v_1 > v_2$ we associate two lists of vertices called ***candidate faces*** $CF_1(e)$ and $CF_2(e)$ which are defined as follows. $CF_1(e) = v_1, v_2, \cdots, v_k, v_{k+1}$ where $v_i \neq v_j$ for $1 \leq i < j \leq k$, and $v_{k+1} = v_i$ for some i, $1 \leq i < k-1$. Also, for each l, $1 < l < k+1$, v_{l+1} is the successor of v_{l-1} in the cyclic list $L_G(v_l)$. $CF_2(e)$ is similarly defined but starting with v_2, v_1.

It is easy to see that CF_1 and CF_2 are uniquely defined. For the graph in figure 2.1, $CF_1(e_2) = 4,2,1,3,2$ and $CF_2(e_2) = 2,4,3,1,2$.

We now need a lemma about biconnected undirected graphs. Let us define a biconnected graph to be *minimal* if for every edge e in the graph $G-e$ is not biconnected. The following lemma is taken from [2] and is stated without proof.

Lemma 3.1: If G is a *minimal* biconnected graph having at least four vertices then G contains a vertex of degree two.

Lemma 3.2: In any biconnected graph G which is not a simple cycle, there is a simple path $P = (v_1,v_2),(v_2,v_3), \cdots ,(v_{r-1},v_r)$, $r \geq 2$, with the intermediate vertices (if any) v_i, $i \neq 1,r$ all having degree 2, such that the graph $G' = G-P$ is biconnected.

Proof: Transform the given graph G to another graph G'' by replacing all paths of the form $P = (v_1,v_2),(v_2,v_3), \cdots ,(v_{r-1},v_r)$ where the vertices v_i, $i \neq 1,r$ all have degree 2, by the edge (v_1,v_r). So for each edge e in G'' we have a corresponding path P_e in G. Note that the degree of any vertex in G'' is at least three. If G'' has multiple edges between some two vertices, say u and w, then in G there must be at least two parallel paths between u and w. Since G is not a simple cycle any one of those paths will serve our purpose. If G'' does not have multiple edges then it must have at least 4 vertices. By lemma 3.1 G'' cannot be minimal. Therefore there is an edge e in G'' such that $G''-e$ is biconnected, which implies that $G-P_e$ is also biconnected. ▪

The following theorem gives a necessary and sufficient condition for a biconnected ordered graph to be embeddable.

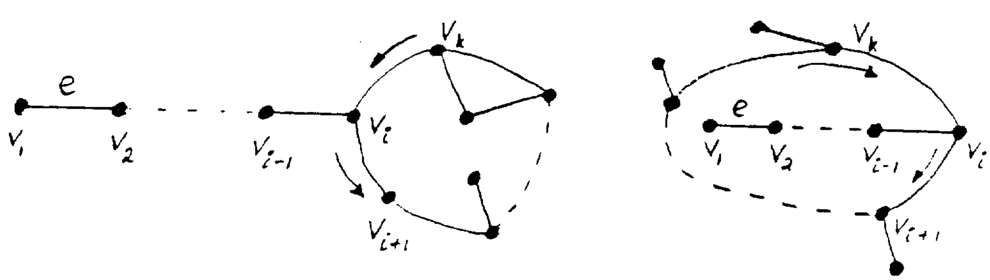

Figure 3.1
Two possible embeddings of $CF_1(e)$

Theorem 1: Let $G = (V,L)$ be an biconnected ordered graph with at least three edges. Then G is embeddable if and only if for each edge e in the graph both the candidate faces $CF_1(e)$ and $CF_2(e)$ represent simple cycles in the graph (i.e. the starting and ending vertices are identical). Moreover, if the graph is embeddable each such distinct candidate face corresponds to a face in the planar embedding.

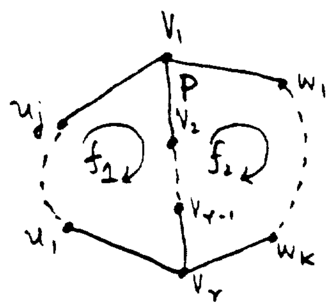

Figure 3.2
The two cycles f_1, f_2 and the path P

Proof:

only if: Suppose for some edge $e = (v_1, v_2)$, $CF_1(e)$ is not a cycle, i.e $CF_1(e) = v_1, v_2, \cdots, v_k, v_{k+1}$ with $v_i = v_{k+1}$ for some i, $1 < i < k-1$. Suppose G is embeddable. Look at the cycle $v_i, \cdots, v_k, v_{k+1}$ in the embedding. Suppose that the edge (v_{i-1}, v_i) is inside this cycle. There can be no other edges (u, v_j), $i < j \leq k$ *inside* this cycle, otherwise u would have appeared instead of v_{j+1} in $CF_1(e)$. From this observation and the fact that the embedding is planar, it follows that v_i is an articulation vertex, which contradicts the biconnectedness of G. The case where (v_{i-1}, v_i) is *outside* the cycle is similar (both cases are depicted in figure 3.1). Similar argument holds if $CF_2(e)$ is not a cycle.

if: The proof of this part is by induction on the number of edges. The basis for the induction are simple cycles, for which the claim is certainly true. Assume that the claim is true for any biconnected ordered graph which has less than k edges. Let G be a biconnected ordered graph which is not a simple cycle and which has k edges. By lemma 3.2, there is a simple path $P = (v_1, v_2), (v_2, v_3), \cdots, (v_{r-1}, v_r)$ with the vertices v_i, $i \neq 1, r$ all having degree 2, such that the graph $G' = G - P$ is biconnected. v_1 and v_r will have degree greater than two. Also assume that $v_1 > v_2$ and $e_{12} = (v_1, v_2)$.

Since all our candidate faces are cycles, if an edge e lies on a candidate face f then either $CF_1(e) = f$ or $CF_2(e) = f$. So each edge will be present in exactly two of these candidate faces. Hence the path P will appear in $CF_1(e_{12})$ and its reverse path will appear in $CF_2(e_{12})$. Let
$f_1 = CF_1(e_{12}) = v_1, v_2, \cdots, v_r, u_1, \cdots, u_j, v_1$,
$f_2 = CF_2(e_{12}) = v_r, v_{r-1}, \cdots, v_1, w_1, \cdots, w_k, v_r$, and
$f_3 = v_1, w_1, \cdots, w_k, v_r, u_1, \cdots, u_j, v_1$.
It follows from the definition of the candidate faces f_1 and f_2 that the vertices u_j, v_2, w_1 appear consecutively in that order in $L_G(v_1)$ and that w_k, v_{r-1}, u_1 appear similarly in $L_G(v_r)$ (see figure 3.2). Therefore for each edge in f_1 or f_2 which is not in P, the new candidate face in G' will be f_3 which is a simple cycle. Thus the candidate faces for G' are the same as

those for G, excepting for f_3 replacing the two faces f_1 and f_2. So for each edge in G' its two candidate faces are again simple cycles. By induction hypothesis G' is embeddable and each distinct candidate face corresponds to a face in its embedding. The orderings of the edges at the vertices v_1 and v_r imply that the end edges (v_1,v_2) and (v_{r-1},v_r) of the path P are both trying to go inside the face corresponding to f_3. It is now a simple matter to insert the path P into that face. Also this action will cut this face into two new faces of G corresponding to the candidate faces f_1 and f_2. This completes the proof of the theorem. ▪

The above theorem leads to the following algorithm for recognizing embeddable biconnected ordered graphs. The algorithm also outputs the faces of the embedding if the graph happens to be embeddable.

```
Algorithm embed-biconnected(G);
begin
    if G is an edge then return;
    if |E| > 3|V|-6 then quit( 'not embeddable');
    for each edge e do
    begin
         mark[e,1]:= false;
         mark[e,2]:= false
    end;
    for each edge e do
         for i:= 1 to 2 do
         begin
              if not mark[e,i] then
               begin
                    f := candidate-face( e, i );
                    if f ≠ cycle then quit( 'not embeddable' );
                    for each edge e' = (v1,v2) in f do
                         if v1 > v2 then mark[e',1]:= true
                          else mark[e',2]:= true;
                    output (f)
              end
         end
end
```

Function call *candidate-face(e,i)* returns the candidate face $CF_i(e)$ and the function can be implemented exactly as described in definition 3.1. In the calls to this function, each edge e can be traversed at most twice, due to the flags mark[e,1] and mark[e,2]. Therefore the algorithm runs in time $O(|V|)$.

4. Articulation Vertices

Now we turn our attention to articulation vertices in ordered graphs. In the following definition we isolate the articulation vertices that create problems.

Definition 4.1: Let G be an ordered graph . An articulation vertex u in G is said to be ***illegal*** if there are four vertices v_1,v_2,v_3,v_4 appearing in that order (not necessarily consecutively) in the cyclic list $L_G(u)$, such that v_2,v_4 are in one biconnected component and v_1,v_3 are in a different component. An articulation vertex which is not ***illegal*** is said to be ***legal***.

The following two lemmas reveal why legal articulation vertices behave nicely while illegal ones don't.

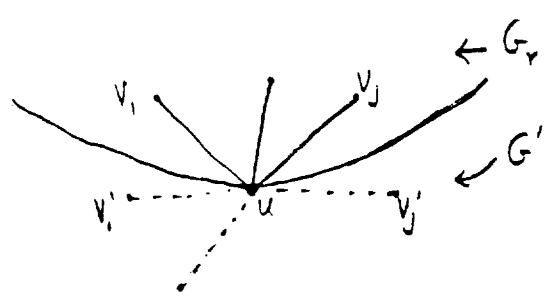

Figure 4.1
A legal articulation vertex u

Lemma 4.1: Let u be an legal articulation vertex in an ordered graph G. Let its removal result in connected subgraphs $G_1', \cdots, G_r'$. Let G_i be the result of adding to G_i' the vertex u and the edges to u from vertices in G_i'. Then G is embeddable if and only if each G_i is embeddable.
Proof: If any G_i is not embeddable then clearly G cannot be embeddable. Let each of the G_i be embeddable. We now use induction on k the number of subgraphs meeting at u. The claim is certainly true for $k = 1$. Let the claim be true for any $k < r$. Since u is legal we have for some $j \geq 1$, vertices $v_1, \cdots, v_j$ appearing consecutively in the cyclic list $L_G(u)$, such that these are the only vertices adjacent to u belonging to some subgraph, say the last one G_r. By induction hypothesis the graph G' obtained from G by removing $G_r - u$ is embeddable. Let v_1', v_j' be the predecessor of v_1 and the successor of v_j respectively in $L_G(u)$. It is possible that $v_1' = v_j'$. Let

us draw the embedding of G' such that edges (u,v_1') and (u,v_j') appear on the outside face, i.e. on the exterior region. This can always be done without destroying the ordering of edges at any vertex [2, page 66]. Similarly make (u,v_1) and (u,v_j) appear on the outside face in the embedding of G_r. All we need to do now is to join the two embeddings at u, as shown in figure 4.1, to get an embedding for G. ▪

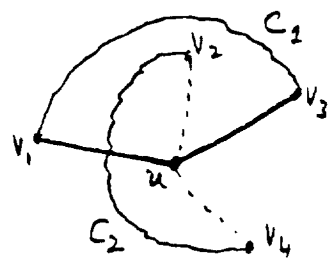

Figure 4.2
An illegal articulation vertex u

Lemma 4.2: If an ordered graph G contains an illegal articulation vertex then G is not embeddable.

Proof: Let u be an illegal articulation vertex in G. We have four vertices v_1,v_2,v_3,v_4 in that order in the cyclic list $L_G(u)$ such that v_1,v_3 are in one biconnected component G_1 and v_2,v_4 are in another G_2 (figure 4.2). Any two edges in a biconnected graph must lie on a simple cycle[1]. Hence the edges (u,v_1) & (u,v_3) lie on a cycle C_1 in G_1 and the edges (u,v_2) & (u,v_4) lie on a cycle C_2 in G_2. It is impossible to draw G on the plane with the these four edges appearing in the prescribed order without C_1 and C_2 crossing each other. Hence G is not embeddable. ▪

We are now in a position to state a necessary and sufficient condition for an ordered graph to be embeddable.

Theorem 4.1: An edge ordered graph G is embeddable if and only if every biconnected component of G is embeddable and every articulation vertex is legal.

Proof: If G is embeddable then any biconnected component of G is also embeddable. Also by lemma 4.2 all its articulation vertices must be legal. This proves the necessary part. We use lemma 4.1 and induction on the number of articulation vertices to prove the sufficient part. ▪

The above theorem gives us the following $O(|V|)$ running time algorithm for recognizing embeddable ordered graphs.

Algorithm *embed-ordered-graph(G)*;
begin
  if $|E| > 3|V|-6$ **then** *quit ('not embeddable')*
  find-biconnected-components(G);
  for each articulation vertex u do
    if *check-illegal(u)* **then** *quit ('not embeddable')*
  for each biconnected component G_i **do** *embed-biconnected*(G_i);
end

Function *check-illegal(u)* can be implemented to run in $O(d)$ time where d is the degree of the vertex u. One way to do this is by reducing $L_G(u)$ successively using certain rules. We leave this as an exercise to the reader. Each edge in G will be involved in at most two calls to this function and hence the time spent in calls to this function is $O(|E|)$. Finding articulation vertices and biconnected components can be done again in $O(|E|)$ time [1]. Each edge will be present in exactly one biconnected component an hence the total time spent in calls to *embed-biconnected* is again $O(|E|)$. Hence the algorithm has running time $O(|E|) = O(|V|)$. Note that in order to get the faces of the embedding the two algorithms need to be altered. This can be done so that faces are also obtained in the same running time $O(|V|)$. But we will spare the reader the details of this task.

5. Conclusions

The problem of embedding edge ordered graphs arose during the implementation of ALI [4], a procedural VLSI design system currently under implementation at Princeton. The algorithm described in this paper (with modifications given in [5]) will be implemented in ALI.

6. Acknowledgements

This problem was originally raised by Professors Bob Sedgewick and Dick Lipton. We wish to thank Professors Dick Lipton and Jacobo Valdes for several useful discussions. The first author's research was supported by DARPA under grant# N0014-82-K-0549. The second author's research was supported by an IBM fellowship.

7. References

[1] Aho, A. V., Hopcroft, J. E., Ullman, J. D., *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[2] Behzad, M., Chartrand, G., Lesniak-Foster, L., *Graphs & Digraphs*, Wadsworth International Group, 1981.

[3] Hopcroft, J. E., Tarjan, R. E., "Efficient Planarity Testing", *J. Assoc. Comput. Mach.*, 21, (549-568), 1974.

[4] Lipton, R. J., North, S. C., Sedgewick, R., Valdes, J., Vijayan, G., "ALI: a Procedural Language to Describe VLSI Layouts", *Proc. of the 19th Design Automation Conference*, Las Vegas, June 1982.

[5] Vijayan, G., Wigderson, A., "Rectilinear Graphs and their Embeddings", (in preparation).