September 1981

Report. No. STAN-CS-81-876

On Linear Area Embedding of Planar Graphs

by

Danny Dolev
Howard Trickey

Department of Computer Science

Stanford University
Stanford, CA 94305

On Linear Area Embedding of Planar Graphs

DANNY DOLEV[†]
IBM Research
San Jose, Calif.

HOWARD TRICKEY[‡]
Stanford University
Stanford, Calif.

Abstract

Planar embedding with minimal area of graphs on an integer grid is one of the major issues in VLSI. Valiant [V] gave an algorithm to construct a planar embedding for trees in linear area; he also proved that there are planar graphs that require quadratic area.

We give an algorithm to embed outer-planar graphs in linear area. We extend this algorithm to work for for every planar graph that has the following property: for every vertex there exists a path of length less than K to the exterior face, where K is a constant.

Finally, finding a minimal embedding area is shown to be $\mathcal{NP}$-complete for forests, and hence for more general types of graphs.

[†] Part of the work was done while the author was at Stanford University and supported by a Chaim Weizmann postdoctoral fellowship and DARPA contract MDA903-80-C-0102.

[‡] Work supported by an NSERC scholarship and NSF grant MCS-80-12907.

§1 Introduction

VLSI design motivates the following class of problems: given a graph, map its vertices onto a plane and its edges onto paths in that plane between the corresponding mapped vertices. Normally there are some restrictions that the mappings must obey, such as a minimum distance between mapped vertices. The maps give a layout, and the problem is to find a layout with a small cost. The mapping restrictions and the cost function together specify a particular member of the class of layout problems.

Embedding of graphs has been extensively studied during the last few years [L, V, FP, BK, CS, R]. In this paper we consider the layout problem when the layouts arc rectilinear embeddings *without* crossover3 and the cost is the area of a box bounding the layout.

Definition: Lct the plane have a Cartesian coordinate system. Call those points with integer coordinates grid points, and the horizontal and vertical line segments connecting those points grid edges. A graph layout is called a *rectilinear embcdding* if thc vertices are mapped to grid points and the edges are mapped onto a path constrained to follow grid edges. It is called a *rectilinear* embedding without crossovers if mapped edges touch only at mapped vertices upon which both edges arc incident. A layout without crossovers is a planar *em* bedding.

Note that a graph has a rectilinear embedding only if the degree of all its vertices is 4 or less. Higher degree vertices can be simulated by a group of $\lceil \log d \rceil$ connected vertices, each of degree at most 4. Because the number of edges in a planar graph is bounded by a linear function of the number of vertices, the simulation will not increase the number of vertices by more than a constant factor. Therefore, we restrict ourselves to graphs of degree at most 4.

Definition: The bounding box area of a rectilinear embedding is

$$(x_{max} - x_{min} + 1) \times (y_{max} - y_{min} + 1),$$

where x_{max} is the maximum x-coordinate of a mapped vertex and y_{max}, x_{min}, and y_{min} are defined similarly.

In this paper "area" will mean "bounding box area" unless otherwise specified. The various notions of graph theory used are defincd in the Appendix.

In [V], Valiant lookcd at the layout. problem for rectilincar cmbcddings (both with and without crossovers), using thc bounding box area cost. He proved that a trcc of vertices with maximum dcgrcc 4 can be laid out without crossovers in an area that is linear in the number of edges (or vertices). He also proved that there are planar graphs requiring quadratic area for such cmbeddings.

Bctwccn trccs and planar graphs in complexity lie the outcrplanar graphs (those whose vcrtices can all bc put on the cxtcrior face in a planar drawing). Lciscrson has shown that outcrplanar graphs can be laid out in lincar arca if crossovcrs are allowed [L]. Wc shall now show that the same is true cvcn without crossovers. Hong and Rosenbcrg [HR] studied graphs that are "almost binary trees". The constructions they used in thcir paper lead to a different planar crnbcdding of outerplanar graphs, and the gcneralization to other types of graphs does not preserve the planarity of the embedding. With our construction, the following generalization is possible: a graph can bc embedded without crossovers in a linear arca if it can be reduced to an outerplanar graph in a number of steps which does not depend on the size of the graph. A reduction step consists of (a) triangulating the graph by adding edges to its interior, and (b) taking the dual of the resulting graph (putting the new vertices inside the old faces). This will be made more precise later on.

The embcddings we obtain, and those obtained in the past, require an area that is a function of the number of vertices. This suggests the question of whether an optimal embedding is achievable. We shall prove that finding an optimal embedding for a forest is $\mathcal{N}$ P-complete.

In Section 2 we describe the embedding of outerplanar graphs. In Section 3 we generalize the family of graphs that are cmbeddable in linear area. The embedding of this family is obtained by recursive use of thc embedding of outerplanar graphs. In Section 4 we provc that embedding in a minimal area is J/P-complete.

§2 Layout of Outerplanar Graphs

Only connected graphs need bc considered in layout problems: if there are a number of separate componcnts, they can bc joined together with dummy edges which can be removed from the final layout. There may bc better layouts for an unconnected graph, but this method is good enough to get a linear area bound, since the graph remains planar and the numbcr of vertices stays the same. Therefore, we consider only connected graphs in this paper. Testing whether a graph is planar or not can be done in linear time [I-IT], as can testing a graph for outerplanarity [I3,M]. Finding the planar and outcrplanar embeddings of thc corresponding graphs can also bc found in linear time [W,B]. Because we are considering only planar graphs, we will assume that the graph is always given with its planar embedding.

A triangulation of a planar graph is a maximal planar graph obtained by the addition of a maximal number of edges that do not cross. A *triangulation of* an *outcrplanar* graph is a graph obtained from an outerplanar one by the addition of a maximal number of edges on the internal side of thc exterior face.

Valiant [V] gives an algorithm to find a lincar area embedding of a tree. Our embedding of outcrplanar graphs is based on this result, with the only exception that we also need to maintain the orientation of the tree. That is, the order of the edges at every branching of the crnbcdded trec must be the samc as in thc original tree, and thus the layout is just a rectilinear deformation of thc givcn trce on the grid.

Theorem 1. *There exists a linear embedding of a tree that is orientation preserving.*

Proof: The proof is just adaptation of Valiant's proof [V80], showing that the embedding is achievable preserving orientation. As in Valiant, we will assumc that all the vertices are of dcgree at most 4.

Valiant's construction is a divide-and-conquer one where two approximately equal-sized parts are rccursively embedded, and thcn joined by lead-out paths from the two

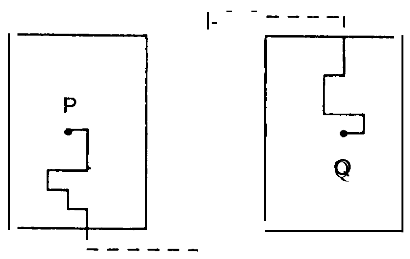

Figure 2.1 Joining subgraphs with lead-out paths

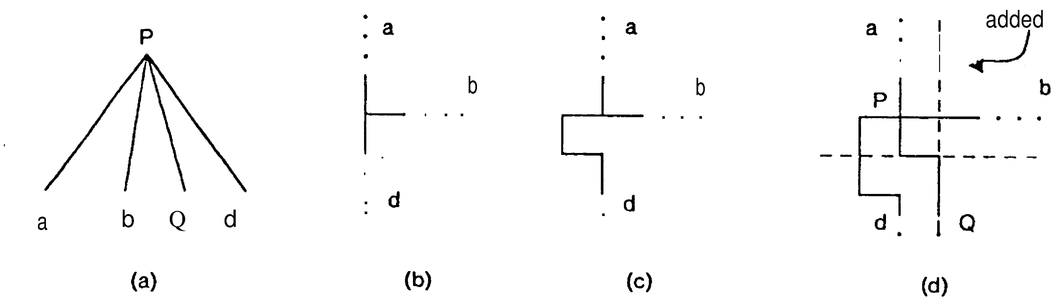

Figure 2.2 Local deformation required for lead-out path

appropriate vertices (see Fig. 2.1). The "leading out" requires that the grid bc augmented with additional horizontal and vertical lines along the lead-out paths (roughly speaking). To preserve the orientation, it is enough to be consistent while sequencing the edges leaving a vertex. Specifically, before looking for a leading out path we may need to make a local change among thc edges already leaving the vertex.

In the casc shown in Fig. 2.2(a), the edge leaving P to Q goes between the edges to b and d. Therefore, thc embedding in Fig. 2.2(b) needs to be deformed to that shown in Fig. 2.2(c), so that there is a place for the edge leading to Q to be connected. The lead-out path starts from this point and goes to the boundary of the ernbcdding, without crossing other edges. To accomplish the deformation at P, it suffices to insert an additional horizontal and vertical grid line, as shown in Fig. 2.2(d). Whenever the lead-out path changes direction, another grid line needs to be inserted to accomodate it.

Valiant's linearity proof is based on a logarithmic bound on the number of direction changes required for any lead-out path. Examination of his proof of that, bound reveals that it still holds cvcn if wc specify which side we lead out on. The base case changes slightly, but it still follows that we can find an orientation preserving and linear area ernbcdding of a tree. ∎

The idea behind the embedding of outcrplanar graphs is that the dual of the triangulated graph is a tree of degree 3. Therefore, we can embcd it in linear arca, and from it obtain the embedding of the original outerplanar graph.

Theorem 2. *Any outerplanar graph with n vertices and degree at most 4 has a planar rectilinear embedding a bounding box area $\leq cn$, for some constant c.*

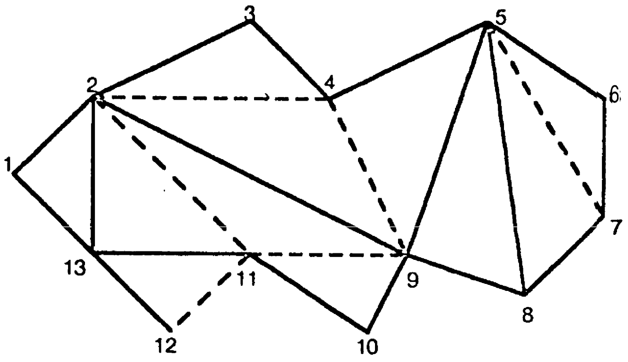

Figure 2.3 Triangulation of G to form TRI(G)

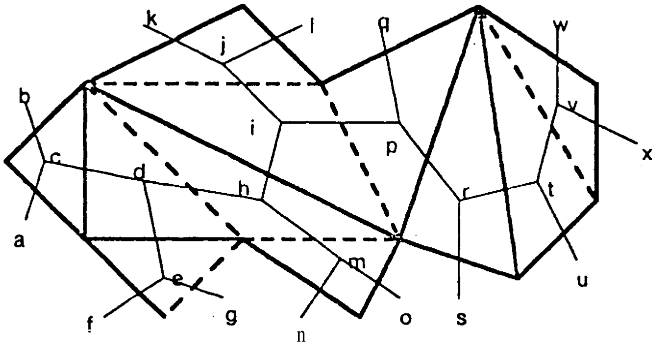

Figure 2.4 Taking the dual of TRI(G) to form T

Proof: Suppose we are given an outerplanar graph G with n vertices of degree at most 4. The outcrplanar graph G can be drawn on a planc with all the vertices on the exterior face, with the edges being some subset oi a triangulation of that face.

Let TRI(G) be the outerplanar graph obtained from G by triangulation. In Fig. 2.3, the dummy edges added to G to construct TRI(G) arc shown as broken lines. The resulting graph may have vertices with degrcc greater than 4, but this will not affect our construction because thc dummy. edges will not appear in the final layout.

Now construct the dual graph T of TRI(G) as follows. Insert a vertcx of T at the ccntcr of cach triangle of TRI(G). Also, add one vcrtcx for cach edge on TRI(G)'s exterior face; locate thcse vcrtices in the exterior facc (just "outsidc" thc corresponding edges). The edges of T join those vertices separated by one edge of TRI(G) (see Fig. 2.4). The outerplanarity of TRI(G) ensures that the dual graph T is, in fact, a tree; if it had a cycle, then TRI(G) would have a vcrtex completely surrounded by a ring of other vertices, violating outerplanarity.

There is a one-one correspondence between the edges of T and the edgcs of TRI(G). Also, TRI(G) has at rnost $2n - 3$ cdgcs (a property of outcrplanar graphs), so thc number of cdgcs in T is boundcd by a lincar function in n.

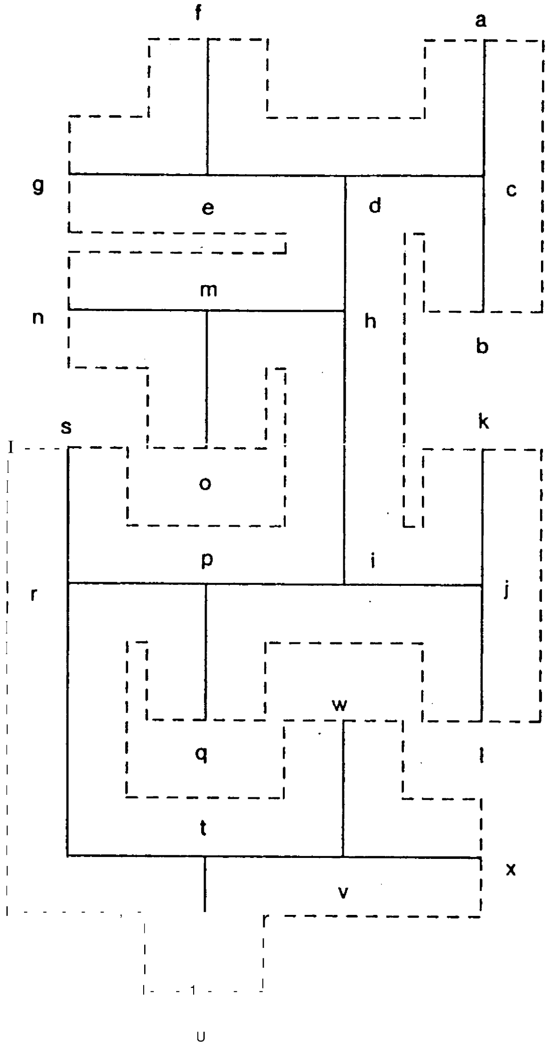

Figure 2.5 T and the regions between adjacent leaves

Except for the leaves, all of T's vertices have degree 3, so the conditions of Valiant's Theorem 5 [V] and our Theorem 1 are satisfied. Hence there is some constant k such that we can always find a rectilinear embedding of T having an area $\leq kn$, with the additional property that the tree's orientation is preserved.

To leave enough space in the embedding to run the edges of the original graph, expand the grid 9 times, horizontally and vertically.

The layout of G can now be done using T as a guide. One can traverse the exterior face of a tree embedded in a plane by following along the edges, always keeping on one side of them. The exterior face of T in Fig. 2.5 is a, c, b, c, $d, h, \ldots, f$, e, d, c, a. Two leaves are *adjacent* if there is no intervening leaf between them in some traversal of the exterior face. In the embedding for T there is a region between two adjacent leaves which is bounded by: (i) the edges followed in the traversal between the leaves; (ii) attached to each leaf, a 4 unit line segment perpendicular to the leaf's edge and on the side closer to the other

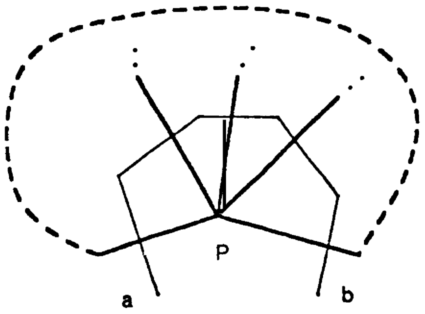

Figure 2.6 A vertex in TRI(G)

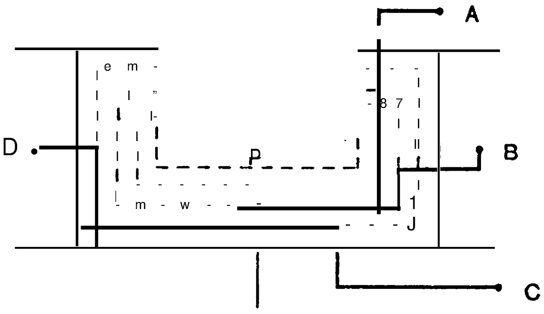

Figure 2.7 Allodating tracks for edge ernbedding

leaf; (iii) line segments parallel to the edges of (i), 4 units away on the side that was being traversed, and just long enough to join together. Fig. 2.5 shows these regions; call the set $\mathcal{R}$.

We now claim that for each vertcx of TRI(G) there is a corresponding region in $\mathcal{R}$, and that no two vertices correspond to the same region in $\mathcal{R}$. Consider Fig. 2.6 where a general vertex P of TRI(G) is shown (TRI(G) drawn with darker lines, T with lighter ones). P must be on the exterior face of TRI(G) because of outerplanarity, and it must have at least two exterior edges incident on it, due to the triangulation. Thus, the method for taking the dual implies that T has two leaves, a and b, corresponding to the exterior edges of TRI(G). Thcre are also edges of T joining any of the other faces around P, so that a and b are adjacent leaves in T. Furthermore, P is the only vertex of TRI(G) between a and b on P's side of T; otherwise there would be more than one edge of TRI(G) crossing an cdgc in the path a.. .b. Notc that all of thc cdgcs from P cross exactly onc of the edges in the path $a\ldots b$, and no more edges thereafter.

Now the embedding of G proceeds by placing each vertex of TRI(G) somewhere in the middle of the corresponding region in $\mathcal{R}$. Then we embed the edges of G, a subset of the edges of TRI(G) such that at most 4 edges are connected to each vertex. Since there are 4 "tracks" (sequences of connected grid line segments) in each region going parallel to the tree cdgcs, there is cnough room for the 4 cdgcs of G to go along until the tree edgc that it must cross is rcachcd. Considcr Fig. 2.7, whcrc cdgcs from P to A, B, C, and D are to bc embcddcd. To thc right of P, the regions that must bc cntcrcd arc cncountcrcd in thc ordcr C, B, A. The edge to C will be embcddcd along thc track closest to the trec edgcs,

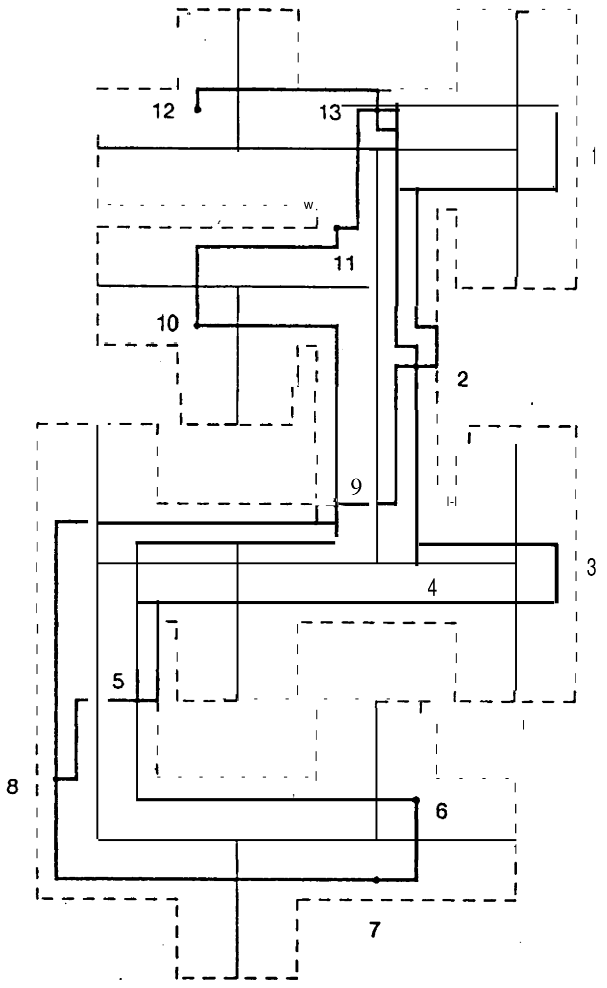

Figure 2.8 Layout of G over *T*

with the edgc to *B* along the next, etc. The same track allocation procedure is repeated to the left of *P*.

The actual crossing into an adjacent region can then be accomplished .anywhere along the boundary between two regions (with the possible exception of places near the corners or near the embedded vertices of G: the expansion ensures that there will still be an allowable crossing place). Thc same argument shows that the adjacent rcgion will also havc a track allocated for the edge being embedded, and that the way to that track will bc clear. Thus, G can- be cmbcddcd over *T* without crossovers.

As a side remark, note that the layout for G has preserved its orientation. In Fig. 2.7,' the regions containing A, B, C, and *D* are cncountcred in clockwise order around *P, so* the track allocation procedure sequences thc edges. leaving *P* in that order. This is the same order as in the original G because *T* was laid out in an orientation-preserving manner.

The solid lines in Fig. 2.8 shows the embcdding for the G in Fig. 2.3.

The embedding found for G has 'a bounding box whose area is $\leq cn$ for somc c, bccausc it is at most $(9a + 8)(9b + 8)$, whcrc a and b arc the dimensions of thc embedding found for T, and $ab \leq kn$. ∎

§3 Layout of Planar Graphs

We shall extend the embedding method for outerplanar graphs to a broad class of planar graphs. Valiant [V] proved that there are planar graphs that require quadratic embedding area. We cannot give a complete characterization of the planar graphs that can be embedded in a linear area, but we will describe a family of graphs that can be so embedded.

The idea behind the construction is to reduce planar graphs to outerplanar graphs using a method similar to that used in the reduction of outerplanar graphs to trees. The triangulation method that we used for outerplanar graphs needs to be extended carefully to planar graphs. The reason is that the dual graph we-obtain from the triangulated one has to be simpler, in some sense, than the original graph. It is possible to triangulate a tree in such a way that the dual will not be a tree; if such a triangulation were used it might not lead to a terminating procedure.

The outerplanar triangulation is defined with respect to a planar embedding of a planar graph. Note that there is a linear algorithm [W] to find the planar embedding of a planar graph based on the planarity testing algorithm of Hopcroft and Tarjan [HT]. The facts that sometimes there is more than one embedding, and that outerplanar graphs may be embedded in a nonouterplanar way, do not affect our results.

3.1 Outerplanar Triangulation of a Tree

We shall describe the outerplanar triangulation of trees of a special form. All the trees we triangulate will be of this form. The triangulation coincides, for a certain embedding, with that of the previous section.

Let T be a tree with root r such that r has more than one child, and r's rightmost child, s, is a leaf. The outerplaner triangulation of a tree is formed by adding edges from every vertex v (except r and s) to its nearest-ancestor-right-child: $NARC(v)$. $NARC(v)$ is found by moving from v towards the root until a vertex is found which has a child just to the right of the one that leads to v: $NARC$(v) is that just-to-the-right child. Informally, $NARC(v)$ is the first vertex on the "previous branch" to v in a depth-first traversal of the tree.

In Fig. 3.1 the new edges added according to this triangulation method are shown as broken lines. Let TRI(T) denote the triangulation of T. The proof of the following lemma is immediate.

Lemma 3. *Let T be a tree. The triangulated graph $TRI(T)$ is outerplanar.* ∎

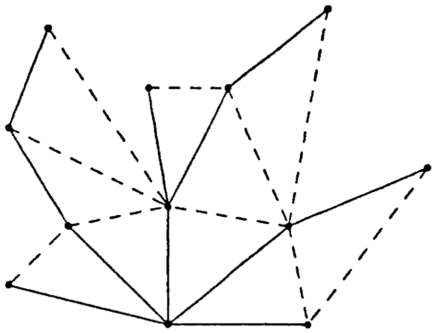

Figure 3.1 Triangulating a tree

3.2 Outerplanar Triangulation and Dual of a Planar Graph

Let G be a planar graph, and suppose that the consitucnt 2-connected blocks and tree blocks have been determined. We would like to triangulate the exterior face so that all the vertices on the exterior face of the graph will be on the exterior face of the triangulated graph. Triangulating the interior of a 2-connected block does not change the exterior face, but triangulating a tree block might. The triangulation of the 2-connected blocks is done separately from that of the tree blocks.

The interior of a 2-connected block should be triangulated in a way that will ensure a "simpler" dual graph. The simplicity measure is defined with respect to the width of the graph.

Definition: Let G be a planar graph. Define the distance *(from the exterior* face) of a vertex, say v, to be the length of the minimal path that connects v to a vertex on the exterior face of G. The distance of a face is defined to be the minimum over all the distances of its vertices. The *width* of *G* is the maximum over all the distances of its vertices.

The width of a tree is zero. Similarly the width of an outerplanar graph (with an outerplanar embedding) is also zero. The width is a function of the embedding of the graph on the plane; therefore, an outerplanar graph might have width greater than zero, in the case that its embedding is not outerplanar.

The notion of width is close to the notion of dual escape number defined by Hong and Rosenberg [HR]. However, the width does not increase after triangulation whereas the dual escape number may increase.

The *outerplanar* triangulated graph of G is denoted by TRI(*G*), and is obtained in the following way:

1. Triangulate the internal region of each 2-connected block by adding a maximal number of internal edges that do not cross, as follows: first add a maximal number of edges to vertices of distance zero, then to those of distance one and so forth, level by lcvcl.

2. Triangulate a branching tree as follows. Let v be the vertex along the face of a 2-connected block at which the tree branches; choose v to be the root of the tree. Let *u*

be the first vertex to the right of v along the face of the 2-connected block. Such a u must exist. Consider the edge (u; v) as the rightmost branch of v and triangulate the tree as described in the previous subsection.

Definition: Let G be a planar graph and $\mathrm{TRI}(G)$ be its outerplanar triangulation. The dual graph, $\mathrm{DUAL}(\mathrm{TRI}(G))$, of $\mathrm{TRI}(G)$ is obtained by inserting a vertex in each triangle of $\mathrm{TRI}(G)$ and a vertex outside each edge of the outer cycle of it; the edges of $\mathrm{DUAL}(\mathrm{TRI}(G))$ connect vertices that are on the two sides of edges of $\mathrm{TRI}(G)$.

Observe that the degree of DUAL(TRI(G)) is three, and that it is a planar graph with a planar embedding. Another fact about DUAL(TRI(G)) is that the only vertices of degree one are on the exterior face. The dual graph defined in graph theory [H] is different from the one just defined. In graph theory there is a single vertex on the exterior face, while in our case there is a vertex outside each edge of the exterior face.

3.3 Reduction of Planar Graphs

The construction used in Section 2 to lay out outerplanar graphs can be used to lay out any graph over its dual triangulation. This will allow us to embed constant width planar graphs in linear area.

Lemma 4. *Let G be a planar graph but not a tree, with vertices of degree at most 4. If $\mathit{DUAL}(\mathit{TRI}(G))$ has an orientation preserving rectilinear embedding without crossovers, with area $\leq A$, then G has an embedding of the same type with area $\leq 81A$.*

Proof: The proof is similar to the argument used in Theorem 2 to embed an outerplanar graph, given a linear embedding for its dual tree. Again, there are regions in and around $\mathrm{DUAL}(\mathrm{TRI}(G))$, each of which contains exactly one vertex of $\mathrm{TRI}(G)$. And the same argument shows that every edge of G need only cross one edge of $\mathrm{DUAL}(\mathrm{TRI}(G))$. Thus, since at most 4 edges leave a region, expanding the grid 9 times horizontally and vertically will allow G to be embedded on top of $\mathrm{DUAL}(\mathrm{TRI}(G))$. ∎

Therefore, a planar graph can be laid out over its dual triangulation, and the latter graph can be laid out in a similar manner, etc. If this process eventually leads to the problem of laying out a tree then the whole construction goes through.

Definition: We say that a planar graph G can be *reduced* to H if there exists a series of graphs $G_1, G_2, \ldots, G_k$ and $T_1, T_2, \ldots, T_{k-1}$ such that:

1. G_1 = G, and $G_k = H$;
2. T_i is the triangulated graph of G_i;
3. G_{i+1} is the dual triangulated graph of T_i.

A *reduction* step is a triangulation followed by a taking of the dual.

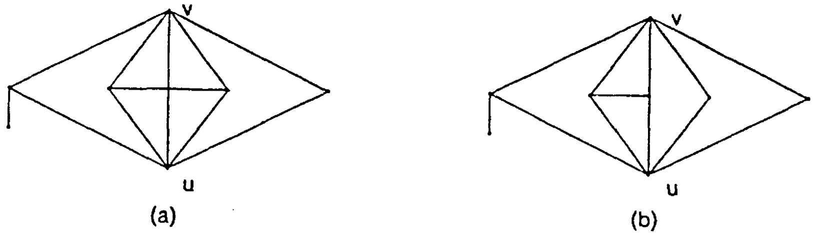

Figure 3.2 (a) honeycombed graph (b) non-honeycombed graph

'Theorem 5. *Let G be a planar graph with n vertices and degree at most* 4. *If G can be reduced to a tree* in *f reduction steps, for some constant f, then G has a planar rectilinear embedding with area $\leq$ Kn, for some constant K.*

Proof: When $f = 0$ then G is a tree, and Theorem 1 shows that there is an embedding of the required type in area $\leq kn$ for some constant *k*. Otherwise, let m be the number of vertices in DUAL(TRI(G)).

Assume by induction that DUAL(TRI(G)) has an embedding with area $\leq K_1m$. This is certainly true for 'the base case. There is a one-one correspondence between the edges of DUAL(TRI(G)) and the edges of TRI(G) (the triangulation of G); let that number of edges be e. Now DUAL(TRI(G)) is connected, so $m - 1 \leq e$. TRI(G) is planar and has n vertices, so $e \leq 2n - 3$, and hence $m \leq 2n$. Finally, Lemma 4 can bc used to show that G has an embedding in area $\leq 2 \cdot 81 \cdot K_1 n$. The final K will be the result of multiplying f constants together, and so it will itself be constant. ∎

Naturally, the above thcorem is more interesting if we can state the conditions under which a planar graph reduces to a trcc in a constant number of steps. In the remainder of this section we show that every planar graph reduces to a tree eventually, and that the number of reduction steps is proportional to the width of the graph.

Some of the results depend on the fact that the dual triangulated. graphs developed in the above layout procedure have a specific form.

Definition: A *honeycombed* graph is a planar graph in which every internal face is a simple cycle and every two faces have at most one edge in common.

Fig. 3.2 illustrates a honeycombed and a non-honeycombed graph. A consequence of this definition is that a honeycornbcd graph docs not have any internal vertcx of dcgrce two. General honeycombed graphs have the problcm that one can have multiple edges bctwccn a pair of vertices. In Fig. 3.2(a), vertices v and u are common to two interior faces; therefore, the edge (u, v) can be added to either of thosefaces without affecting planarity. The graphs we use later on are honeycombed of degree 3. The following theorem proves that the.above case cannot happen in honeycombed graphs of degree 3.

Theorem 6. *Let G be a honeycombed graph of degree 3. Any two non-adjacent vertices is.common to at most one internal.face. Every* pair *of adjacent vertices are common only to the two* faces *on either* side of *the edge between them.*

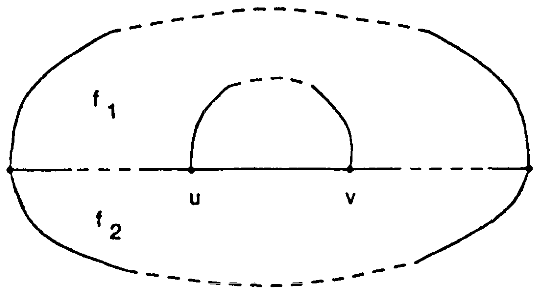

Figure 3.3 Vertices common to 3 internal faces

Proof: By definition, every internal face of a honeycombed graph is a cycle. Since the degree of every vertex is at most 3, if two cycles have a common vertex then one of the edges leaving that vertex is also common to the two cycles.

This fact implies that if two non-adjacent vertices are common to two cycles, then there are at least two edges common to them; this contradicts the assumption that the graph is honeycombed.

The case that is left is where two adjacent vertices are common to several internal faces. Fig. 3.3 shows the only way that two vcrtices, u and v, can share 3 internal faces in a graph of degree 3. It can be seen that that graph is not honeycombed because faces f_1 and f_2 have at least two edges in common. Therefore, two adjacent vertices cannot be common to 3 or more internal faces in a honeycombed graph of degree 3. ▮

We want to be able to triangulate a face by adding edges from any given vertex to all the non-adjacent vertices of that face, but this might create multiple edges between a pair of vc:ticcs if they share morc than one face. One of our later proofs requires that there be no multiple cdgcs; a corollary of Theorcm 6 is that this cannot happcn in honeycombed graphs of degree 3.

Corollary 7. *Let G be a honeycombed graph* of *degree* 3. *Let F be* any internal face in *G and v 'be an arbitrary vertex on F. Then F can be triangulated with edges leaving v, without creating multiple edges.*

Proof: Every face in G is a simple cycle. Theorem 6 implies that a vcrtcx is adjaccnt to exactly' two vertices along the face to which it belongs. Moreover, an edge between a vertcx and any non-adjacent vertex on thc face can be drawn only on the inside the face *F*. Therefore, we can triangulate *F* with edges leaving any of its vertices without producing multiple edges in the graph. ▮

Thc following lcmma justifies the way wc dcfined the triangulation.. In honeycombed graphs of dcgrcc 3, thc triangulation does not produce faces of maximum distance if the original graph does not have any.

Lemma 8. *Let G be a honeycombed graph of degree 3, and let l be the maximum distance in G. The triangulated graph $TRI(G)$ contains a face of distance l only if G contains such a face.*

Proof: If $l = 0$, then obviously all the vertices of $\mathrm{TRI}(G)$ are of distance zero, as in G. So assume that $l > 0$ and that every face contains a vertex of distance less than l.

The graph G is a honeycombed graph, so every face is a simple cycle. Thus, every cycle contains a vertex of distance less than l. By Corollary 7 we can use that vertex to triangulate the face. This triangulation would produce triangles, each of which contains a vertex' of distance less than l. But all the faces in the triangulated graph $\mathrm{TRI}(G)$ are triangles. Thus, we have proved that all the faces in $\mathrm{TRI}(G)$ contain vertices of distance less than l, which implies that none of its faces contain only vertices of distance 1. ∎

Lemma 9. *Let G be a honeycombed graph of degree 3. The width of $TRI(G)$ is not greater than the width of G.*

Proof: By Lemma 3, the triangulation of the tree blocks does not change the width of the graph. On the other hand, the width of a 2-connected block can only be reduced by triangulation. ∎

The following property of dual graphs enables us to use these results about honeycombed graphs.

Lemma 10. *Let G be a planar graph. $DUAL(TRI(G))$ is a honeycombed* graph *of degree* 3.

Proof: Let $\mathrm{TRI}(G)$ be the triangulated graph of G used to produce $\mathrm{DUAL}(\mathrm{TRI}(G))$. It is clear that DUAL(TRI(G)) is of degree 3. Also, every internal vertex of $\mathrm{DUAL}(\mathrm{TRI}(G))$ is of degree 3, because it is placed in the center of a triangle in $\mathrm{TRI}(G)$. This proves that every internal face of DUAL(TRI(G)) is a simple cycle: the degree 3 restriction eliminates the possibility of trec leaves on the face. Also, for there to be a non-simple cycle, there would have to be vertices of degree at least 4.

It remains to show that every two internal faces have at most one edge in common. We will prove' that each internal face is produced by a single vertex of $\mathrm{TRI}(G)$; that is, every edge along the face corresponds to an edge leaving that vertex.

Consider a general internal cycle of $\mathrm{DUAL}(\mathrm{TRI}(G))$, shown in Fig. 3.4. Every vertex of DUAL(TRI(G)) is a center of a triangle of $\mathrm{TRI}(G)$. Therefore, the edges (u_k, u_1) and (u_1, u_2) are crossed by two cdgcs that leave the same vcr tex, say r. The edge (u_2, u_3) will also be crossed by an edge leaving r. Continuing, one finds that every edge in the cycle is crossed by a vertex leaving r. By construction, every edge of $\mathrm{DUAL}(\mathrm{TRI}(G))$ crosses exactly one edge of $\mathrm{TRI}(G)$. Thus, there can be exactly one vertex of $\mathrm{TRI}(G)$ inside the face, which proves our claim.

So, in every face of $\mathrm{DUAL}(\mathrm{TRI}(G))$ there is one vertex of $\mathrm{TRI}(G)$, with edges leaving it and crossing all the edges along the face. Since the graph is planar and every two vertices are connected by at most one cdgc, cvcry two internal cycles have at most one cdgc in common. This completes the proof that $\mathrm{DUAL}(\mathrm{TRI}(G))$ is a honeycombed graph-of degree 3. ∎

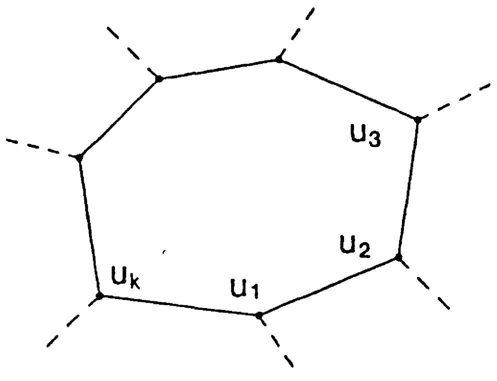

Figure 3.4 Internal cycle of DUAL(TRI(G))

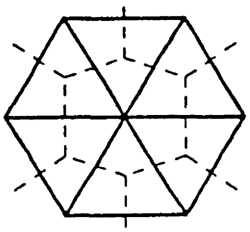

Figure 3.5 Wheel in TRI(G) (solid lines)
and the corresponding face in DUAL(TRI(G)) (broken lines)

The proof of the following fact is similar to part of the proof of Lemma 10.

Lemma 11. *Let G be a planar graph. Every internal face of $DUAL(TRI(G))$ is a dual of a 'wheel" **subgraph** of $TRI(G)$ (**see** Fig. 9.5).* ∎

We want to show that by triangulating and constructing the dual triangulated graph we reduce the width of the graph, so that the new graph is "simpler", and after a finite number of steps we will reach a graph of width zero. But the situation is not quite so. simple: one application of duality and triangulation is not enough to reduce the width of the graph. We shall prove that two consecutive applications are enough. First, here is a lemma used in the proof.

Lemma **12.** *Let T be a triangulated planar graph and T_1 be $TRI(DUAL(T))$. Let v be a non-leaf **vertex** in T_1 and F be the corresponding triangle in T. The distance of v* in *T_1 is equal to the distance of F in T.*

Proof: The dcfinition of dual implies that every triangle of distance zero in T corresponds to a vertex of distance zero in DUAL(T), and thus also in TRI(DUAL(T)). The rest of the proof is by induction. Assume that the claim holds for all triangles of distance $k > 0$ and lct F be a triangle in T of distance $k + 1$, and v be the corresponding vcrtcx in T_1.

Dccause F is of distance $k + 1$, onc of its vertices, say u, is of distance $k + 1$ in T. This vertex has a neighboring vertex (not on F).of distance k; moreover, thcse vertices are on a triangle of distance k in T. Let v' bc.the vertex corresponds to this triangle-in T_1. By the inductive assumption, the distance of v' in T_1 is k.

By Lcmrna 11, v and v' are on the same face in DUAL(T). Thcrcfore, there exists a triangulation of the face corresponding to u in which the distance of v becomes at most $k + 1$, which proves the lemma. ∎

Theorem 13. *Let T be a triangulated planar graph. Let*

$$T_1 = TRI(DUAL(T))$$
$$T_2 = TRI(DUAL(T_1)).$$

If the width of T is *positive, then the width of T_2 is less than the width of T.*

Proof: Let M be the set of vertices of T with maximal distance. Consider two cases:

'Case *A:* Every triangle of T contains a vertex not in *M*. In this case the distance of every triangle, by definition, is less than the maximum distance in *T*. Thus, the previous lemma implies that the width of T_1 is less than that of T. The width of T_2 cannot be greater than that of T_1; thus the theorem is proved.

Case B: There exist trianglcs in T with all their vertices in M. By the previous lemma it is clear that the width of T_1 cquals the width of *T*. Let M_1 be the set of vertices having the maximum distance in T_1. We shall prove that Casc B could not hold for T_1.

Assume to the contrary that T_1 contains a triangle with all its vertices in M_1. By Lemma 8, DUAL(T_1) contains a face all of whose vertices have the maximum distance. A face in a dual triangulated graph is dcrivcd from a wheel in the triangulated graph, a s cxplained in Lcmma 11. Thus, T_1 contains such a wheel, made up of triangles with distance equal to the maximum distance (width) of T_1; but this is impossible because the planarity of T_1 implies that the distance of the center of the wheel is greatcr than the distance of the face of the wheel. Therefore, not all the vertices of these triangles have the maximum distance. This proves that Case B does not hold for T_1, which implies, by Case A, that the width of T_2 is smaller than that of T_1. This completes the proof of the theorem. ∎

Theorem 13 implies that the reduction decreases the width, or more precisely:

Theorem 14. *Every planar graph can be reduced to a free in a number of steps proportional to its width.*

Proof: Let G be a planar graph. Let w be the width of G, and $k = 2w + 1$. Define G_1 to be G, and for $1 \leq i \leq k$ define T_i = TRI(G_i) and G_{i+1} = DUAL(T_i). Choose H to be G_k.

By Thcorcm 13, the width of T_{2i+1} is smaller than that of T_{2i-1}. Thcrcforc, cventually thcrc will bc a graph of width zero. Lemma 3 implies that further reduction steps leave it with width zero. Thus, H is a tree, which. proves the theorem. ∎

So finally, we come to thc main result of this section.

Theorem 15. *If G is a member of a class of planar graphs of degree 4 whose width does not depend on the size of the graph, then G has a linear area* rectilinear *embedding without crossovers.*

Proof: Let w be the width of G. Theorem 14 shows that at most $2w + 1$ reduction steps suffice to reduce G to a tree, so the conditions of Theorem 5 are satisfied. ∎

§4 $\mathcal{NP}$-Completeness of Optimal Forest Embedding

Given a forest and an integer *A*, the forest layout problem is to find whether or not there is a planar rectilinear embedding with area less than or equal to *A*. In this section we will show that the forest layout problem is A/P-complete. This will be done by transforming the S-partition problem to it.

In the 3-partition problem there is a set of integers $x_1, \ldots, x_{3m}$ such that

$$\sum_{i=1}^{3m} x_i = mB$$

and $B/4 < x_i < B/2$ for $1 \leq i \leq 3m$. Thc question is whether the set can be partitioned into m disjoint sets such that each set sums to *B*. This problem is known to be $\mathcal{NP}$-complete [GJ].

Consider the tree in Fig. 4.1. Call it the frame tree. There are vertices at every grid point except for $m = 2n$ holes of size *B*. (The case for m odd will be considered later; it is, just a trivial modification.)

Lemma 16. *The only embedding of the frame tree with a bounding box area of*

$$(472 + 2B + 3) \times (2B + 3)$$

or less and leaving mB free grid points is the embedding given in Fig. *4.1.*

Proof: The tree has $(4n + 2B + 3) \times (2B + 3) - mB$ vertices, so the embedding is required to use every grid point for a vcrtcx or else leavc it frcc. This mcans that no cdgc of thc tree can be stretched to a path of 2 units, for that would take up a grid point in the middle that is not used for embedding a graph vertex.

The vertical branches of the framc tree cannot bc bent because the degree 4 vertices would rcquire an edge stretch or a pcrmutation of the order of the branches, and such a bcnding would run into the corner areas. The corner areas can only be bent inwards (because of the degree 3 nodcs), and thcrc is no room to bcnd inwards. Thcrc are some lincs that can be bent outwards but this just makcs the bounding box bigger.

Therefore, the given layout is the only one possible. ∎

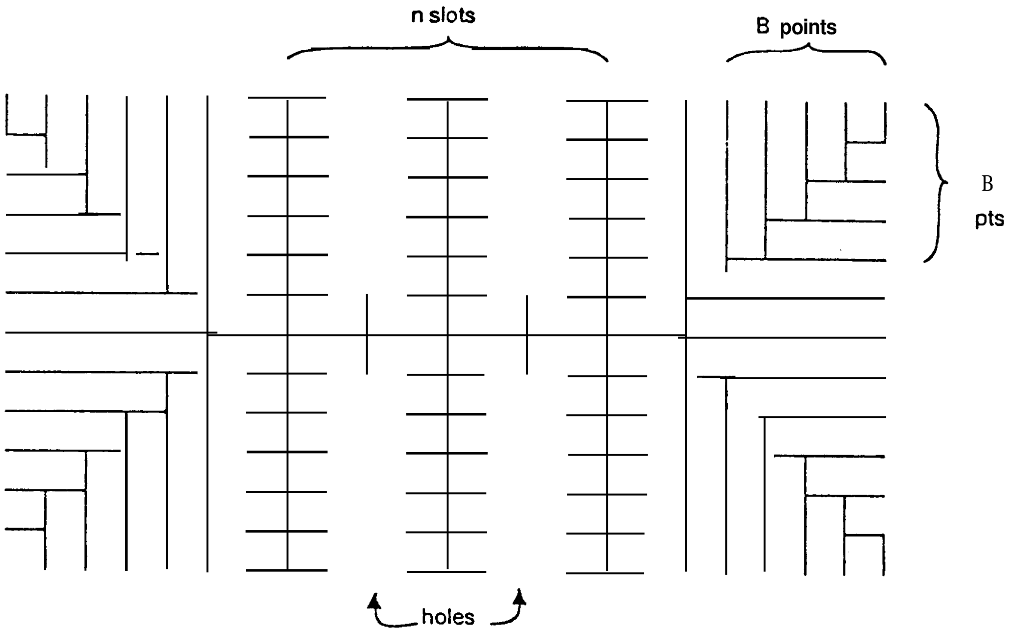

Figure 4.1 Frame Tree for $n = 2$, $B = 5$

Theorem 17. *The forest layout problem is $\mathcal{NP}$-complete.*

Proof: Given an instance of the 3-partition problem, construct the frame tree and add 3m other pieces, unconnected to that tree: for each x_i there is a piece consisting of x; vertices joined into a line by x; — 1 edges. If m is odd, use the frame graph for the next higher even. number and fill in one of the vertical holes. Now we claim that the 3-partition problem instance has a solution iff there is an embedding of this forest with a bounding box area of $(4n + 2B + 2) \times (2B + 3)$. For, by the lemma, if there is such an embedding then it must be as shown in Fig. 4.1 with the extra pieces filling up the holes. Since all the grid points are to be used, this gives a solution to the 3-partition problem, because the size restrictions on the x's imply that there must be exactly three pieces in each hole. Conversely, given a solution to the 3-partition problem, a suitable embedding can be found by filling the holes in the frame tree with the pieces corresponding to the partitioned sets.

The given transformation can clearly be done in polynomial time. Also, the layout problem is in $\mathcal{NP}$ because one can simply guess a mapping of all the vertices to grid points and then verify that the edges can all be put along the connecting lines. Therefore, the forest layout problem is J/P-complete. ∎

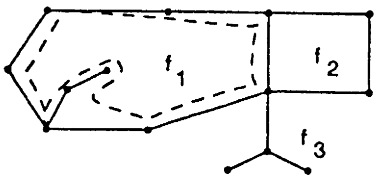

Figure A.1 f_1, f_2 and f_3 are the faces.
The dotted line shows the face f_1.

Appendix

This appendix presents the basic graph theory definitions used in the paper. Some of the standard definitions are not given here; they may be found in [H].

A graph G = (V, E) consists of a set V of vertices and a set E of edges. The degree of a vertex is the number of edges incident on it. The degree of a grpah is the maximum over the degrees of its vertices. A planar graph is a graph that can be embedded in the plane such that no two edges intersect. In the paper we consider only graphs that arc planar, and therefore we assume that the graph is always given with its embedding on the plane.

A path in a graph is a sequence $v_1, e_1, v_2, \ldots, e_k, v_k$, with $v_i \in V, e_i \in E$, and $e_i \neq e_j$ for $i \neq j$. A cycle is a closed path. A simple cycle is a path in which $v_1 = v_k$, but otherwise $v_i \neq v_j$ when $i \neq j$.

Let $F \subseteq V$ be a subset of vertices. We say that F forms a face of G (of the embedding of G) if:

i. We can draw a closed line touching all of the vertices in F, but not cutting across any edges, so that all of the rest of the graph is embedded on one side of the line (see Fig. A. 1);

ii. F is maximal: i.e., for every $F' \supset F$, F' does not satisfy (i).

Note that a face is bounded by a simple cycle with tree branches attached to it.

An outerplanar graph is a planar graph, all of whose vertices form a single face. Observe that every tree is an outerplanar graph, because all of its vertices are on one face, as shown in Fig. A.2.

Let G be a connected planar graph. Choose a face of G and call it an *exterior face.* All the rest of the graph is embedded inside that face. We choose the unbounded region implied by the embedding, such as f_3 in Fig. A.l, to be the exterior face.

If the graph is 2-connected, then the exterior face is a cycle called the exterior cycle of the graph.

The exterior face of the graph induces a decomposition of the graph into blocks; each block being either a 2-connected block or a tree block. A *2-connected* block consists of a cycle along its exterior face of vertices from the exterior face of G and all the subgraph

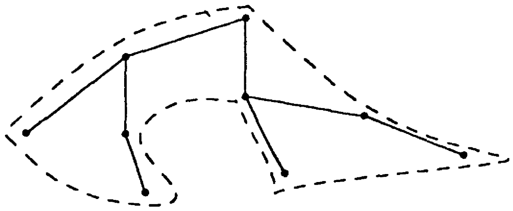

Figure A.2 A tree is outerplanar

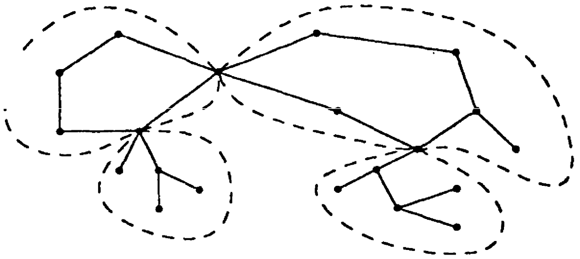

Figure A.3 The blocks of a planar graph

embedded inside that cycle. A tree block is a l-connected subgraph, all of whose vertices are on the exterior face. If a graph is connected and contains a cycle, then all the tree blocks branch from the exterior cycle of 2-connectcd blocks. (See Fig. A.3 for an example of these definitions.)

A maximal planar (respectively *outerplanar*) graph is one to which no edge can be added without losing planarity (respectively, outerplanarity).

Ref ercnces

[B] Brehaut, W. An Efficient Outerplanarity Algorithm. *Proc. 8th S-E Conf. Combinatorics, Graph Theory, and* Computing, 1977,pp. 99-113.

[BK] Brent, R.P., and Kung, H.T. On the Area Embedding of Binary Tree Layouts. Technical Report *TR-CS-79-07, The* Australian National University, Department *of* Computer *Science,* July, 1979.

[CS] Cutler, M., and Shiloach, Y. Permutation Layout. Networks 8,1978, pp. 253-278.

[FP] Fischer, M.J., and Paterson, M.S. Optimal Tree Layout. *12th* Annual ACM *STOC,* 1980.

[GJ] Garey, M.R., and Johnson, D.S. Computers and *In* tractability. Freeman, San Francisco, 1979.

[H] Harary, F. *Graph Theory.* Addison-Wesley, Reading, Mass., 1969.

[HR] Hong, Jia-Wei, and Rosenberg, A.L. Graphs that are Almost Binary Trees. 13th Annual ACM *STOC,* Milwaukee 1981, pp. 334-341.

[HT] Hopcroft, J., and Tarjan, R. Efficient Planarity Testing. *JACM* 21:4, *1974,* pp. 549-568.

[L] Leiserson, C.E. Area-Efficient Graph Layouts (for VLSI). *21st Annual IEEE* Symposium on FOCS, 1980.

[M] Mitchell, S.L. Linear Algorithms to Recognize Outerplanar and Maximal Outerplanar Graphs. *Information Processing Letters* 9:5, 1979, pp. 229-232.

[R] Rosenberg, A.L. On Embedding Graphs in Grids. Technical *Report RC 7559 (#2668),* IBM Wat*son Research Center,* February, 1979.

[V] Valiant, L.G. Universality Considerations in VLSI Circuits. *IEEE Trans. on* Computers, February 1981, pp. 135-140.

[W] Williamson, S.G. The Tree of Paths of a Graph. Publication *de l'Institu t de* Recherche *Mathématique* Avancke, Laboratoire *Associé* au C.N.R.S., *n*°1, *Université Louis* Pasteur, Strasbourg, France, 1980.