A SEPARATOR THEOREM FOR PLANAR GRAPHS

by

Richard J. Lipton and Robert E. Tarjan

STAN-CS-77-627
OCTOBER 1977

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

A Separator Theorem for Planar Graphs

Richard J. Lipton */
Computer Science Department
Yale University
New Haven, Connecticut 06520

Robert Endre Tarjan **/
Computer Science Department
Stanford University
Stanford, California 94305

August, 1977

Abstract.

Let G be any n-vertex planar graph. We prove that the vertices of G can be partitioned into three sets A, B, C such that no edge joins a vertex in A with a vertex in B, neither A nor B contains more than $2n/3$ vertices, and C contains no more than $2\sqrt{2}\sqrt{n}$ vertices. We exhibit an algorithm which finds such a partition A, B, C in O(n) time.

Keywords: algorithm, divide-and-conquer, nested dissection, planar graph, separator.

*/ This research was supported in part by the U.S. Army Research Office, Grant No. DAAG 29-76-G-0338.

**/ This research was supported in part by National Science Foundation grant MCS-75-22870 and by the Office of Naval Research contract N00014-76-C-0688.

Reproduction in whole or in part is permitted for any purpose of the United States Government.

1. Introduction.

A useful method for solving many kinds of combinatorial problems is "divide-and-conquer" [1]. In this method the problem of interest is divided into two or more smaller problems. The subproblems are solved by applying the method recursively, and the subproblem solutions are combined to give the solution to the original problem. Three things are necessary for the success and efficiency of divide-and-conquer: (i) the subproblems must be of the same type as the original and independent of each other (in a suitable sense); (ii) the cost of solving the original problem given the solutions to the subproblems must be small; and (iii) the subproblems must be significantly smaller than the original. One way to guarantee that the subproblems are small is to make them all roughly the same size [1].

We wish to study general conditions under which the divide-and-conquer approach is useful. Consider problems which are defined on graphs. Let S be a class of graphs*/ closed under the subgraph relation (i.e., if $G_1 \in S$ and G_2 is a subgraph of G_1, then $G_2 \in S$). An $f(n)$ -separator theorem for S is a theorem of the following form:

> There exist constants $a < 1$, $\beta > 0$ suchthatif G is any n-vertex graph in S, the vertices of G can be partitioned into three sets A, B, C such that no edge joins a vertex in A with a vertex in B, neither A nor B contains more than an vertices, and C contains no more than $\beta f(n)$ vertices.

If such a theorem holds for the class of graphs S, and if the appropriate vertex partitions A, B, C can be found fast, then a n-umber of problems defined on graphs in S can be solved efficiently using divide-and-conquer. For a given graph G in S, the sets A and B define the subproblems. The cost of combining the subproblem solutions is a function of the size of C (and thus of $f(n)$).

*/ The appendix contains the graph-theoretic definitions used in this paper.

Previously known separator theorems include the following:

(A) Any n-vertex binary tree can be separated into two subtrees, each with no more than $2n/3$ vertices, by removing a single edge. For an application of this theorem, see [13].

(B) Anyn-vertextree can be divided into two parts, each with no more than $2n/3$ vertices, by removing a single vertex.

(C) A grid graph is any subgraph of the infinite two-dimensional square grid illustrated in Figure 1. A &-separator theorem holds for the class of grid graphs. For an application, see [5].

(D) A one-tape Turing machine graph [16] is a graph representing the computation of a one-tape Turing machine. A &-separator theorem holds for such graphs. For an application, see [15].

[Figure 1]

One might conjecture that the class of all suitably sparse graphs has an $f(n)$ -separator theorem for some $f(n) = o(n)$. However, the following result of Erdös, Graham, and Szemerédi [4] shows that this is not the case,

Theorem C. For every $\epsilon > 0$ there is a positive constant $c = c(\epsilon)$ such that almost all */ graphs G with $n = (2+\epsilon)k$ vertices and ck edges have the property that after the omission of any k vertices, a connected component of at least k vertices remains.

*/ By "almost all" we mean that the fraction of graphs possessing the property tends with increasing n to one.

Although sparsity by itself is not enough to give a useful separator theorem, planarity is. In Section 2 of this paper we prove that a $\sqrt{n}$ -separator theorem holds for all planar graphs. In Section 3 we provide a linear-time algorithm for finding a vertex partition satisfying the theorem. This algorithm and the divide-and-conquer approach combine to give efficient algorithms for a wide range of problems on planar graphs. Section 4 mentions some of these applications, which we shall discuss more fully in a subsequent paper.

2. Separator Theorems.

To prove our results we need to use three facts about planarity.

Theorem 1 (Jordan Curve Theorem [6]). Let C be any closed curve in the plane. Removal of C divides the plane into exactly two connected regions, the "inside" and the "outside" of C .

Theorem 2 [7]. Any n-vertex planar graph with $n \geq 3$ contains no more than $3n-6$ edges.

Theorem 3 (Kuratowski's Theorem [12]). A graph is planar if and only if it contains neither a complete graph on five vertices (Figure 2(a)) nor a complete bipartite graph on two sets of three vertices (Figure 2(b)) as a generalized subgraph.

[Figure 2]

From Kuratowski's Theorem we can easily obtain the following lemma and its corollary.

Lemma 1. Let G be any planar graph. Shrinking any edge of G to a single vertex preserves planarity.

Proof. Let G* be the shrunken graph, let (x_1,x_2) be the edge shrunk, and let x be the vertex corresponding to x_1 and x_2 in G* . If G* is not planar then G* contains a Kuratowski graph as a generalized subgraph. But this subgraph corresponds to a Kuratowski graph which is a generalized subgraph of G . Figure 3 illustrates the possibilities. □

[Figure 3]

Corollary 1. Let G be any planar graph. Shrinking any connected subgraph of G to a single vertex preserves planarity.

Proof. Immediate from Lemma 1 by induction on the number of vertices in the subgraph to be shrunk. □

In some applications it is useful to have a result more general than the kind of separator theorem described in the introduction, We shall therefore consider planar graphs which have non-negative costs on the vertices. We shall prove that any such graph can be separated into two parts, each with cost no more than two-thirds of the total cost, by removing $O(\sqrt{n})$ vertices. The desired separator theorem is the special case of equal-cost vertices.

Lemma 2. Let G be any planar graph with non-negative vertex costs summing to no more than one. Suppose G has a spanning tree of radius r. Then the vertices of G can be partitioned into three sets A, B, C, such that no edge joins a vertex in A with a vertex in B, neither A nor B has total cost exceeding $2/3$, and C contains no more than $2r+1$ vertices, one the root of the tree.

Proof. Assume no vertex has cost exceeding $1/3$; otherwise the lemma is true. Embed G in the plane. Make each face a triangle by adding a suitable number of additional edges. Any non-tree edge (including each of the added edges) forms a simple cycle with some of the tree edges. This cycle is of length at most $2r+1$ if it contains the root of the tree, at most $2r-1$ otherwise. The cycle divides the plane (and the graph) into two parts, the inside and the outside of the cycle. We claim that at least one such cycle separates the graph so that neither the inside nor

the outside contains vertices whose total cost exceeds $2/3$, This proves the lemma.

Proof of claim. Let (x,z) be the non-tree edge whose cycle minimizes the maximum cost either inside or outside the cycle. Break ties by choosing the non-tree edge whose cycle has the smallest number of faces on the same side as the maximum cost. If ties remain, choose arbitrarily.

Suppose without loss of generality that the graph is embedded so that the cost inside the (x,z) cycle is at least as great as the cost outside the cycle. If the vertices inside the cycle have total cost not exceeding $2/3$, the claim is true. Suppose the vertices inside the cycle have total cost exceeding $2/3$. We show by case analysis that this contradicts the choice of (x,z) . Consider the face which has (x,z) as a boundary edge and lies inside the cycle. This face is a triangle; let y be its third vertex. The properties of (x,y) and (y,z) determine which of the following cases applies. Figure 4 illustrates the cases.

[Figure 4]

(1) Both (x,y) and (y,z) lie on the cycle. Then the face (x,y,z) is the cycle, which is impossible since vertices lie inside the cycle.

(2) One of (x,y) and (y,z) (say (x,y)) lies on the cycle. Then (y,z) is a non-tree edge defining a cycle which contains within it the same vertices as the original cycle but one less face. This contradicts the choice of (x,z) .

(3) Neither (x,y) nor (y,z) lies on the cycle.

(a) Both (x,y) and (y,z) are tree edges. This is impossible since the tree itself contains no cycles.

(b) One of (x,y) and (y,z) (say (x,y)) is a tree edge. Then (y, z) is a non-tree edge defining a cycle which contains one less vertex (namely y) within it than the original cycle. The inside of the (y,z) cycle contains no more cost and one less face than the inside of the (x,z) cycle. Thus if the cost inside the (y,z) cycle is greater than the cost outside the cycle, (y,z) would have been chosen in place of (x,z) .

On the other hand, suppose the cost inside the (y, z) cycle is no greater than the cost outside. The cost outside the (y,z) cycle is equal to the cost outside the (x,z) cycle plus the cost of y . Since both the cost outside the (x,z) cycle and the cost of y are less than $1/3$, the cost outside the (y,z) cycle is less than $2/3$, and (y,z) would have been chosen in place of (x,z) .

(C) Neither (x,y) nor (y,z) is a tree edge. Then each of (x,y) and (y,z) defines a cycle, and every vertex inside the (x,z) cycle is either inside the (x,y) cycle, inside the (y,z) cycle, or on the boundary of both. Of the (x,y) and (y,z) cycles, choose the one (say (x,y)) which has inside it more total cost. The (x,y) cycle has no more cost and strictly fewer faces inside it than the (x,z) cycle. Thus if the cost inside the (x,y) cycle is greater than the cost outside, (x,y) would have been chosen in place of (x,z) .

On the other hand, suppose the cost inside the (x, y) cycle is no greater than the cost outside. Since the inside of the (x,z) cycle has cost exceeding $2/3$, the (x,y) cycle and its inside together have cost exceeding $1/3$, and the outside of the (x,y) cycle has cost less than $2/3$. Thus (x,y) would have been chosen in place of (x,z).

Thus all cases are impossible, and the (x,z) cycle satisfies the claim. □

Lemma 3. Let G be any n-vertex connected planar graph having non-negative vertex costs summing to no more than one. Suppose that the vertices of G are partitioned into levels according to their distance from some vertex v, and that $L(\ell)$ denotes the number of vertices on level ℓ. If r is the maximum distance of any vertex from v, let $r+1$ be an additional level containing no vertices. Given any two levels ℓ_1 and ℓ_2 such that levels 0 through ℓ_1-1 have total cost not exceeding $2/3$ and levels ℓ_2+1 through $r+1$ have total cost not exceeding $2/3$, it is possible to find a partition A, B, C of the vertices of G such that no edge joins a vertex in A with a vertex in B, neither A nor B has total cost exceeding $2/3$, and C contains no more than $L(\ell_1) + L(\ell_2) + \max\{0, 2(\ell_2-\ell_1-1)\}$ vertices.

Proof. If $\ell_1 \geq \ell_2$, let A be all vertices on levels 0 through ℓ_1-1, B all vertices on levels ℓ_1+1 through r, and C all vertices on level ℓ_1. Then the lemma is true. Thus suppose $\ell_1 < \ell_2$. Delete the vertices in levels ℓ_1 and ℓ_2 from G. This separates the remaining vertices of G into three parts (all of which may be empty): vertices on levels 0 through ℓ_1-1, vertices on levels ℓ_1+1 through ℓ_2-1,

and vertices on levels ℓ_2+1 and above. The only part which can have cost exceeding $2/3$ is the middle part,

If the middle part does not have cost exceeding $2/3$, let A be the most costly part of the three, let B be the remaining two parts, and let C be the set of vertices on levels ℓ_1 and 12. Then the lemma is true.

Suppose the middle part has cost exceeding $2/3$. Delete all vertices on levels ℓ_2 and above and shrink all vertices on levels ℓ_1 and below to a single vertex of cost zero. These operations preserve planarity by Corollary 1. The new graph has a spanning tree of radius $\ell_2-\ell_1-1$ whose root corresponds to vertices on levels ℓ_1 and below in the original graph.

Apply Lemma 2 to the new graph. Let A*, B*, C* be the resulting vertex partition. Let A be the set among A* and B* having greater cost, let C consist of the vertices on levels ℓ_1 and ℓ_2 in the original graph plus the vertices in C* minus the root of the tree, and let B contain the remaining vertices in G. By Lemma 2, A has total cost not exceeding $2/3$. But $A \cup C^*$ has total cost at least $1/3$, so B also has total cost not exceeding $2/3$. Furthermore C contains no more than $L(\ell_1) + L(\ell_2) + 2(\ell_2-\ell_1-1)$ vertices. Thus the lemma is true. □

Theorem 4. Let G be any n-vertex planar graph having non-negative vertex costs summing to no more than one. Then the vertices of G can be partitioned into three sets A, B, C such that no edge joins a vertex in A with a vertex in B, neither A nor B has total cost exceeding $2/3$, and C contains no more than $2\sqrt{2}\sqrt{n}$ vertices.

Proof. Assume G is connected. Partition the vertices into levels according to their distance from some vertex v . Let L(1) be the number of vertices on level ℓ . If r is the maximum distance of any vertex from v , define additional levels -1 and $r+1$ containing no vertices.

Let ℓ_1 be the level such that the sum of costs in levels 0 through ℓ_1-1 is less than $1/2$, but the sum of costs in levels 0 through ℓ_1 is at least $1/2$. (If no such ℓ_1 exists, the total cost of all vertices is less than $1/2$, and $B = C = \emptyset$ satisfies the theorem.) Let k be the number of vertices on levels 0 through ℓ_1 . Find a level ℓ_0 such that $\ell_0 \leq \ell_1$ and $|L(\ell_0)| + 2(\ell_1-\ell_0) \leq 2\sqrt{k}$. Find a level ℓ_2 such that $\ell_1+1 \leq \ell_2$ and $|L(\ell_2)| + 2(\ell_2-\ell_1-1) \leq 2\sqrt{n-k}$. If two such levels exist, then by Lemma 3 the vertices of G can be partitioned into three sets A, B, C such that no edge joins a vertex in A with a vertex in B neither A nor C has cost exceeding $2/3$, and C contains no more than $2(\sqrt{k} + \sqrt{n-k})$ vertices. But $2(\sqrt{k} + \sqrt{n-k}) \leq 2(\sqrt{n/2} + \sqrt{n/2}) = 2\sqrt{2}\sqrt{n}$. Thus the theorem holds if suitable levels ℓ_0 and ℓ_2 exist.

Suppose a suitable level ℓ_0 does not exist. Then, for $i \leq \ell_1$, $L(i) \geq 2\sqrt{k} - 2(\ell_1-i)$. Since $L(0) = 1$, this means $1 \geq 2\sqrt{k} - 2\ell_1$, and $\ell_1 + 1/2 \geq \sqrt{k}$. Thus $\ell_1 = \lfloor \ell_1 + 1/2 \rfloor > \lfloor \sqrt{k} \rfloor$, and

$$k = \sum_{i=0}^{\ell_1} L(i) > \sum_{i=\ell_1-\lfloor\sqrt{k}\rfloor}^{\ell_1} 2\sqrt{k} - 2(\ell_1-i) \geq (4\sqrt{k} - 2\lfloor\sqrt{k}\rfloor)(\lfloor\sqrt{k}\rfloor+1)/2 \geq$$

$\sqrt{k}(\lfloor\sqrt{k}\rfloor+1) > k$. This is a contradiction. A similar contradiction arises if a suitable level ℓ_2 does not exist. This completes the proof for connected graphs.

Now suppose G is not connected. Let $G_1, G_2, \ldots, G_k$ be the connected components of G , with vertex sets $V_1, V_2, \ldots, V_k$, respectively. If no connected component has total vertex cost exceeding $1/3$, let i be the minimum index such that the total cost of-. $V_1 \cup V_2 \cup \ldots \cup V_i$ exceeds $1/3$. Let $A = V_1 \cup V_2 \cup \ldots \cup V_i$, let $B = V_{i+1} \cup V_{i+2} \cup \ldots \cup V_k$, and let $C = \emptyset$. Since i is minimum and the cost of V_i does not exceed $1/3$, the cost of A does not exceed $2/3$. Thus the theorem is true.

If some connected component (say G_i) has total vertex cost between $1/3$ and $2/3$, let A = Vi , $B = V_1 \cup \ldots \cup V_{i-1} \cup V_{i+1} \cup \ldots \cup V_k$, and $C = \emptyset$. Then the theorem is true.

Finally, if some connected component (say G_i) has total vertex cost exceeding $2/3$, apply the above argument to G_i . Let A*, B*, C* be the resulting partition. Let A be the set among A* and B* with greater cost, let C = C* , and let B be the remaining vertices of G . Then A and B have cost not exceeding $2/3$ and the theorem is true,

This proves the theorem for all planar graphs. In all cases the separator C is either empty or contained in only one connected component of G. □

Corollary 2 (&Y-Separator Theorem). Let G be any n-vertex planar graph. The vertices of G can be partitioned into three sets A, B, C such that no edge joins a vertex in A with a vertex in B , neither A nor B contains more than $2n/3$ vertices, and C contains no more than $2\sqrt{2}\sqrt{n}$ vertices.

Proof. Assign to each vertex of G a cost of $1/n$. The corollary follows from Theorem 4. □

It is natural to ask whether the constant factor of $2/3$ in Theorem 1 can be reduced to $1/2$ if the constant factor of $2\sqrt{2}$ is allowed to increase. The answer is yes.

Corollary 3. Let G be any n-vertex planar graph having non-negative vertex costs summing to no more than one. Then the vertices of G can be partitioned into three sets A, B, C such that no edge joins a vertex in A with a vertex in B, neither A nor B has total cost exceeding $1/2$, and C contains no more than $\frac{2\sqrt{2}\sqrt{n}}{1-\sqrt{2/3}}$ vertices.

Proof. Let $G = (V,E)$ be an n-vertex planar graph. We shall define sequences of sets (A_i), (B_i), (C_i), (D_i) such that

(i) A_i, B_i, C_i, D_i partition V.

(ii) No edge joins A_i with B_i, A_i with D_i, or B_i with D_i.

(iii) The cost of A_i is no greater than the cost of B_i and the cost of B_i is no greater than the cost of $A_i \cup C_i \cup D_i$.

(iv) $|D_i| \leq 2|D_{i-1}|/3$.

Let $A_0 = B_0 = C_0 = \emptyset$, $D_0 = V$. Then (i)-(iv) hold. If A_{i-1}, B_{i-1}, C_{i-1}, D_{i-1} have been defined and $D_{i-1} \neq \emptyset$, let G^* be the subgraph of G induced by the vertex set D_{i-1}. Let A^*, B^*, C^* be a vertex partition satisfying Corollary 2 on G^*. Without loss of generality, suppose A^* has no more cost than B^*. Let A_i be the set among $A_{i-1} \cup A^*$, B_{i-1} with less cost, let B_i be the set among $A_{i-1} \cup A^*$, B_{i-1} with greater cost, let $C_i = C_{i-1} \cup C^*$, and let $D_i = B^*$. Then (i), (ii), (iii), and (iv) hold for A_i, B_i, C_i, D_i.

Let k be the largest index for which A_k, B_k, C_k, D_k are defined. Then $D_k = \emptyset$. Let $A = A_k$, $B = B_k$, $C = C_k$. By (i), A, B, C

partition V . By (ii), no edge joins a vertex in A with a vertex in B , By (iii), neither A nor B has cost exceeding 1/2 . By (iv), the total number of vertices in C is bounded by $\sum_{i=0}^{\infty} 2\sqrt{2}\sqrt{n}\ (2/3)^{i/2} = \frac{2\sqrt{2}\sqrt{n}}{1-\sqrt{2/3}}$. □

Another natural question is whether graphs which are "almost" planar have a $\sqrt{n}$ -separator theorem. The finite element method of numerical analysis gives rise to one interesting class of almost-planar graphs. We shall extend Theorem 4 to apply to such graphs.

A finite element graph is any graph formed from a planar embedding of a planar graph by adding all possible diagonals to each face. (The finite element graph has a clique corresponding to each face of the embedded planar graph.) The embedded planar graph is called the skeleton of the finite element graph and each of its faces is an element of the finite element graph.

Theorem 5. Let G be an n-vertex finite element graph with non-negative vertex costs summing to no more than one. Suppose no element of G has more than k boundary vertices. Then the vertices of G can be partitioned into three sets A, B, C such that no edge joins a vertex in A with a vertex in B , neither A nor B has total cost exceeding 2/3 , and C contains no more than $4\lfloor k/2 \rfloor \sqrt{n}$ vertices.

Proof. Let G* be the skeleton of G . Form G** from G* by inserting one new vertex into each face of G* containing four or more vertices and connecting the new vertex to each vertex on the boundary of the face. Then G** is planar. Apply Theorem 4 to G** . Let A**, B**, C** be the resulting vertex partition. This partition satisfies the theorem except that certain edges in G but not in G** may join A** and B** .

These edges are diagonals of certain faces of G^* ; call these bad faces. Each bad face must contain one of the new vertices added to G^* to form G^{**} , and this vertex must be in C^{**} .

Form G from C^{**} by deleting all new vertices and adding to G^{**} , for each bad face, either the set of vertices in A^{**} on the boundary of the bad face, or the set of vertices in B^{**} on the boundary of the bad face, whichever is smaller. Let A be theremainingold vertices in A^{**} and let B^{**} be the remaining old vertices in B^{**} . Then no edge in G joins A and B , neither A nor B contains more than $2n/3$ vertices, and C contains no more than $2\sqrt{2}\ \lfloor k/2 \rfloor \sqrt{n+a}$ vertices, where a is the number of faces of G^* containing four or more vertices. Using Euler's theorem, it is not hard to show that the number of faces of G^* containing four or more vertices is at most n-2 . Thus $|C| \leq \lfloor k/2 \rfloor \sqrt{n}$, and the theorem is true. □

Corollary 4. Let G be any n-vertex finite element graph. Suppose no element of G has more than k boundary vertices. The vertices of G can be partitioned into three sets A, B, C such that no edge joins a vertex in A with a vertex in B , neither A nor B contains more than $2n/3$ vertices, and C contains no more than $4\lfloor k/2 \rfloor \sqrt{n}$ vertices.

The last result of this section shows that Theorem 4 and its corollaries are tight to within a constant factor; that is, if $f(n) = o(\sqrt{n})$, no $f(n)$ -separator theorem holds for planar graphs.

Theorem 6. For any k, let $G = (V,E)$ be a $k \times k$ square grid graph (a $k \times k$ square section of the infinite grid graph in Figure 1). Let A be any subset of V such that $\alpha n < |A| \leq n/2$, where $n = k^2$ and α is a positive constant less than $1/2$. Then the number of vertices in $V-A$ adjacent to some vertex in A is at least $k \cdot \min\{1/2, \sqrt{\alpha}\}$.

Proof. Without loss of generality, suppose that the number r of rows of G which contain vertices in A is no less than the number c of columns of G which contain vertices in A. Then $\alpha n \leq |A| \leq rc \leq r^2$ and $r \geq \sqrt{\alpha}\, k$.

If r^* is the number of rows of G which contain only vertices in A, then $kr^* \leq |A| \leq n/2$, and $r^* \leq k/2$. If $r^* = 0$, then $|A| > r > \sqrt{\alpha}\, k$. If $r^* \neq 0$, then $r = k$ and $|A| > r - r^* = k - r^* \geq k/2$. □

It is an open problem to determine the smallest constant factor which can replace $2\sqrt{2}$ in Theorem 4.

3. An Algorithm for Finding a Good Partition.

The proof of Theorem 4 leads to an algorithm for finding a vertex partition satisfying the theorem. To make this algorithm efficient, we need a good representation of a planar embedding of a graph. For this purpose we use a list structure whose elements correspond to the edges of the graph. Stored with each edge are its endpoints and four pointers, designating the edges immediately clockwise and counter-clockwise around each of the endpoints of the edge. Stored with each vertex is some incident edge. Figure 5 gives an example of such a data structure.

[Figure 5]

Partitioning Algorithm.

Step 1: Find a planar embedding of G and construct a representation for it of the kind described above.

Time: $O(n)$, using the algorithm of [10].

Step 2: Find the connected components of G and determine the cost of each one. If none has cost exceeding 2/3, construct the partition as described in the proof of Theorem 4. If some component has cost exceeding 2/3, go to Step 3.

Time: $O(n)$ [9].

Step 3: Find a breadth-first spanning tree of the most costly component. Compute the level of each vertex and the number of vertices $L(\ell)$ in each level ℓ.

Time: $O(n)$.

Step 4: Find the level ℓ_1 such that the total cost of levels 0 through ℓ_1-1 does not exceed 1/2 , but the total cost of levels 0 through ℓ_1 does exceed 1/2 . Let k be the number of vertices in levels 0 through ℓ_1 .

Time: $O(n)$.

Step 5: Find the highest level $\ell_0 \leq \ell_1$ such that $L(\ell_0)+2(\ell_1-\ell_0) \leq 2\sqrt{k}$. Find the lowest level $\ell_2 \geq \ell_1+1$ such that $L(\ell_2)+2(\ell_2-\ell_1-1) \leq 2\sqrt{n-k}$.

Time: $O(n)$.

Step 6: Delete all vertices on level ℓ_2 and above. Construct a new vertex x to represent all vertices on levels 0 through ℓ_0 . Construct a Boolean table with one entry per vertex. Initialize to true the entry for each vertex on levels 0 through ℓ_0 and initialize to false the entry for each vertex on levels ℓ_0+1 through ℓ_2-1 . The vertices on levels 0 through ℓ_0 correspond to a subtree of the breadth-first spanning tree generated in Step 3. scan the edges incident to this tree clockwise around the tree. When scanning an edge (v,w) with V in the tree, check the table entry for w . If it is true, delete edge (v,w) . If it is false, change it to true, construct an edge (x,w) , and delete edge (v,w) . The result of this step is a planar representation of the shrunken graph to which Lemma 2 is to be applied. See Figure 6.

Time: $O(n)$.

[Figure 6]

Step 7: Construct a breadth-first spanning tree rooted at x in the new graph. (This can be done by modifying the breadth-first spanning tree constructed in Step 3.) Record, for each vertex v , the parent of v in the tree, and the total cost of all descendants of v including v itself. Make all faces of the new graph into triangles by scanning the boundary of each face and adding (non-tree) edges as necessary.

Time: $O(n)$.

Step 8: Choose any non-tree edge (v_1,w_1) . Locate the corresponding cycle by following parent pointers from v_1 and w_1 . Compute the cost on each side of this cycle by scanning the tree edges incident on either side of the cycle and summing their associated costs. If (v,w) is a tree edge with v on the cycle and w not on the cycle, the cost associated with (v,w) is the descendant cost of w if v is the parent of w , and the cost of all vertices minus the descendant cost of v if w is the parent of v . Determine which side of the cycle has greater cost and call it the "inside". See Figure 7.

Time: $O(n)$.

[Figure 7]

Step Let (v_i,w_i) be the non--tree edge whose cycle is the current candidate to complete the separator. If the cost inside the cycle exceeds 2/3 , find a better cycle by the following method.

Locate the triangle (v_i,y,w_i) which has (v_i,w_i) as a boundary edge and lies inside the (v_i, w_i) cycle. If either (v_i,y) or (y,w_i) is a tree edge, let (v_{i+1},w_{i+1}) be the non-tree edge among (v_i,y) and (y,w_i) . Compute the cost

inside the (v_{i+1}, w_{i+1}) cycle from the cost inside the (v_i, w_i) cycle and the cost of v_i, y, and w_i. See Figure 4.

If neither (v_i, y) nor (y, w_i) is a tree edge, determine the tree path from y to the (v_i, w_i) cycle by following parent pointers from y. Let z be the vertex on the (v_i, w_i) cycle reached during this search. Compute the total cost of all vertices except z on this tree path. Scan the tree edges inside the (y, w_i) cycle, alternately scanning an edge in one cycle and an edge in the other cycle. Stop scanning when all edges inside one of the cycles have been scanned. Compute the cost inside this cycle by summing the associated costs of all scanned edges. Use this cost, the cost inside the (v_i, w_i) cycle, and the cost on the tree path from y to z to compute the cost inside the other cycle. Let (v_{i+1}, w_{i+1}) be the edge among (v_i, y) and (y, w_i) whose cycle has more cost inside it.

Repeat Step 9 until finding a cycle whose inside has cost not exceeding 2/3.

Time: O(n) (see proof below).

Step 10: Use the cycle found in Step 9 and the levels found in Step 4 to construct a satisfactory vertex partition as described in the proof of Lemma 3. Extend this partition from the connected component chosen in Step 2 to the entire graph as described in the proof of Theorem 4.

Time: O(n).

This completes our presentation of the algorithm. All steps except Step 9 obviously run in O(n) time. We urge readers to fill in the details of this algorithm; we content ourselves here with proving that Step 9 requires O(n) time.

Proof of Step 9 Time Bound. Each iteration of Step 9 deletes at least one face from the inside of the current cycle. Thus Step 9 terminates after $O(n)$ iterations. The total running time of one iteration of Step 9 is $O(1)$ plus time proportional to the length of the tree path from y to z plus time proportional to the number of edges scanned inside the (v_i, y) and (y, w_i) cycles. Each vertex on the tree path from y to z (except z) is inside the current cycle but on the boundary or outside of all subsequent cycles. For every two edges scanned during an iteration of Step 9, at least one edge is inside the current cycle but outside all subsequent cycles. It follows that the total time spent traversing tree paths and scanning edges, during all iterations of Step 9, is $O(n)$. Thus the total time spent in Step 9 is $O(n)$. □

By making minor modifications to this algorithm, one can construct an $O(n)$ -time algorithm to find a vertex partition satisfying Theorem 5, and $O(n)$ -time algorithms to find vertex partitions satisfying Corollary 2 and Corollary 4.

4. Applications.

The separator theorem proved in Section 2 allows us to obtain many new complexity results since it opens the way for efficient application of divide-and-conquer on planar graphs. We mention a few such applications here; we shall present the details in a subsequent paper.

Generalized nested dissection. Any system of linear equations whose sparsity structure corresponds to a planar or finite element graph can be solved in $O(n^{3/2})$ time and $O(n \log n)$ space. This result generalizes the nested dissection method of George [5].

Pebbling. Any n-vertex planar acyclic directed graph with maximum in-degree k can be pebbled using $O(\sqrt{n} + k \log n)$ pebbles. See [8,16] for a description of the pebble game.

The Post Office Problem. Knuth's "post office" problem [11] can be solved in $O((\log n)^2)$ time and $O(n)$ space. See [3,17] for previous results.

Data Structure Embedding Problems. Any planar data structure can be efficiently embedded into a balanced binary tree. See [2,14] for a description of the problem and some related results.

Lower Bounds on Boolean Circuits. Any planar circuit for computing Boolean convolution contains at least cn^2 gates for some positive constant c.

Acknowledgments.

We would like to thank Stanley Eisenstat, Robert Floyd, Donald Rose, and Daniel Sleator for many helpful discussions and much thoughtful criticism.

Appendix: Graph-Theoretic Definitions

A graph $G = (V,E)$ consists of a set V of vertices and a set E of edges. Each edge is an unordered pair (v,w) of distinct vertices. If (v,w) is an edge, v and w are adjacent and (v,w) is incident to both v and w. A path of length k with endpoints v, w is a sequence of vertices $v = v_0, v_1, v_2, \ldots, v_k = w$ such that (v_{i-1}, v_i) is an edge for $1 \leq i \leq k$. If all the vertices $v_0, v_1, \ldots, v_{k-1}$ are distinct, the path is simple. If $v = w$, the path is a cycle. The distance from v to w is the length of the shortest path from v to w. (The distance is infinite if v and w are not joined by a path.) The level of a vertex v in a graph G with respect to a fixed root r is the distance from r to v.

If $G_1 = (V_1, E_2)$ and $G_2 = (V_2, E_2)$ are graphs, G_1 is a subgraph of G_2 if $V_1 \subseteq V_2$ and $E_1 \subseteq E_2$. G_1 is a generalized subgraph of G_2 if $V_1 \subseteq V_2$ and there is a mapping f from E_1 into the set of paths of G_2 such that, for each edge $(v,w) \in E_1$, $f((v,w))$ has endpoints v and v, and no two paths $f((v_1,w_1))$ and $f((v_2,w_2))$ share a vertex except possibly an endpoint of both paths. If $G = (V_1, E_1)$ is a graph and $V_1 \subseteq V_2$, the graph $G_1 = (V_1, E_1)$ where $E_1 = E_2 \cap \{(v,w) \mid v,w \in V_1\}$ is the subgraph of G_2 induced by the vertex set V_1. If $G_1 = (V_1, E_1)$ is a subgraph of $G_2 = (V_2, E_2)$, then shrinking G_1 to a single vertex in G_2 means forming a new graph G_2' from G_2 by deleting from G_2 all vertices in V_1 and all their incident edges, adding a new vertex x to G_2, and adding a new edge (x,w) to G_2 for each edge $(v,w) \in E_2$ such that $v \in V_1$ and $w \notin V_1$.

A graph is connected if any two vertices in it are joined by a path. The connected components of a graph are its maximal connected subgraphs. A clique is a graph such that any two vertices are joined by an edge. A tree is a connected graph containing no cycles. We shall generally assume that a tree has a distinguished vertex, called a root. If T is a tree with root r and v is on the (unique) simple path from r to w, V is an ancestor of w and w is a descendant of v. If in addition (v, w) is an edge of T, then v is the parent of w and W is a child of v. The radius of a tree is the maximum distance of any vertex from the root. A spanning tree T of a graph G is a subgraph of G which is a tree and which contains all the vertices of G. T is a breadth-first spanning tree with respect to a root r if, for any vertex v, the distance from r to v in T is equal to the distance from r to v in G.

A graph $G = (V, E)$ is planar if there is a one-to-one map f_1 from v into points in the plane and a map f_2 from E into simple curves in the plane such that, for each edge $(v,w) \in E$, $f_2((v,w))$ has endpoints $f_1(v)$ and $f_2(w)$, and no two curves $f_2((v_1,w_1))$, $f_2((v_2,w_2))$ share a point except possibly a common endpoint. Such a pair of maps f_1, f_2 is a planar embedding of G. The connected planar regions formed when the ranges of f_1 and f_2 are deleted from the plane are called the faces of the embedding. Each face is bounded by a curve corresponding to a cycle of G, called the boundary of the face. We shall sometimes not distinguish between a face and its boundary. A diagonal of a face is an edge (v,w) such that v and w are non-adjacent vertices on the boundary of the face.

References

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

[2] R. A. DeMillo, S. C. Eisenstat, and R. J. Lipton, "Preserving average proximity in arrays," Georgia Institut of Technology Technical Report (1976).

[3] D. Dobkin and R. J. Lipton, "Multi-dimensional searching problems," SIAM J. Comput. 5 (1976), 181-186.

[4] P. Erdös, R. L. Graham, and E. Szemerédi, "On sparse graphs with dense long paths," STAN-CS-75-504, Computer Science Dept., Stanford University (1975).

[5] J. A. George, "Nested dissection of a regular finite element mesh," SIAM J. Num. Anal. 10 (1973), 345-367.

[6] D. W. Hall and G. Spencer, Elementary Topology, Wiley, New York, 1955.

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

[8] J. Hopcroft, W. Paul, and L. Valiant, "On time versus space," Journal ACM 24 (1977),332-337.

[9] J. E. Hopcroft and R. E. Tarjan, "Efficient algorithms for graph manipulation," Comm. ACM 16 (1973), 372-378.

[10] J. E. Hopcroft and R. E. Tarjan, "Efficient planarity testing," Journal ACM 21 (1974), 549-568.

[11] D. E. Knuth, The Art of Computer Programming, Volume 3: Sorting and Searching, Addison-Wesley, Reading, Mass., 1973.

[12] C. Kuratowski, "Sur le probleme des corbes gauches en topologie," Fundamenta Mathematicae 15 (1930), 271-283.

[13] P. M. Lewis, R. E. Stearns, and J. Hartmanis, "Memory bounds for recognition of context-free and context-sensitive languages," IEEE Conference on Switching Theory and Logical Design (1965), 191-202.

[14] R. J. Lipton, S. C. Eisenstat, and R. A. DeMillo, "Space and time hierarchies for classes of control structures and data structures," Journal ACM 23 (1976),710-732.

[15] M. S. Paterson, "Tape bounds for time-bounded Turing machines," J. Comp. Sys. Sci. 6 (1972), 116-124.

[16] W. J. Paul, R. E. Tarjan, and J. R. Celoni, "Space bounds for a game on graphs," Math. Sys. Theory 10 (1977), 239-251.

[17] M. J. Shamos, "Problems in computational geometry," unpublished manuscript.

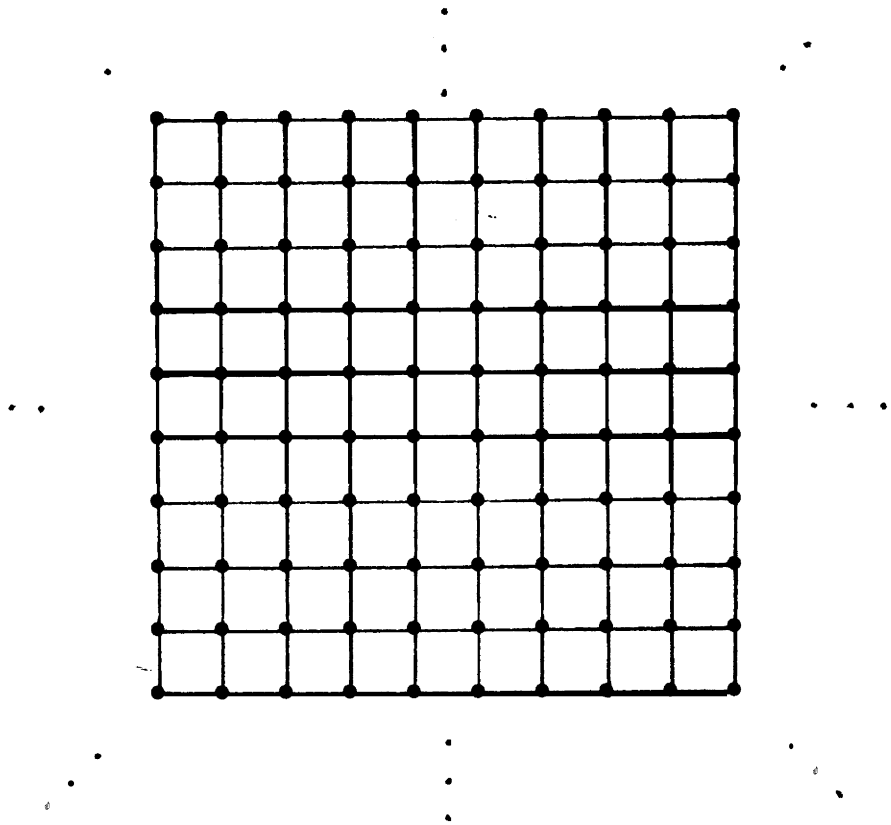


Figure 1. Infinite two-dimensional square grid.

(a) (b)

Figure 2. Kuratowski subgraphs.
(a) K_5 . (b) $K_{3,3}$.

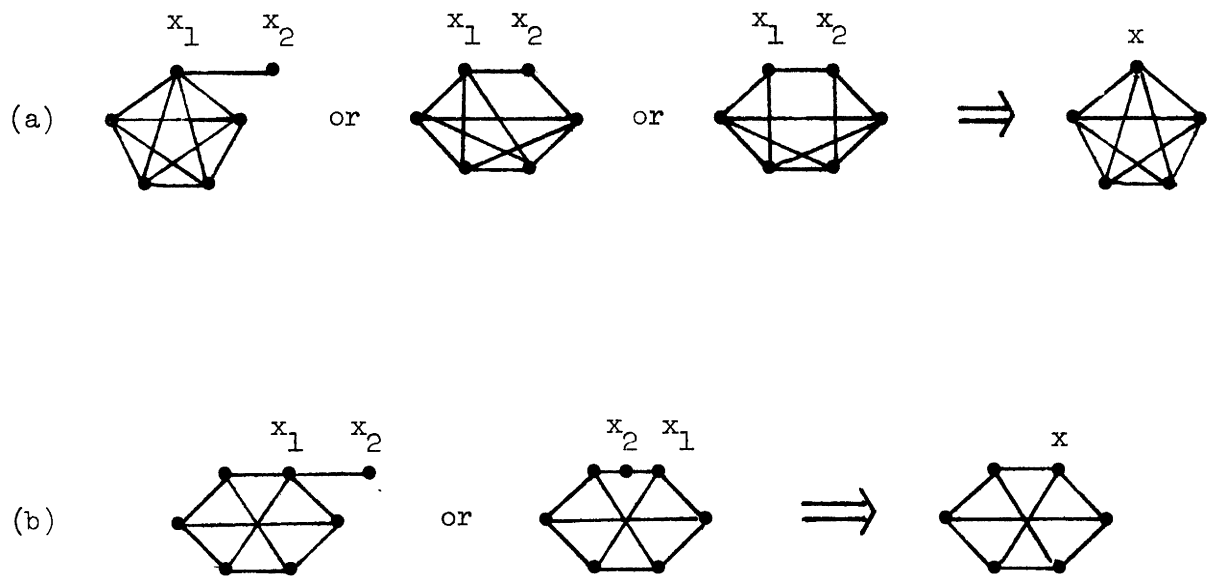


Figure 3. Shrinking an edge to form a Kuratowski graph. Original graph must contain a Kuratowski graph as a generalized subgraph.

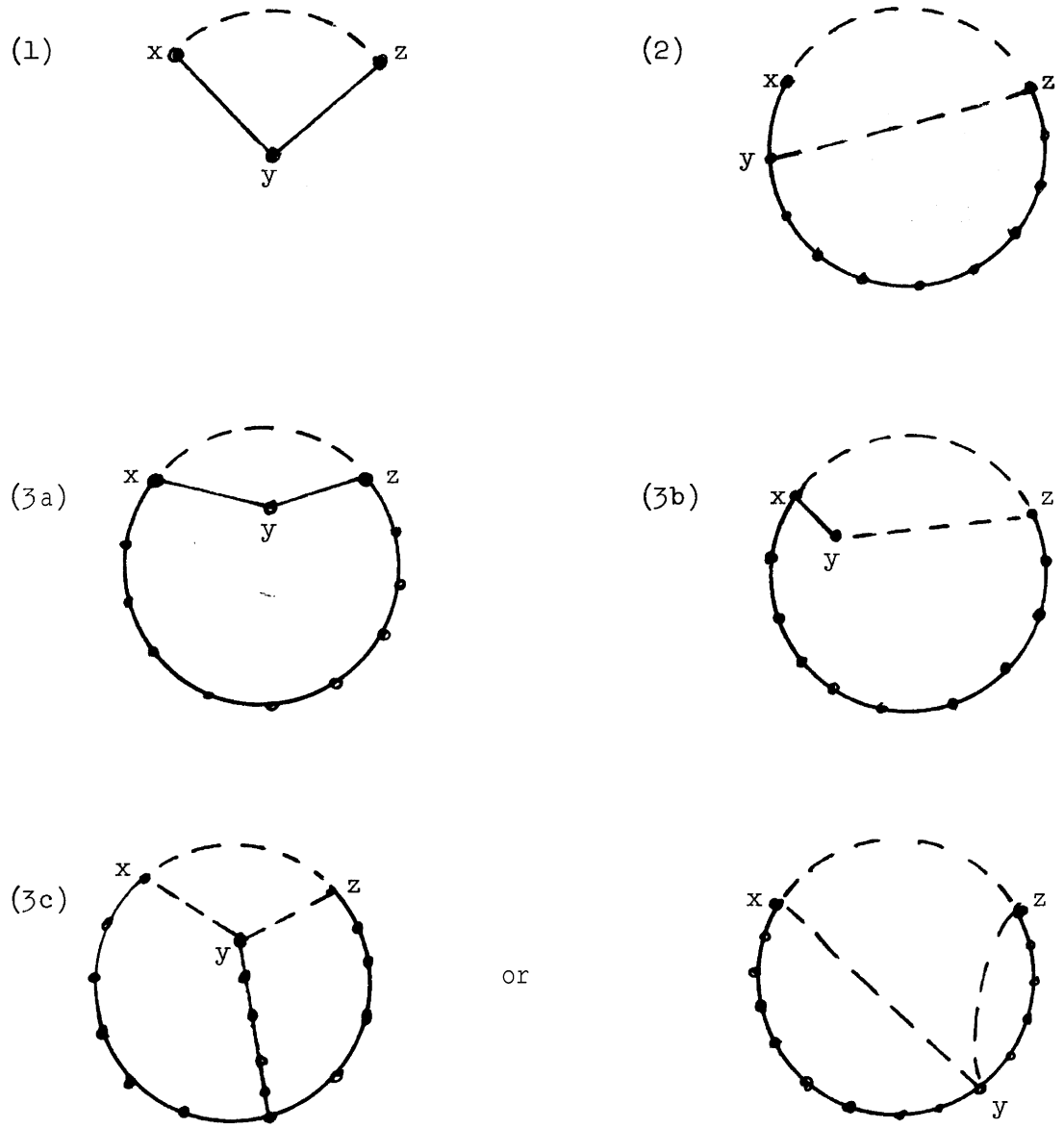
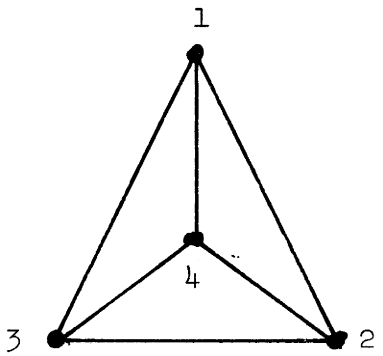


Figure 4. Cases for proof of Lemma 2, Solid edges are tree edges; dotted edges are non-tree edges.



Vertex incidences

1	e_1
2	e_1
3	e_2
4	e_3

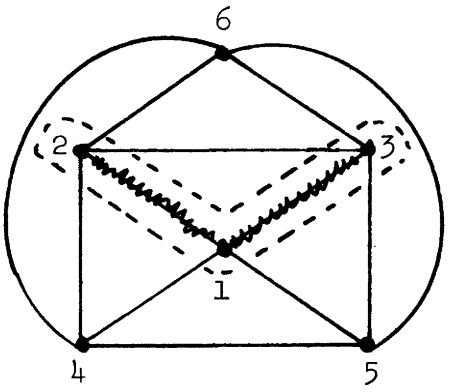
Edges and neighbors

			c_1	cc_1	c_2	cc_2
e_1	1	2	e_3	e_2	e_4	e_5
e_2	1	3	e_1	e_3	e_6	e_4
e_3	1	4	e_2	e_1	e_5	e_6
e_4	2	3	e_5	e_1	e_2	e_6
e_5	2	4	e_1	e_2	e_6	e_1
e_6	3	4	e_4	e_2	e_1	e_5

Figure 5. Representation of an embedded planar graph.

(c = clockwise, cc = counter-clockwise.)

(a)



(b) (1,4), ~~(2,4)~~, (2,6), ~~(2,3)~~, ~~(3,2)~~, ~~(3,6)~~, (3,5), ~~(1,5)~~

(c)

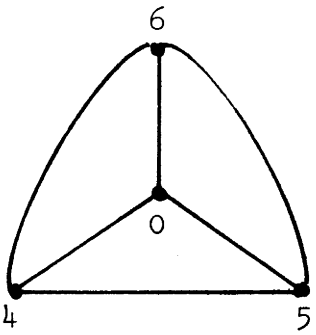


Figure 6. Shrinking a subtree of a planar graph.

(a) Original graph. Subtree denoted by ~~wwwwwwww~~ .

(b) Edges scanned around subtree. Those forming loops and multiple edges in shrunken graph are crossed out.

(c) Shrunken graph. Vertex 0 replaces sub-tree.

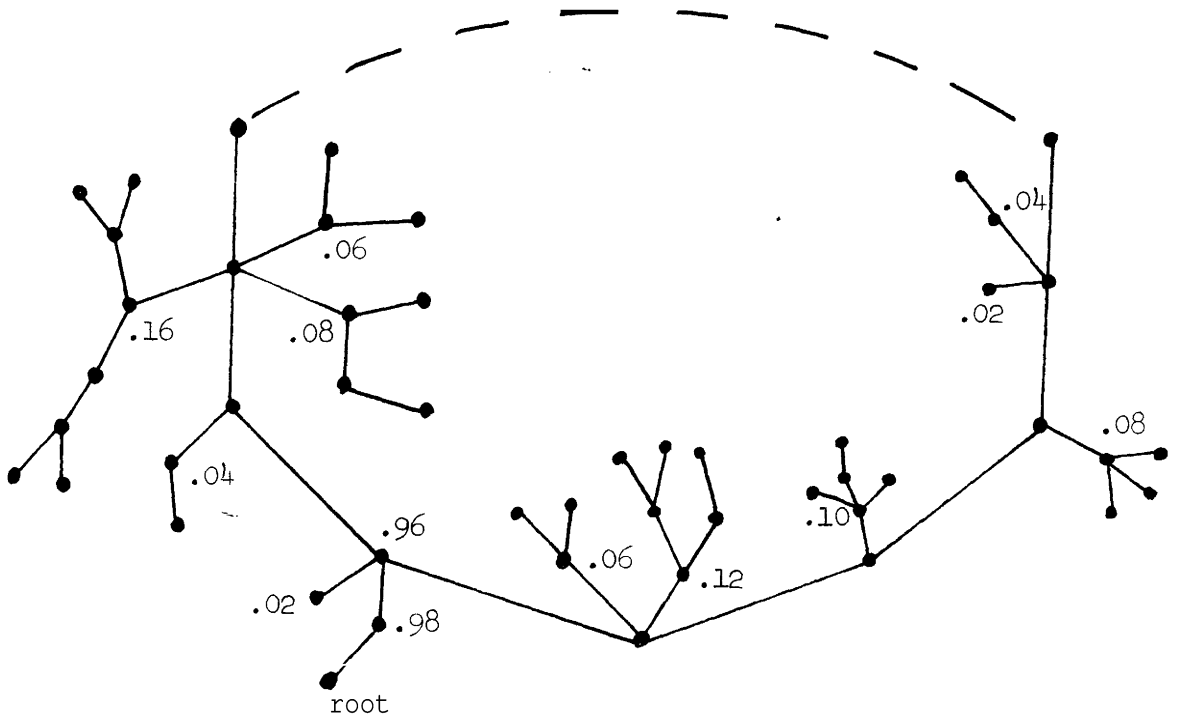


Figure 7. Cycle constructed in Step 8. All vertices have cost .02 . Numbers on vertices are descendant costs. The total cost inside the cycle is .48 , outside the cycle is .34 , and on the cycle is .18 .