MINIMUM s-t CUT OF A PLANAR UNDIRECTED NETWORK IN $O(n \log^2(n))$ TIME

JOHN H. REIF†

Abstract. Let N be a planar undirected network with distinguished vertices s, t, a total of n vertices, and each edge labeled with a positive real (the edge's cost) from a set L. This paper presents an algorithm for computing a minimum (cost) s-t cut of N. For general L, this algorithm runs in time $O(n \log^2(n))$. For the case when L contains only integers $\leqq n^{O(1)}$, the algorithm runs in time $O(n \log(n) \log\log(n))$. Our algorithm also constructs a minimum s-t cut of a planar graph (i.e., for the case $L = \{1\}$) in time $O(n \log(n))$. Our algorithm can also be used to compute a minimum cut for a general undirected planar network.

The fastest previous algorithm for computing a minimum s-t cut of a planar undirected network (Itai and Shiloach [SIAM J. Comput., 8 (1979), pp. 135–150]) has time $O(n^2 \log(n))$; the s-t cut is a byproduct of the maximum flow computed by their algorithm. The best previous time bound for minimum s-t cut of a planar graph (Cheston, Probert and Saxton [report, Dept. Computer Science, Univ. Saskatchewan, 1977]) was $O(n^2)$.

Key words. planar, network, minimum s-t cut, graph algorithm

1. Introduction. The importance of computing a minimum s-t cut of a network is illustrated by Ford and Fulkerson's [6], [7] theorem which states that the value of the minimum s-t flow of a network is precisely the minium s-t cut. The best known algorithm (Sleator [12] and Sleator and Tarjan [13]) for computing the maximum s-t flow or minimum s-t cut of a *sparse directed or undirected network* (with n vertices and $O(n)$ edges) has time[1] $O(n^2 \log(n))$. This paper is concerned with a *planar undirected network* N, which occurs in many practical applications.

Ford and Fulkerson [6], [7] have an elegant maximum s-t flow algorithm for the case N is *(s, t)-planar* (both s and t are on the same face) which when efficiently implemented by priority queues as described in Itai and Shiloach [9] has time $O(n \log(n))$. Moreover, $O(n)$ executions of their algorithm suffice to compute the maximum flow of a general *planar network* in total time $O(n^2 \log(n))$. Also, Cheston, Probert and Saxton [3] have an $O(n^2)$ algorithm for the minimum s-t cut of a planar graph and Shiloach [9] gives an $O(n \log(n))^2$ algorithm for the minimum cut of a planar graph.

Let $Q_L(n)$ be the asymptotic time complexity to maintain a priority queue of $O(n)$ elements with costs from a set L of nonnegative reals, and with $O(n)$ insertions and deletions. For the general case, $Q_L(n) = O(n \log(n))$ as described in Aho, Hopcroft and Ullman [1]. For the special case when L is a set of positive integers $\leqq n^{O(1)}$, Boas, Kaas and Zijlstra [2] show $Q_L(n) = O(n \log\log(n))$. It is obvious that if L is of constant cardinality then $Q_L(n) = O(n)$.

A key element of the Ford and Fulkerson [6], [7] algorithm for (s, t)-planar networks was an efficient reduction to finding a minimum cost path between two vertices in a sparse network. Dijkstra [4] gives an algorithm for a generalization of this problem (to find a minimum cost path from a fixed "source" vertex s to each other vertex). Dijkstra's algorithm may be implemented (see Aho, Hopcroft and

supported in part by the National Science Foundation under grant NSF-MCS79-21024 and the Office of Naval Research under contract N00014-80-C-0647.

† Aiken Computation Laboratory, Harvard University, Cambridge, Massachusetts 02138.

[1] We assume throughout this paper that our machine model is a unit cost criteria RAM (see Aho, Hopcroft and Ullman [1]).

Preprint of paper appearing in SIAM Journal on Computing, Vol. 12, No. 1, February 1983, pp. 71-81.

Ullman [1]) in time $O(Q_L(n))$ for a sparse network with n vertices, and L is the set of nonnegative reals labeling the edges.

Our algorithm for computing the minimum s-t cut of a planar undirected network has time $O(Q_L(n) \log (n))$. This algorithm also utilizes an efficient reduction to minimum cost path problems. Our fundamental innovation is a "divide and conquer" approach for cuts on the plane.

The paper is organized as follows: The next section gives preliminary definitions of graphs, networks, minimum cuts, maximum flows, and duals of planar networks. Section 3 gives the Ford–Fulkerson algorithm for (s, t)-planar graphs. Section 4 describes briefly an efficient algorithm due to Itai and Shiloach [9] for finding a minimum cut intersecting a given face of the primal network. Our divide and conquer approach is described and proved in § 5. Section 6 presents our algorithm for minimum s-t cuts of planar networks. Finally, § 7 concludes the paper.

2. Preliminary definitions

2.1. Graphs. Let a *graph* $G = (V, E)$ consist of a *vertex set* V and a collection of *edges* E. Each edge $e \in E$ *connects* two vertices $u, v \in V$ (edge e is a *loop* if it connects identical vertices). We let $e = \{u, v\}$ denote edge e connects u and v. Edges e, e' are *multiple* if they have the same endpoints. Let a *path* be a sequence of edges $p = e_1, \cdots, e_k$ such that $e_i = \{v_{i-1}, v_i\}$ for $i = 1, \cdots, k$ (we say p *traverses* vertices $v_0, \cdots, v_k$). Let p be a *cycle* if $v_0 = v_k$ (cycles containing the same edges are considered identical). A path p' is a *subpath* of p if p' is a subsequence of p. Let G be a *standard graph* if G has neither multiple edges nor loops and is triconnected. Generally we let $n = |V|$ be the number of vertices of graph G. If G is planar, then by Euler's formula G contains at most $6n - 12$ edges.

2.2. Networks. Let an *undirected network* $N = (G, c)$ consist of an undirected graph $G = (V, E)$ and a mapping c from E to the positive reals. For each edge $e \in v$, $c(e)$ is the *cost* of e. For any edge set $E' \subseteq E$, let $c(E') = \sum_{e \in E'} c(e)$. Let the *cost* of path $p = e_1, \cdots, e_k$ be $c(p) = \sum_{i=1}^{k} c(e_i)$. Let a path p from vertex u to vertex v be *minimum* if $c(p) \leqq c(p')$ for all paths p' from u to v. Let $N = (G, c, s, t)$ be a *standard network* if (G, c) is an undirected network, with $G = (V, E)$ a standard graph, and s, t are distinguished vertices of V (the *source*, *sink*, respectively). Note that triconnectivity can easily be achieved by adding $O(n)$ edges with cost 0.

2.3. Minimum cuts and maximum flows in networks. Let $N = (G, c, s, t)$ be a standard network with $G = (V, E)$. An edge set $X \subseteq E$ is an *s-t cut* if $(V, E - X)$ has no paths from s to t. Let s-t cut X be *minimum* if $c(X) \leqq c(X')$ for each s-t cut X'. See Fig. 1.

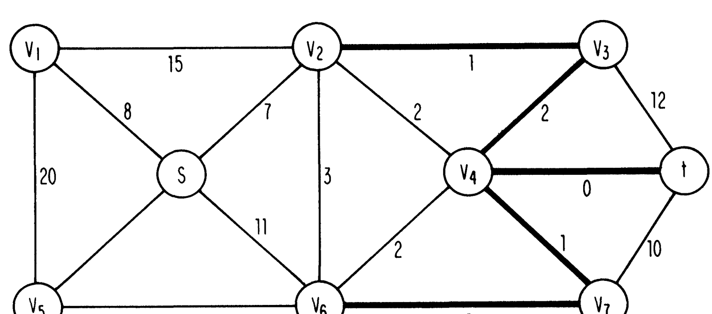

FIG. 1. *A network N with source s and sink t. The heavily drawn edges indicate a minimum s-t cut $\{\{v_2, v_3\}, \{v_3, v_4\}, \{v_4, t\}, \{v_4, v_7\}, \{v_6, v_7\}\}$ with cost 5.*

Let A be the set of directed edges $\{(u, v)|\{u, v\} \in E\}$. A function f mapping A to the nonnegative reals is a *flow* if

(i) For all $e \in A$, $f(e) \leqq c(e)$, and

(ii) For all $v \in V$, if $v \notin \{s, t\}$ then IN $(f, v) =$ OUT (f, v), where

$$\text{IN}(f, v) = \sum_{(u,v)\in A} f(u, v) \quad \text{and} \quad \text{OUT}(f, v) = \sum_{(v,u)\in A} f(v, u).$$

The *value* of the flow f is OUT $(f, s) -$ IN (f, t). The following motivates our work on minimum s-t cuts:

THEOREM 1 (Ford and Fulkerson [7]). *The maximum value of any flow is the cost of a minimum s-t cut.*

2.4. Planar networks and duals. Let $G = (V, E)$ be a planar standard graph, with a fixed embedding on the plane. G partitions the plane into connected regions. Each connected region is called a *face* and has a corresponding cycle of edges which it borders. For each edge $e \in E$, let $D(e)$ be the corresponding *dual edge* connecting the two faces bordering e. Let $D(G) = (\mathcal{F}, D(E))$ be the *dual graph* of G, with vertex set $\mathcal{F} =$ the faces of G, and with edge set $D(E) = \bigcup_{e\in E} D(e)$. Note that the dual graph is not necessarily standard (i.e., it may contain multiple edges and loops), but is planar. Let a cycle q of $D(G)$ be a *cut-cycle* if the region bounded by q contains exactly one of s or t. Note that a cycle is a cut-cycle independent of the way in which the dual graph is embedded in the plane, although a particular embedding may change which of s or t the cycle contains. See Figs. 1 and 2. The following proposition is trivial to derive:

PROPOSITION 1. *D induces a 1–1 correspondence between the s-t cuts of G and the cut-cycles of $D(G)$.*

Let $N = (G, c, s, t)$ be a *planar* standard network, with $G = (V, E)$ planar. Let the *dual network* $D(N) = (D(G), D(c))$ have edge costs $D(c)$, where the edge cost of each dual edge $D(e)$ is the cost of the original edge $e \in E$. (Generally we will use just c in place of $D(c)$ where no confusion will result.) See Fig. 3. For each face $F_i \in \mathcal{F}$, let a cut-cycle q in $D(N)$ be *F_i-minimum* if q contains F_i on (rather than inside) the cycle q and $c(q) \leqq c(q')$ for all cut-cycles q' containing F_i. The next proposition is easy but tedious to prove.

PROPOSITION 2. *A minimum s-t cut has the same cost as a minimum cost cut-cycle of $D(G)$.*

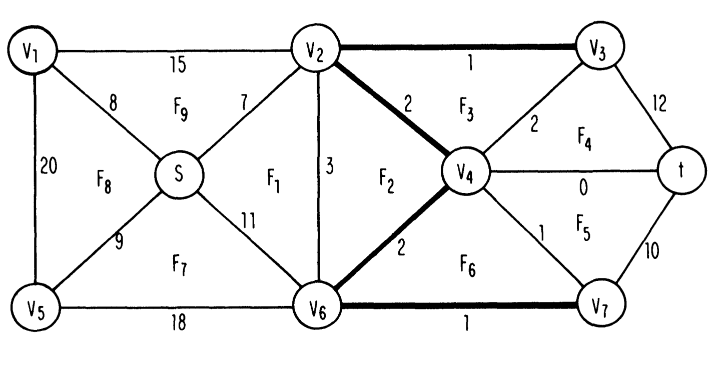

FIG. 2. *The same planar network N as in Fig. 1, with faces $F_1, \cdots, F_{10}$, and with a nonminimal s-t cut $X = \{\{v_2, v_3\}, \{v_2, v_4\}, \{v_4, v_6\}, \{v_6, v_7\}\}$ of cost 6, indicated by heavily drawn edges.*

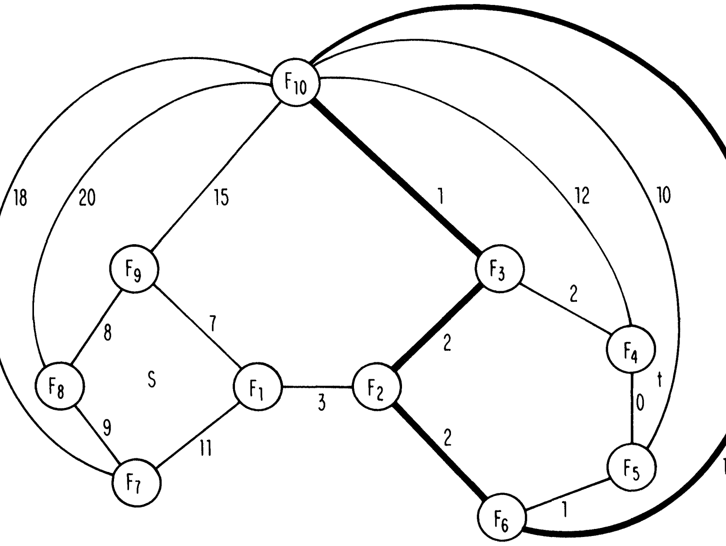

FIG. 3. *The dual network $D(N)$ derived from the planar network N of Figs.* 1 *and* 2. *The heavily drawn edges give an F_2-minimum cut cycle $D(X)=\{\{F_{10}, F_3\}, \{F_3, F_2\}, \{F_2, F_6\}, \{F_6, F_{10}\}\}$ which is the dual of the s-t cut X given in Fig.* 2.

3. Ford and Fulkerson's minimum s-t cut algorithm for (s,t)-planar networks. Let $N=(G,c,s,t)$ be a planar standard network. G (as well as N) is (s,t)-planar if there exists a face F_0 containing both s and t. Let planar network N' be derived from N by adding on edge e_0 connecting s and t with cost ∞. Let e_0 be embedded onto a line segment from s to t in F_0, which separates F_0 into two new faces F_1 and F_2. Ford and Fulkerson [6] have the following elegant characterization of the minimum s-t cut of (s,t)-planar network N.

THEOREM 2. *There is a* 1–1 *correspondence between the s-t cuts of N and the paths of $D(N')$ from F_2 to F_1 and avoiding $D(e_0)$. Furthermore, this correspondence preserves edge costs. Therefore, the minimum s-t cuts of N correspond to the minimum cost paths in $D(N')$ from F_2 to F_1.*

By use of Dijkstra's [4] shortest path algorithm, we have:

COROLLARY 2. *A minimum cut of (s,t)-planar network N with n vertices may be computed in time $O(Q_L)(n))$, where $L=$* range (c).

Note that applications of this corollary include the $O(n \log(n))$ time minimum s-t cut algorithm of Itai and Shiloach [9] for (s,t)-planar undirected networks, and the $O(n)$ time minimum s-t cut algorithm of Cheston, Probert and Saxton [3] for (s,t)-planar graphs.

4. An efficient algorithm for F-minimum cut cycles. Let $N=(G,c,s,t)$ be a planar standard network, with $G=(V,E)$ and $L=\text{range}(c)$. Our algorithm for minimum s-t cuts will require efficient construction of an F-minimum cut-cycle for a given face F. For completeness, we very briefly describe here an algorithm for this, due to Itai and Shiloach [9].

Let $\mathscr{F}_s$ be the set of faces bordering s and let $\mathscr{F}_t$ be the faces bordering t. Let a $\mu(s,t)$ *path* be a minimum cost path in $D(N)$ from a face of $\mathscr{F}_s$ to a face of $\mathscr{F}_t$.

PROPOSITION 3 (Itai and Shiloach [9]). *Let μ be a $\mu(s,t)$ path traversing faces $F_1,\cdots,F_d$. Let $D(X_i)$ be a F_i-minimum cut-cycle of $D(N)$ for $i=1,\cdots,d$. Then X_{i_0} is a minimum s-t cut of N, where $c(X_{i_0})=\min\{c(X_i)|i=1,\cdots,d\}$.*

To compute a $\mu(s, t)$ path in time $O(Q_L(n))$, let M be the planar network derived from $D(N)$ by adding new vertices v_s, v_t and an edge connecting v_s to each face in $\mathscr{F}_s$ and an edge connecting each face in $\mathscr{F}_t$ to v_t. Let the cost of each of these edges be 1. Let p be a minimum cost path in M from v_s to v_t. Then p, less its first and last edges, is a $\mu(s, t)$ path. See Fig. 4.

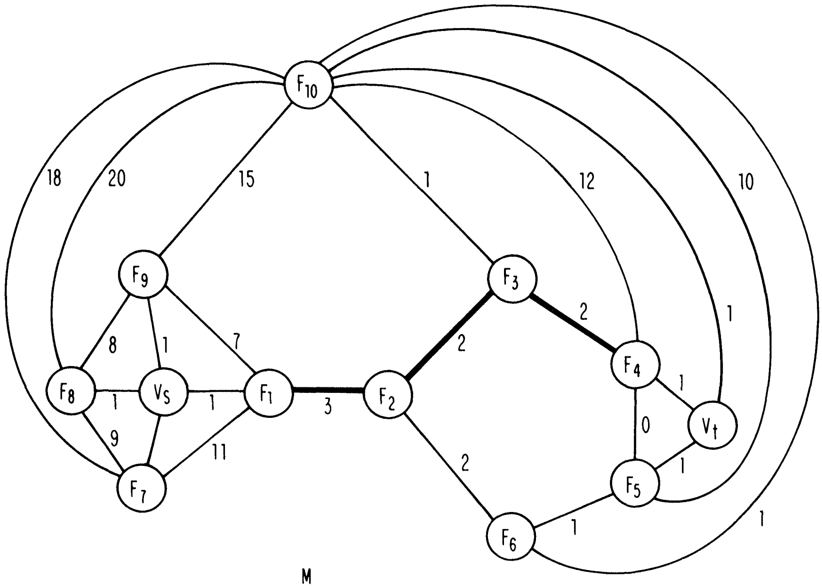

FIG. 4. *Network M derived from the dual network $D(N)$ given in Fig. 3. The heavily drawn edges are the $\mu(s, t)$-paths.*

Let μ be a $\mu(s, t)$ path in $D(N)$ traversing faces $F_1, \cdots, F_d$. By viewing μ as a horizontal line segment with s on the left and t on the right for each edge $D(e)$ of $D(N)$ which is not in $\mu(s, t)$ but is connected to a face F_i, $D(e)$ may be considered to be connected to F_i from *below* or *above* (or both). Let μ' be a copy of μ traversing new vertices $x_1, \cdots, x_d$. Let D' be the network derived from $D(N)$ by reconnecting to x_i each edge entering F_i from above. See Fig. 5. If p is a path of D', then a corresponding path $\hat{p}$ in $D(N)$ is constructed by replacing each edge and face appearing in μ' with the corresponding edge or face of μ. Clearly, $c(p) = c(\hat{p})$.

THEOREM 3 (Itai and Shiloach [9]). *If p is a minimum cost path connecting F_i and x_i in D', then $\hat{p}$ is an F_i-minimum cut-cycle of $D(N)$.*

By applying Corollary 2 to Theorem 3 we have:

COROLLARY 3. *This is an $O(Q_L(n))$ time algorithm to compute an F_i-minimum cut-cycle for any face F_i of a $\mu(s, t)$ path in $D(N)$.*
Note that for restricted L this may be more efficient than the $O(n \log n)$ upper bound given by Itai and Shiloach [9]; for example this gives an $O(n)$ time algorithm for an F_i-minimum cut-cycle of a planar graph.

5. A divide and conquer approach. Let μ be a $\mu(s, t)$ path of $D(N)$ traversing faces $F_1, \cdots, F_d$ as in § 4. Note that any s-t cut of planar network N must contain an edge bounding on a face in $\{F_1, \cdots, F_d\}$. The algorithm of Itai and Shiloach [9] for computing a minimum s-t cut of N is to construct an F_i-minimum cut-cycle $D(X_i)$ in $D(N)$ for each $i = 1, \cdots, d$. This may be done by $d = O(n)$ executions of the $O(Q_L(n))$ time algorithm of Corollary 3. Then by Proposition 3, X_{i_0} is a minimum s-t

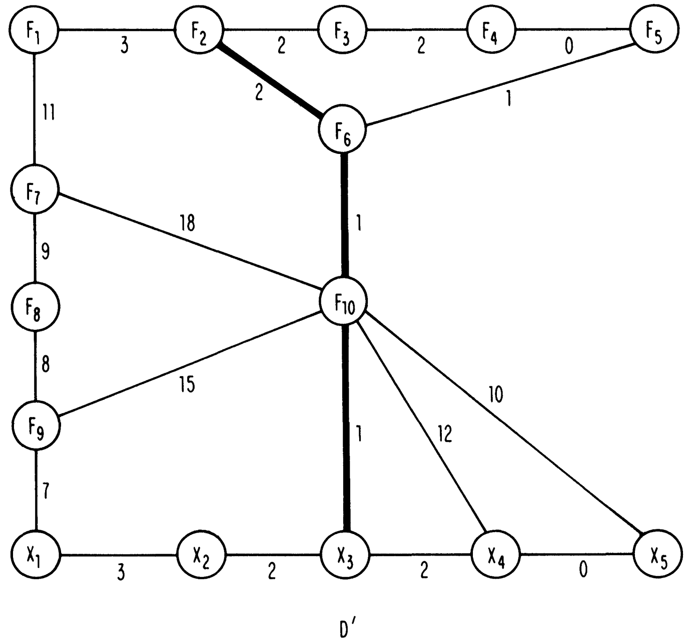

FIG. 5. *Network D' derived from dual network $D(N)$ of Fig. 3 using the $\mu(s, t)$-path of Fig. 4. The heavily drawn edges give the F_2-minimum cut-cycle $D(X)$ of Fig. 3.*

cut where $c(X_{1_0}) = \min\{c(X_1), \cdots, c(X_d)\}$. In the worst case, this requires $O(Q_L(n) \cdot n)$ total time. This section presents a divide and conquer approach which utilizes recursive executions of an F_i-minimum cut algorithm.

LEMMA 1. *Let F_i, F_j be distinct faces of μ, with $i < j$. Let p be any F_j-minimum cut-cycle of $D(N)$ such that the closed region R bounded by p contains s. Then there exists an F_i-minimum cut-cycle q contained entirely in R. (See Fig. 6.)*

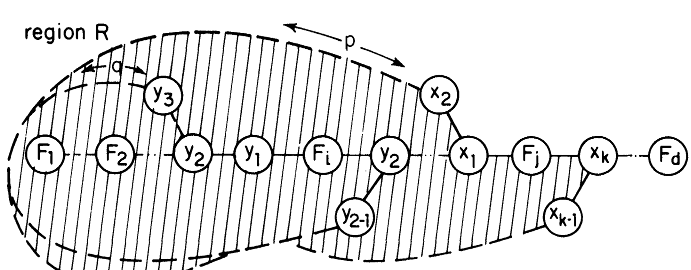

FIG. 6. *$F_1, F_2, \cdots, F_d$ is a $\mu(s, t)$ path in $D(N)$. $p = (F_j, x_1, x_2, \cdots, x_k)$ is a F_j-minimum cut-cycle enclosing region R. The F_i-minimum cut-cycle $q = (F_i, y_1, y_2, \cdots, y_l)$ is contained in R.*

Proof. Let q be any F_i-minimum cut-cycle. Let q' be the cut-cycle derived from q by repeatedly replacing subpaths of q connecting faces traversed by μ with the appropriate subpaths of μ (only apply replacements for which the resulting q' is a cut-cycle). Observe $c(q') \leqq c(q)$ (else we can show μ is not a $\mu(s, t)$ path). Let R' be the closed region bounded by q'. Suppose $R' \not\subset R$. Then there must be a subpath q_1 of q' connecting faces F^a, F^b of p such that q_1 only intersects R' at F^a and F^b. Let p_1 be the subpath of p connecting F^a and F^b in R'. We claim $c(p_1) \leqq c(q_1)$. Suppose $c(p_1) > c(q_1)$. By our construction of q', either q_1 avoids F_j, $F_j = F^a$ or $F_j = F^b$. In any case, we may derive a cut-cycle p' from p by substituting q_1 for p_1. But this implies $c(p')<c(p)$, contradicting our assumption that p is an F_i-minimum cut-cycle. Now substitute p_1 for q_1 in q'. The resulting cut-cycle is no more costly than q', since $c(p_1)\leqq C(q_1)$. See Fig. 7. The lemma follows by repeated application of this process. □

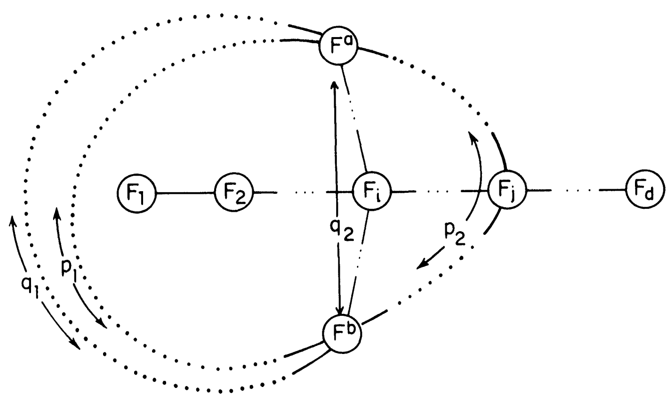

FIG. 7. *$F_1, F_2, \cdots, F_d$ is a $\mu(s, t)$-path, $p = p_1 \cdot p_2$ is a cut-cycle containing F_j. $q = q_1 \cdot q_2$ is a cut-cycle containing F_i. If $c(q_1)<c(p_1)$, then $p' = q_1 \cdot q_2$ is a cut-cycle containing F_i and with cost $c(p')<c(p)$.*

The above lemma implies a method for dividing the planar standard network N, given an s-t cut X. The network derived from N by deleting all edges of X can be partitioned into two networks N^s, N^t, where no vertex of N^s has a path to t, and no vertex of N^t has a path to s. Also, each edge $e \in X$ must have connections to a vertex of N^s and a vertex of N^t.

Let $N_0 = \text{DIVIDE}(N, X, s)$ be the standard planar network consisting of N^s,

(i) with a new vertex t_0 and
(ii) a new edge $\{u, t_0\}$ with cost $c(\{u, v\})$, for each edge $\{u, v\} \in X$ such that u is a vertex of N^s and v is a vertex of N^t;
(iii) finally (to insure N_0 is standard) merging multiple edges and setting the cost of each resulting edge to be the sum of the costs of the multiple edges from which it was derived. See Figs. 8 and 9.

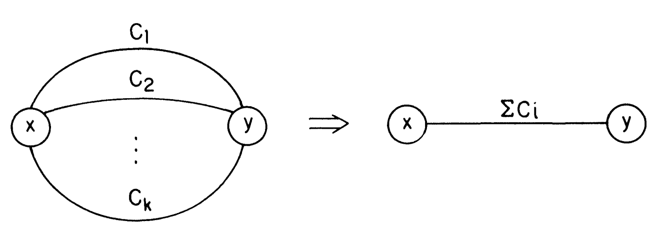

Fig. 8. *The merging into a single edge of multiple edges connected to vertex x and vertex y.*

Similarly, let $N_1 = \text{DIVIDE}(N, X, t)$ be the standard planar network consisting of N^t,

(i) with a new vertex s_1, and
(ii) for new edge $\{s_1, v\}$ with cost $c(\{u, v\})$, for each edge $\{u, v\} \in X$ such that u is a vertex of N^s and v is a vertex of N^t, and finally applying step (iii) above. See Fig. 9.

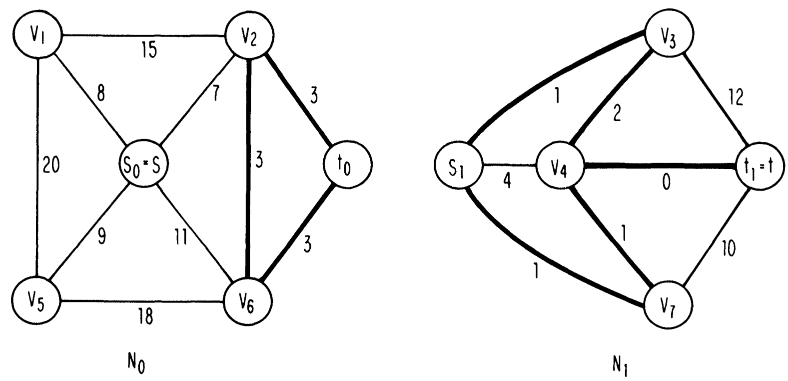

FIG. 9. *The networks* $N_0 = \text{DIVIDE}(N, X, s)$ *and* $N_1 = \text{DIVIDE}(N, X, t)$ *derived from the network N and s-t cut X given in Fig. 2.* N_0 *and* N_1 *will be further subdivided by the cuts* X_0, X_1 *respectively, indicated by heavily drawn edges.*

Let E be the set of edges of network N and let Y be a subset of the edges of $N_0 = \text{DIVIDE}(N, X, s)$ or of $N_1 = \text{DIVIDE}(N, X, t)$. Then let $E(Y)$ be the set of edges of E that were mapped into edges of Y when N_0 or N_1 was created. The next theorem follows immediately from the above Lemma 1 and Proposition 3.

THEOREM 4. *Let X be an s-t cut of a planar standard network N such that $D(X)$ is an F-minimum cut-cycle, for some face F in a $\mu(s, t)$ path of $D(N)$. Let X_0 be a minimum s-t_0 cut of* $N_0 = \text{DIVIDE}(N, X, s)$ *and let X_1 be a minimum s_1-t cut of* $N_1 = \text{DIVIDE}(N, X, t)$. *Then $E(X_0)$ or $E(X_1)$ is a minimum s-t cut of N.*

6. The minimum *s-t* cut algorithm for planar networks. Theorem 4 yields a very simple but efficient divide and conquer algorithm for computing minimum s-t cut of a planar standard network. We assume the Ford and Fulkerson [6] algorithm given in § 3:

(i) (s, t)-PLANAR-MIN-CUT(N) which computes a minimum s-t of (s, t)-planar standard network N in time $O(Q_L(n))$.

We also assume algorithms (given in § 4):

(ii) $\mu(s, t)$ PATH($D(N)$) computes a $\mu(s, t)$ path of $D(N)$ in time $O(Q_L(n))$.

(iii) F-MIN-CUT(N, F_i, μ) computes q, where $D(q)$ is an F_i-minimum cycle of N (for any F_i in $\mu(s, t)$ path μ), in time $O(Q_L(n))$.

RECURSIVE ALGORITHM PLANAR-MIN-CUT(N, μ).
input planar standard network $N = (G, c, s, t)$, where $G = (V, E)$, and $\mu(s, t)$ path μ.
begin
  Let $F_1, \cdots, F_d$ be the faces traversed by μ.
  if $d = 1$ **then return** (s, t)-PLANAR-MIN-CUT (N);
  else begin
    $X \leftarrow F$-MIN-CUT $(N, F_{\lfloor d/2 \rfloor}, \mu)$
    $N_0 \leftarrow$ DIVIDE (N, X, s); $N_1 \leftarrow$ DIVIDE (N, X, t);
    Let μ_0 and μ_1 be the subpaths of μ contained in N_0
    and N_1, respectively
    $X_1 \leftarrow$ PLANAR-MIN-CUT (N_1, μ_1); $X_0 \leftarrow$ PLANAR-MIN-CUT (N_0, μ_0)
    if $c(E(X_0)) \leqq c(E(X_1))$ **then return** $E(X_0)$ **else return** $E(X_1)$;
    end;
  end

Associated with this recursive algorithm we define a *call tree* T whose root is N and whose descendants are the networks input to the algorithm on recursive calls. Let d be the number of faces traversed by μ, the $\mu(s, t)$ path of N. If $d=1$ then root N has no children. Otherwise, N has left child N_0 and right child N_1, as computed in the algorithm, and so on.

For any $\omega \in \{0, 1\}^*$ inductively let $N_\omega = (G_\omega, c_\omega, s_\omega, t_\omega)$ be the planar standard network and let μ_ω be the $\mu(s_\omega, t_\omega)$ path in N_ω defined by some recursive calls to PLANAR-MIN-CUT. Suppose PLANAR-MIN-CUT (N_ω, μ_ω) is called. If μ_ω contains only one face, then let $N_{\omega 0}$ and $N_{\omega 1}$ be empty networks, and let $\mu_{\omega 0}$ and $\mu_{\omega 1}$ be empty paths. Else let X_ω be the set s_ω-t_ω cut of N_ω computed by the call to F-MIN-CUT$(\cdot)$, let $N_{\omega 0}$, $N_{\omega 1}$ be the planar standard networks constructed by the calls to DIVIDE, and let $\mu_{\omega 0}$, $\mu_{\omega 1}$ be the subsets of μ contained in $N_{\omega 0}$, $N_{\omega 1}$. Then it is easy to verify that $\mu_{\omega 0}$ is a $\mu(s_{\omega 0}, t_{\omega 0})$ path in $N_{\omega 0}$ and $\mu_{\omega 1}$ is a $\mu(s_{\omega 1}, t_{\omega 1})$ path in $N_{\omega 1}$, and the length of $\mu_{\omega 0}$ and the length of $\mu_{\omega 1}$ are each $\leqq \lceil \frac{1}{2} d_\omega \rceil$, where d_ω is the length of μ_ω. Hence there can be no more than $\lceil \log(d) \rceil$ mutually recursive calls, so the call tree T has depth at most $\lceil \log(d) \rceil \leqq \lceil \log(n) \rceil$, where n is the number of nodes in N.

Let m be the number of edges of N and let m_ω be the number of edges of N_ω. The following theorem provides an upper bound of $2m+2^r$ on the number of edges of networks of depth r in the call tree T.

THEOREM 5. *For each* $r \geqq 0$, $\sum_{\omega \in \{0,1\}^r} m_\omega \leqq 2m + 2^r$.

Proof. Note that by definition of DIVIDE, the edges of $N_{\omega 0}$ or $N_{\omega 1}$ are derived from disjoint sets of edges of N_ω. Fix an edge e of N. Let e_ω be the edge (if it exists) of N_ω derived from a set of edges of N containing e. Let edge e *contribute* to N_ω if $e \neq \{s_\omega, t_\omega\}$ and let e *fully contribute* to N_ω if e_ω contains neither s_ω nor t_ω. For each $r \geqq 0$, let $B_r(e) = \{e_\omega | e_\omega \neq \{s_\omega, t_\omega\}$ and $\omega \in \{0, 1\}^r\}$. Thus $|B_r(e)|$ is the number of networks of depth r in T to which edge e contributes.

Let the strings of $\{0, 1\}^*$ be ordered lexicographically. We require a technical lemma.

LEMMA 2. $|B_r(e)| \leqq 2$, *and furthermore if* $B_r(e) = \{e_\omega, e_z\}$ *for* $\omega < z$, $z \in \{0, 1\}^r$, *then edge* e_ω *is connected to* t_ω *and edge* e_z *is connected to* s_z.

This lemma states that e contributes to at most two networks of depth r in T, and e fully contributes to no two distinct networks of depth r. For example, consider edge $e = \{v_2, v_3\}$ of network N given in Fig. 2. Edge e fully contributes to N. In Fig. 9, edge e contributes to N_0 by edge $e_0 = \{v_2, t_0\}$ and also contributes to N_1 by edge $e_1 = \{s_1, v_3\}$. Furthermore, in Fig. 10 edge e contributes to N_{00} by edge $e_{00} = \{v_2, t_{00}\}$ and in Fig. 11 edge e contributes to N_{11} by edge $e_{11} = \{s_{11}, v_3\}$ but e contributes to neither N_{01} nor N_{10}.

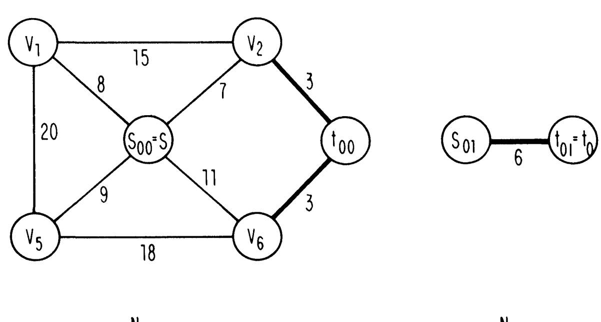

FIG. 10. *Networks* N_{00} = DIVIDE (N_0, X_0, s_0) *and* N_{01} = DIVIDE (N_0, X_0, t_0) *derived from network* N_0 *with* s-t_0 *cut* X_0 *of Fig.* 9.

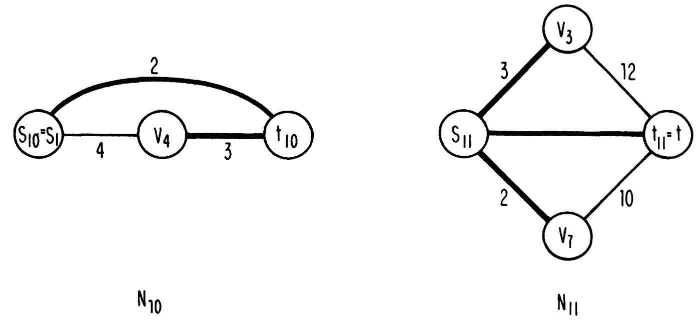

FIG. 11. *Networks* $N_{10} = \text{DIVIDE}(N_1, X_1, s_1)$ *and* $N_{11} = \text{DIVIDE}(N_1, X_1, t_1)$ *derived from network* N_1 *with* s_1-t *cut* X_1 *of Fig.* 9.

Proof of Lemma 2 *by induction.* Suppose for some fixed r_0, this lemma holds for all $r \leqq r_0$. If $B_{r_0}(e) = \varnothing$ then clearly $B_{r_0+1}(e) = \varnothing$. Suppose $1 \leqq |B_{r_0}(e)| \leqq 2$ and consider any $e_\omega \in B_{r_0}(e)$. If $e_\omega \notin X_\omega$ then by definition of DIVIDE, either $e_\omega = e_{\omega 0}$ appears in $N_{\omega 0}$ or $e_\omega = e_{\omega 1}$ appears in $N_{\omega 1}$, but not both. On the other hand, if $e_\omega \in X_\omega$, then $e_{\omega 0}$ appears in $N_{\omega 0}$ connected to $t_{\omega 0}$ and also $e_{\omega 1}$ appears in $N_{\omega 1}$ connected to $s_{\omega 1}$. In either case, if $|B_{r_0}(e)| = 1$, then $|B_{r_0+1}(e)| \leqq 2$. Otherwise suppose $|B_{r_0}(e)| = 2$ so there exists some $e_z \in B_{r_0}(e)$ with $\omega < z$. By the induction hypothesis, e_ω is connected to t_ω and e_z is connected to s_z. Thus for $j = 0, 1$ edge $e_{\omega j}$ (if it exists) is connected to $t_{\omega j}$ and edge e_{zj} (if it exists) is connected to s_{zj}. Hence if $e_\omega \in X_\omega$ then $e_{z1} = \{s_{z1}, t_{z1}\}$. In each case, $|B_{r_0+1}(e)| \leqq 2$. □

To complete the proof of Theorem 5, observe that $|\{\{s_\omega, t_\omega\} | \omega \in \{0, 1\}^r\}| = 2^r$. Hence

$$\sum_{\omega \in \{0,1\}^r} m_\omega \leqq \left(\sum_{e \in E} |B_r(e)| \right) + |\{\{s_\omega, t_\omega\} | \omega \in \{0, 1\}^r\}| \leqq 2m + 2^r$$

by Lemma 2. □

THEOREM 6. *Given a planar standard network* $N = (G, c, s, t)$ *with* $L = \text{range}(c)$, *and* μ *is a* $\mu(s, t)$ *path of* N *then* PLANAR-MIN-CUT (N, μ) *computes a minimum* s-t *cut of* N *in time* $O(Q_L(n) \log(n))$.

Proof. The total time cost is

$$\sum_{\substack{\omega \in \{0,1\}^r \\ 0 \leqq r \leqq \lceil \log(n) \rceil}} O(Q_L(m_\omega)) = \sum_{0 \leqq r \leqq \lceil \log(n) \rceil} O(Q_L(2m + 2^r)) \quad \text{by Theorem 5,}$$

$$= O(Q_L(n) \log(n)) \quad \text{since } 2m + 2^{\log(n)} = O(n). \qquad \square$$

By known upper bounds on the cost of maintaining queues (as discussed in the Introduction), we also have:

COROLLARY 4. *A minimum* s-t *cut of* N *is computed in time* $O(n \log^2(n))$ *for general* L *(i.e., a set of positive reals), in time* $O(n \log(n) \log\log(n))$ *for the case where* L *is a set of positive integers bounded by a polynomial in* n *and in time* $O(n \log(n))$ *for the case where* N *is a graph with identically weighted edges.*

7. Conclusion. We have presented a divide and conquer method for computing a minimum s-t cut of a planar undirected network which improves on the running time of the algorithm of Itai and Shiloach [9] by a factor of $n/\log n$. An additional attractive feature of this algorithm is its *simplicity*, as compared to other algorithms for computing minimum s-t cuts for sparse networks (Galil and Naamad [8], Shiloach [10] and Sleator and Tarjan [13]).

[1] A. AHO, J. HOPCROFT AND J. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] P. VAN EMDE BOAS, R. KAAS AND E. ZIJLSTRA, *Design and implementation of an efficient priority queue*, Math. Systems Theory, 10 (1977), pp. 99–127.

[3] G. CHESTON, R. PROBERT AND C. SAXTON, *Fast algorithms for determination of connectivity sets for planar graphs*, Dept. Computer Science, Univ. Saskatchewan, 1977.

[4] E. DIJKSTRA, *A note on two problems in connections with graphs*, Numer. Math., 1 (1959), pp. 269–271.

[5] S. EVEN AND R. TARJAN, *Network flow and testing graph connectivity*, this Journal, 4 (1975), pp. 507–518.

[6] C. FORD AND D. FULKERSON, *Maximal flow through a network*, Canad. J. Math., 8 (1956), pp. 399–404.

[7] ———, *Flows in Networks*, Princeton Univ. Press, Princeton, NJ, 1962.

[8] Z. GALIL AND A. NAAMAD, *Network flow and generalized path compression*, in Proc. of Symposium on Theory of Computing, Atlanta, Georgia, 1979.

[9] A. ITAI AND Y. SHILOACH, *Maximum flow in planar networks*, this Journal, 8 (1979), pp. 135–150.

[10] Y. SHILOACH, *An $O(nI \cdot \log^2 I)$ maximum-flow algorithm*, Computer Science Dept., Stanford Univ., Stanford, CA, 1978.

[11] ———, *A multi-terminal minimum cut algorithm for planar graphs*, this Journal, 9 (1980), pp. 214–219.

[12] D. SLEATOR, *An $O(nm \log n)$ algorithm for maximum network flow*, Ph.D. dissertation, Stanford Univ., Stanford, CA, 1980.

[13] D. SLEATOR AND R. TARJAN, *A data structure for dynamic trees*, 13th Annual ACM Symposium on Theory of Computing, 1981, pp. 114–122.