MEMORANDUM
RM-5383-PR
JULY 1967

MAXIMIZING FLOW THROUGH A NETWORK WITH NODE AND ARC CAPACITIES

R. D. Wollmer

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

MEMORANDUM
RM-5383-PR
JULY 1967

MAXIMIZING FLOW THROUGH A NETWORK WITH NODE AND ARC CAPACITIES

R. D. Wollmer

This research is supported by the United States Air Force under Project RAND—Contract No. F44620-67-C-0045—monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of the United States Air Force.

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA • 90406

PREFACE

This Memorandum presents an algorithm for maximizing the flow from one node to another through a network whose arcs and nodes have limited capacities. It is a product of RAND and Air Force interest in network flows and is currently being used in an interdiction study which is the subject of a forthcoming RAND Memorandum. If the reader is interested in an extensive description of the foundations of network theory, he should read L. R. Ford and D. R. Fulkerson, Flows in Networks, The RAND Corporation, R-375-PR, December 1960.

This study should be useful to persons in the Operations Analysis Office, Hq. USAF (AFGOA), the Intratheater Transportation Study Group, and others who deal with interdiction and transportation problems.

SUMMARY

In many actual network flow situations, nodes as well as arcs have limited capacities. This Memorandum presents for such a network, an algorithm for maximizing flow from a source node to a sink node. The algorithm allows us to treat these situations without introducing artificial arcs and nodes, as has been done in the past. Eliminating the artificial arcs and nodes simplifies network analysis since it always results in half as many nodes, as well as less than half as many arcs if the original arcs are undirected.

In addition, the following generalization of Ford and Fulkerson's max-flow, min-cut theorem is presented and proven. (See L. R. Ford and D. R. Fulkerson, Flows in Networks, The RAND Corporation, R-375-PR, December 1960.) Consider two subsets of the nodes, X and Y, whose union is the set of all network nodes and such that the source node is a member of X, and the sink node is a member of Y. Then, forming a cut set separating the source and sink are the nodes in the intersection of X and Y, and the set of all arcs (i, j), such that i is a member of X - Y, and j is a member of Y - X. Letting a cut set's value be the sum of the capacities of all its arcs and nodes, it follows that the maximum flow is equal to the minimum value of all cut sets separating the source and sink.

CONTENTS

I. INTRODUCTION

The problem dealt with in this Memorandum is one of maximizing the flow through a network in which the nodes and arcs have limited capacity. Specifically, a network consisting of arcs and nodes is given. The nodes are points where two or more arcs meet, and the arcs are line segments connecting two nodes. Each arc has a positive capacity representing the maximum amount of flow that may pass across it, and each node has a positive capacity representing the maximum amount of flow that may enter or leave it. If the network is directed, each arc also has a direction indicating the direction along which flow is allowed to pass. For undirected networks, flow may pass in either direction on an arc. Figure 1, page 5, shows an example of a directed network where it is desired to find the maximum amount that may pass from a node known as the source to one called the sink, subject to the arc and node capacity constraints.

A previous study of this problem split each node into two nodes and an arc.* In the case of undirected networks, this method also requires replacing each arc by two directed arcs. The algorithm presented here should allow substantial savings in computer time and core storage for undirected networks and may also allow good savings for directed networks.

Section II describes the linear program that maximizes flow through the network. Section III illustrates how cut sets are used to solve the problem, and Sec. IV gives the algorithm for finding a maximum flow

*L. R. Ford and D. R. Fulkerson, Flows in Networks, The RAND Corporation, R-375-PR, December 1960.

pattern. Section V contains proofs that the linear program's constraints (conservation at the nodes, arc capacities, and node capacities) hold. Section VI shows how the maximum flow algorithm terminates and that the maximum flow equals the minimum cut, while Sec. VII demonstrates the algorithm with an example.

II. LINEAR PROGRAM FORUMULATION

The problem of maximizing flow, v, through a network with node and arc capacities can be formulated as a linear program. Let c(i) be the capacity of node i; c(i, j) the capacity of arc (i, j)*; x(i, j) the flow on arc (i, j); A(i) the set of nodes to which arcs are directed from node i; B(i) the set of nodes from which arcs are directed to i; S the source node; and $\bar{S}$ the sink. The linear programming formulation of the network problem is as follows.

Maximize v subject to:

$$v,\ x(i,j) \geq 0 \quad \text{all } i,j$$

$$\text{I} \qquad x(i,j) \leq c(i,j) \quad \text{all } (i,j)$$

$$\text{II a)} \quad \sum_{j \in A(i)} x(i,j) \leq c(i) \quad i \neq \bar{S}$$

$$\text{b)} \quad \sum_{j \in B(\bar{S})} x(j,\bar{S}) \leq c(\bar{S})$$

$$\text{III a)} \quad \sum_{j \in A(S)} x(S,j) - \sum_{j \in B(S)} x(j,S) = v$$

$$\text{b)} \quad \sum_{j \in A(i)} x(i,j) - \sum_{j \in B(i)} x(j,i) = 0 \quad i \neq S, \bar{S}$$

$$\text{c)} \quad \sum_{j \in A(\bar{S})} x(\bar{S},j) - \sum_{j \in B(\bar{S})} x(j,\bar{S}) = -v \quad .$$

Relationship I represents the arc capacities and II the node capacities. Note that for all nodes other than the source and sink,

*For undirected networks, c(i, j) = c(j, i) for all (i, j).

the node capacity may be expressed either in terms of the flow into or the flow out of the node since these two quantities are equal. For the source and sink, these two quantities are unequal and the node capacity must be expressed in terms of the greatest of these two quantities. Thus the source node capacity limits flow out of the source while the sink capacity limits flow into the sink. Relationships III are balance equations showing that the net flow out of the source and the net flow into the sink both equal the value of the flow and that the net flow into or out of any other node is zero. Note that any one of the balance equations may be dropped since they sum up identically to zero. Physically, this dependency expresses the fact that every unit of arc flow enters a node and leaves a node. For the network of Fig. 1, the linear program would be, maximize v subject to:

$$v,\ x(i,j) \geq 0 \qquad \text{all } i,j$$

$$x(S,a) \leq 5$$

$$x(S,b) \leq 1$$

$$x(S,c) \leq 6$$

$$x(a,b) \leq 3$$

$$x(a,\bar{S}) \leq 5$$

$$x(b,c) \leq 4$$

$$x(b,\bar{S}) \leq 7$$

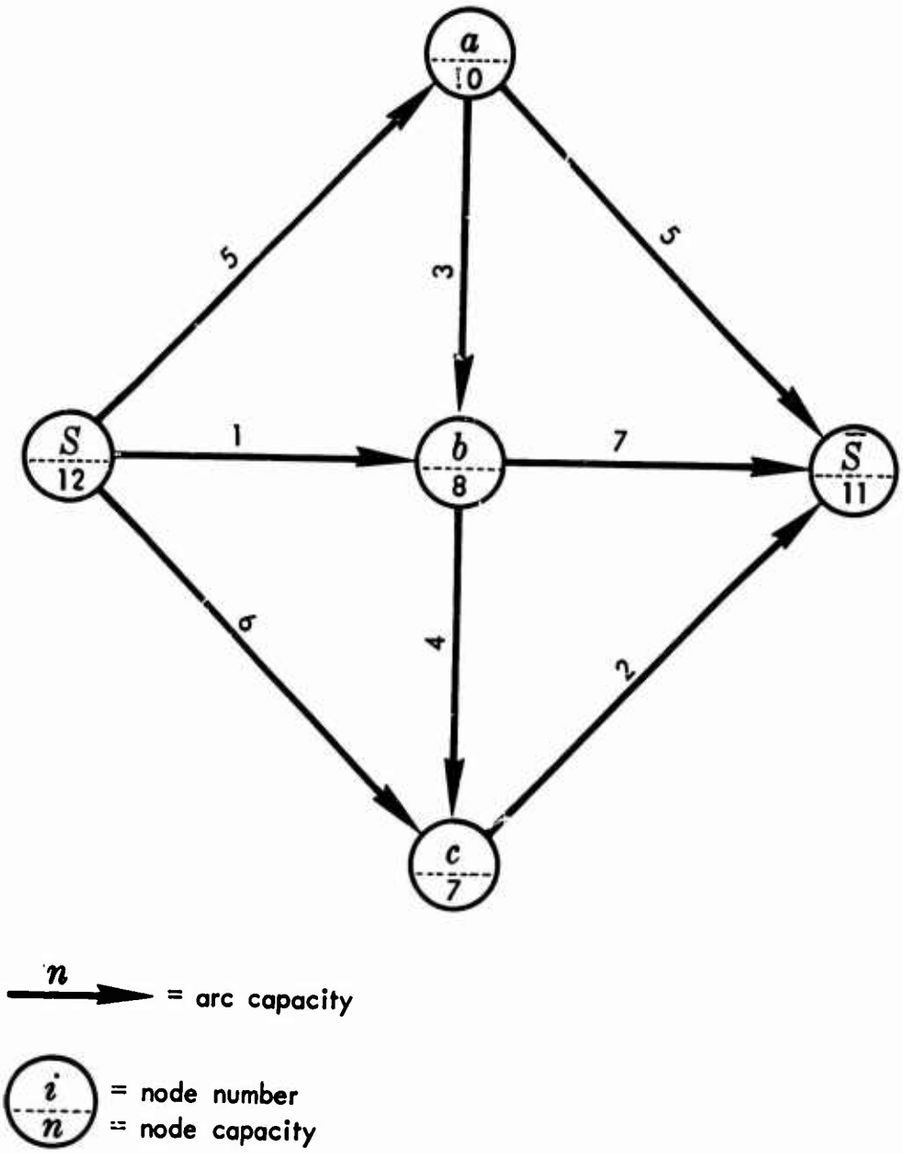


Fig. 1 -- Example of a network

$$x(c, \bar{S}) \leq 2$$

$$x(S, a) + x(S, b) + x(S, c) \leq 12$$

$$x(a, b) + x(a, \bar{S}) \leq 10$$

$$x(b, c) + x(b, \bar{S}) \leq 8$$

$$x(c, \bar{S}) \leq 7$$

$$x(a, \bar{S}) + x(b, \bar{S}) + x(c, \bar{S}) \leq 11$$

$$x(S, a) + x(S, b) + x(S, c) - v = 0$$

$$x(a, b) + x(a, \bar{S}) - x(S, a) = 0$$

$$x(b, c) + x(b, \bar{S}) - x(S, b) - x(a, b) = 0$$

$$x(c, \bar{S}) - x(S, c) - x(b, c) = 0$$

$$-x(a, \bar{S}) - x(b, \bar{S}) - x(c, \bar{S}) + v = 0 \quad .$$

A set of nodes $a_1, \ldots, a_n$ is called a chain from a_1 to a_n if either (a_i, a_{i+1}) or (a_{i+1}, a_i) is an arc for $i = 1, \ldots, n - 1$. If, in this chain, (a_i, a_{i+1}) is an arc for each i, the chain is a directed one. A chain in which all the a_i are distinct is a path. A chain in which $a_1 = a_n$ is called a cycle. Also paths and cycles that come from directed chains are directed paths and cycles.

Ford and Fulkerson* show that any set of $x(i, j) \geq 0$ which satisfies the balance equations may be expressed as the sum of flows along directed paths from S to $\bar{S}$ and directed cycles where the sum of flows along the paths is v. Furthermore, if both arcs (i, j) and (j, i) exist, any feasible flow value v may be obtained with non-zero flow on, at most, one of these arcs, by replacing $x(i, j)$ with $\max [x(i, j) - x(j, i), 0]$.

*Ibid.

III. CUT SETS

The solution method presented in this Memorandum will take advantage of the properties of a cut set, defined below. This definition is a generalization of the cut set definition given in Ref. 1 for solving network problems with arc capacities.

Definition: Given a set of nodes N and arcs A, consider two subsets, X and Y, of the nodes such that $X \cup Y = N$. If $S \in X$ and $\bar{S} \in Y$, then the set of nodes in $X \cap Y$ together with the set of arcs (i, j), such that $i \in X - Y$ and $j \in Y - X$, form a cut set separating S and $\bar{S}$. The value $V(X, Y)$ of this cut set is equal to the total capacity of its arcs and nodes.

If S and $\bar{S}$ are the source and sink, respectively, it follows that since the source is in X and the sink is in Y, any path from source to sink must use at least one arc or one node in the cut set. Thus the maximum flow cannot exceed the value of any cut set. In particular, it cannot exceed the minimum value of all cut sets. It will be shown later on that, as in networks with arc capacities only, the maximum flow equals the minimum cut. Some pertinent relationships between flows and cuts follow.

Lemma 1. Suppose a flow pattern consisting of one unit of flow along a directed chain from the source node S to node a is given. Suppose also that (X, Y) is a cut set and $S \in X - Y$. Then for $i \in X - Y$, $j \in Y$ (note that $X - Y$ and Y are complements):

$$\sum x(i, j) - \sum x(j, i) = \begin{cases} 1, & a \in Y \\ 0, & a \in X - Y \end{cases}.$$

Proof. A quick check shows that the theorem holds if the chain consists of one arc. Suppose it holds for k arc chains. Consider an arbitrary $k + 1$ arc chain, $S, a_1, \ldots, a_{k+1}$. There are four possible

cases to consider.

Case 1: $a_k, a_{k+1} \in Y$.

Since by hypothesis the theorem holds for k arc paths, the above quantity must be one if one neglects the unit of flow on arc (a_k, a_{k+1}). Furthermore, the unit of flow on this arc contributes nothing to the quantity in question, so it must still be one.

Case 2: $a_k \in Y$, $a_{k+1} \in X - Y$.

Neglecting the flow on the last arc yields a value of one for the quantity, and the unit of flow on arc (a_k, a_{k+1}) reduces it to zero.

Case 3: $a_k \in X - Y$, $a_{k+1} \in Y$.

Neglecting the flow on the last arc yields zero. Adding the flow on the last arc adds one to the quantity bringing it to one.

Case 4: $a_k, a_{k+1} \in X - Y$.

Neglecting the last arc yields zero for the quantity. Adding the unit of flow on the last arc does not change this.

Since the theorem is preserved in all four cases, the theorem itself follows from induction.

Corollary 1. Suppose a flow pattern is given consisting of one unit of flow along a directed cycle. Then the quantity referred to in Lemma 1 is zero.

Proof. Suppose the cycle contains a node in X - Y. Designating one such node as S, one has a directed chain from S to S for which the quantity in question must be zero by Lemma 1. On the other hand, if all

nodes of the cylce are in Y, then all x(i, j) in the summations referred to are zero and the theorem still holds.

Theorem 1: Suppose a flow pattern is given satisfying the balance equations (III) of Sec. II. Suppose also that $S \in X - Y$ and $\bar{S} \in Y$. Then

$$\sum x(i, j) - \sum x(j, i) = v, \quad i \in X - Y, \; j \in Y .$$

Proof: Multiplication by v and Lemma 1 show that the theorem holds if the flow pattern consists of v units of flow along a directed chain from S to $\bar{S}$. Similarly, Corollary 1 shows it holds if the pattern consists of units of flow only along one directed cycle. The theorem itself follows since the flow pattern can be decomposed into units of flow along directed chains from S to $\bar{S}$ and directed cycles such that the sum of these units on the chains is v.

Theorem 2: The maximum flow cannot exceed the value of any cut set separating S and $\bar{S}$.

Proof: Suppose $S \in X - Y$. By Theorem 1, $v \leq \sum x(i, j)$, $i \in X - Y$, $j \in Y$. However, the right hand side of the above inequality is equal to

$$\sum x(i, j) + \sum x(i, k), \quad i \in X - Y, \; j \in Y - X, \; k \in X \cap Y.$$

The first summation term in this expression cannot exceed the total arc capacities in (X, Y), and the second cannot exceed the total node capacities in (X, Y). Thus the theorem holds for $S \in X - Y$. If $S \in X \cap Y$, (X, Y) contains node S and therefore it suffices to show that $v \leq c(S)$. To see this, attach an artificial node S' and an artificial arc (S', S), each with infinite capacity. Let S' be the source and $\bar{S}$ the sink of the new network. Note that any feasible flow (one that satisfies the constraints of the linear program in Sec. II) can be converted into a

feasible flow for the new network by setting $x(S, S') = v$. Thus the maximum flow of the original network cannot exceed the maximum flow of the new one. Furthermore, letting X consist of the nodes S' and S, and Y consist of all nodes except S', gives a cut set whose value is at least as great as the maximum flow of the network (since $S' \in X - Y$) and is equal to the capacity of node S.

These relationships will be used to show that the algorithm presented in Sec. IV yields a maximum flow.

IV. A MAXIMUM FLOW ALGORITHM

The algorithm for finding a maximum flow pattern starts out with a flow of zero on all arcs. Then the algorithm attempts to find a chain from source to sink with unused capacity on all of its arcs and nodes. If one is found, as much flow as possible is sent across it. Then an attempt is made to find a new chain, until finally, none can be found and the algorithm terminates. The chains found will not necessarily be directed. This means that, in general, sending flow along the chain increases the flow on some arcs and decreases it on others. The chains found may contain cycles. It will be shown, however, that while nodes may appear more than once in a chain, they may not appear more than twice, and that arcs may appear only once. Furthermore, only a saturated node may appear twice with the included cycle reducing the node flow to balance out an increase by the rest of the chain. Flow can be pushed across a chain until either the flow on one of its arcs or nodes reaches capacity or until the flow on one of its arcs is reduced to zero.

The attempt to find a chain from source to sink is a labeling technique in which the presence of a label at a node indicates that a chain with unused capacity has been found from the source to it. Initially the source is labeled and all other nodes are unlabeled. Attempts are made to label the unlabeled nodes by examining the labeled ones. If the sink becomes labeled, units of flow are sent along the source-sink chain found and the labeling process is repeated in an attempt to find a new source-sink chain. The algorithm terminates when no such chain can be found.

The label at an arbitrary labeled node, a, will be designated by the three-component vector $L(a) = (a_1, a_2, a_3)$, with the individual components being referred to as $L_1(a)$, $L_2(a)$, and $L_3(a)$. In addition, each node, a, labeled or unlabeled will have a quantity $\bar{c}(a)$ which represents its unused capacity. The meaning of the label components are as follows: If $L_1(a) = b^+$, then a chain to node a has been found along which there are at least $L_3(a)$ units of unused capacity except possibly at node a, and the last arc along this chain is (b, a). If $L_1(a) = b^-$ then, as before, a chain has been found from the source to node a along which there are at least $L_3(a)$ units of unused capacity; however, in this case, the last arc is (a, b). Note that if units of flow were sent along the chain, an increase in the quantity x(b, a) would result in the former case, while a decrease in the quantity x(a, b) would result in the latter case. The quantity $L_2(a)$, if it exists, can only take the form b^- and has the same meaning as $L_1(a)$. It will be assigned only if no units may be sent to a along the chain designated by $L_1(a)$ without violating the node capacity at node a. This, of course, can only happen if $L_1(a)$ is of the form b^+, and $\bar{c}(a) = 0$. If $L_2(a)$ does not exist, it will be designated by the symbol *. When $L_2(a) \neq *$, in which case two such chains to a have been found, the quantity $L_3(a)$ will be such that $L_3(a)$ units may be sent along either chain.

If the above conditions hold, note that if i is labeled and j is unlabeled with $x(j, i) > 0$, then j may be labeled, since any chain to node i with unused capacity plus a flow reduction on arc (j, i) is a chain to j with unused capacity (Step 3a). If, in the above, one has $x(i, j) < c(i, j)$, node j can be labeled if either $L_1(a)$ or $L_2(a)$ is

of the form b^- or if $\bar{c}(i) > 0$ (Steps 3b and 3c). On the other hand, if i and j are both labeled and $L_1(j) = b^+$, $L_2(j) = *$, $\bar{c}(j) = 0$, and $x(j, i) > 0$, then $L_2(j)$ may be assigned the value i^- (Step 3d), thereby allowing one to label nodes, k, from j if $x(j, k) < c(j, k)$. These points are the basis of the maximum flow algorithm presented below.

MAXIMUM FLOW ALGORITHM

1. Set $\bar{c}(a) = c(a)$, all a, and all $x(i, j) = 0$.
2. Assign the source the label $[-, *, \infty]$ and let all other nodes be unlabeled.
3. If $\bar{S}$ is unlabeled*
   a. i is labeled, j unlabeled, and $x(j, i) > 0$, then assign j the label $[i^-, *, \min (x(j, i), L_3(i))]$.
   b. i is labeled with $L_1(i)$ or $L_2(i)$ equal to b^-, j is unlabeled, and $x(i, j) < c(i, j)$, assign j the label $[i^+, *, \min (c(i, j) - x(i, j), L_3(i))]$.
   c. i is the source or i is labeled with $L_1(i) = b^+$ and $\bar{c}(i) > 0$, j is unlabeled, and $x(i, j) < c(i, j)$, assign j the label $[i^+, *, \min (c(i, j) - x(i, j), L_3(i), \bar{c}(i))]$.
   d. i is labeled, j is labeled with $\bar{c}(j) = 0$, $L_1(j) = b^+$, $L_2(j) = *$, and $x(j, i) > 0$, change the label at node j to $[L_1(j), i^-, \min (x(j, i), L_3(i), L_3(j))]$.
4. Repeat Step 3 until either $\bar{S}$ is labeled or no more labels can be assigned. In the former case, increase the source-sink flow by $F = \min [\bar{c}(\bar{S}), L_3(\bar{S})]$ using the flow augmenting routine below. Otherwise, go to Step 6.

*For undirected networks, one considers arcs (i, j) and (j, i) at the same time in Step 3 by taking advantage of the fact that at most only one of x(i, j) and x(j, i) need be non-zero.

5. If $\bar{c}(\bar{S}) > 0$, go back to Step 2. Otherwise go to Step 6.

6. Terminate as the source-sink flow is optimal.

When the sink is labeled, the flow augmenting routine traces backwards a source-sink chain and sends min $[\bar{c}(\bar{S}), L_3(\bar{S})]$ units of flow along this chain. The basis for the path tracing routine is the fact that if node i is labeled, then the chain found by the labeling procedure to node i is the chain found to either node $|L_1(i)|$ or $|L_2(i)|$ followed by arc $(|L_1(i)|, i)$, $(i, |L_1(i)|)$, or $(i, |L_2(i)|)$ as the case may be. (If $L_1(i) = b^{\pm}$, then $|L_1(i)| = b$.) If the last arc in this chain is $(i, |L_1(i)|)$ or $(i, |L_2(i)|)$, then one may use the chain to $|L_1(i)|$ or $|L_2(i)|$ designated by $L_1(L_1(i))$ or $L_1(L_2(i))$. This is the reason for setting $L_2(a_{k+1}) = *$ in Cases 2 and 3. Also, in Case 3, $L_2(a_k)$ is set equal to *, since decreasing flow along arc (a_k, a_{k+1}) brings the flow at node a_k below capacity, thus allowing one to use the path to node a_k designated by $L_1(a_k)$ in future calculations.

FLOW AUGMENTING ROUTINE

1. Let $F = \min [\bar{c}(\bar{S}), L_3(\bar{S})]$. Set $k = 1$, $a_1 = \bar{S}$, and reduce $\bar{c}(\bar{S})$ by F/2 units.

2. Case 1: $L_2(a_k) = *$ and $L_1(a_k) = b^+$. Set $a_{k+1} = b$. Then increase the value of $x(a_{k+1}, a_k)$ by F units and reduce the value of $\bar{c}(a_k)$ and $\bar{c}(a_{k+1})$ by F/2 units.

   Case 2: $L_2(a_k) = *$ and $L_1(a_k) = b^-$. Set $a_{k+1} = b$. Then reduce the value of $x(a_k, a_{k+1})$ by F units, increase the value of $\bar{c}(a_k)$ and $\bar{c}(a_{k+1})$ by F/2 units, and set $L_2(a_{k+1}) = *$.

Case 3: $L_2(a_k) = b^-$. Set $a_{k+1} = b$ and then set $L_2(a_k) = *$ and $L_2(a_{k+1}) = *$.

Reduce the value of $x(a_k, a_{k+1})$ by F units and increase the value of $\bar{c}(a_k)$ and $\bar{c}(a_{k+1})$ by F/2 units.

3. Increase k by 1.

4. If $a_k = S$, reduce $\bar{c}(a_k)$ by F/2 units and terminate. Otherwise, go back to step 2.

Sections V and VI show that the maximum flow algorithm, together with the flow augmenting routine, actually yields a maximum flow from source to sink.

V. JUSTIFICATION OF THE FLOW AUGMENTING ROUTINE

This section is to show that upon termination of the flow augmenting routine, the constraints of the linear program of Sec. II hold. The next section will show, in addition, that the algorithm terminates with the objective function v maximized. The justification of the flow augmenting routine will be broken into three parts. The first will show that the node conservation equations are satisfied; the second that the nonnegativity and capacity constraints on the arc flows are satisfied; and the third that the node capacity constraints are satisfied.

Before proceeding to these three parts, the following theorem, which will be used throughout, must be proven.

Theorem 3. Let $L_k = L_1(a_k)$ if $L_1(a_k) = a^{\pm}_{k+1}$, or $L_k = L_2(a_k)$ if $L_2(a_k) = a^{-}_{k+1}$. Then L_{k+1} must have been assigned prior to L_k.

Proof. By Step 3 of the algorithm, $L_1(a_{k+1})$ must have been obtained prior to L_k, and the theorem holds if $L_{k+1} = L_1(a_{k+1})$. Suppose $L_{k+1} = L_2(a_{k+1})$. Then $\bar{c}(a_{k+1}) = 0$ by Step 3d. Furthermore, in determining a_{k+1} from the flow augmenting routine, Case 1 must have occurred, for if not, $L_2(a_{k+1})$ is immediately set to * contradicting $L_{k+1} = L_2(a_{k+1})$. Thus $L_1(a_k) = a^{+}_{k+1}$, and could only have been assigned this value by algorithm Step 3b, proving the theorem.

CONSERVATION CONSTRAINTS HOLD

The node conservation constraints (relationships III) are shown to hold by: Theorem 4, which shows that the procedure terminates with no intermediate nodes in the path traced being the source or the sink;

Lemma 2, which shows that flow may only leave the source and may only enter the sink; and Theorem 5 which shows that the conservation equations are indeed satisfied.

Theorem 4. The flow augmenting routine terminates with $a_k = S$. Furthermore, if $a_1, \ldots, a_n$ is the sequence obtained, then $k \neq 1$ implies $a_k \neq \bar{S}$ and $k \neq n$ implies $a_k \neq S$.

Proof. Suppose a_k, $k > 1$, is not the source. Since either a_k^+ is equal to $L_1(a_{k-1})$, or a_k^- is equal to $L_1(a_{k-1})$ or $L_2(a_{k-1})$, it follows from the steps of the labeling routine that a_k is labeled and hence $L_1(a_k)$, and possibly $L_2(a_k)$, exist. Thus a_{k+1} exists. Furthermore, if all $a_k \neq S$, the sequence of a_k must be infinite, and Theorem 3 would be contradicted. Thus the flow augmenting routine terminates with k equal to some number, n. By Theorem 3, $a_k = \bar{S}$ implies $k = 1$, since the algorithm terminates once the sink is labeled; and $a_k = S$ implies $k = n$, since the flow augmenting routine terminates once the source is reached.

Lemma 2. $x(j, S) = 0$ for all $j \in B(S)$ and $x(\bar{S}, j) = 0$ for all $j \in A(\bar{S})$.

Proof. Suppose the proposition holds at the beginning of the flow augmenting routine and $a_1, \ldots, a_n$ is the sequence of nodes found. $L_{n-1} = S^+$, where L_{n-1} is as defined in Theorem 3, for if not, either $L_1(a_{n-1})$ or $L_2(a_{n-1})$ is equal to S^-, implying $x(a_{n-1}, S) > 0$. Thus flow is increased on arc (S, a_{n-1}), and since only $a_n = S$, flow is unchanged on all other arcs with S as an endpoint. Similarly, $L_1 = a_2^+$, and flow is increased on arc $(a_2, \bar{S})$ and unchanged on all other arcs with $\bar{S}$ as an endpoint. Thus it holds at the routine's

conclusion and, consequently, the next time the routine is entered. Since it holds initially with all $x(i, j) = 0$, the theorem follows from induction.

Theorem 5. At the conclusion of the flow augmenting routine, the following node conservation constraints (III), repeated below, hold:

III

$$\text{a)} \sum_{j \in A(S)} x(S, j) - \sum_{j \in B(S)} x(j, S) = v$$

$$\text{b)} \sum_{j \in A(i)} x(i, j) - \sum_{j \in B(i)} x(j, i) = 0 \qquad i \neq S, \bar{S}$$

$$\text{c)} \sum_{j \in A(\bar{S})} x(\bar{S}, j) - \sum_{j \in B(\bar{S})} x(j, \bar{S}) = -v.$$

Proof: Suppose it holds initially. Let $a_1, \ldots, a_n$ be the sequence found by the flow augmenting routine. In Step 2, $x(a_{k+1}, a_k)$ is increased by F units if Case 1 holds, and $x(a_k, a_{k+1})$ is decreased by F units if Case 2 or Case 3 holds. In all three cases, $\sum x(a_k, j) - \sum x(\bar{j}, a_k)$, $j \in A(a_k)$, $\bar{j} \in B(a_k)$, is decreased by F units, and $\sum x(a_{k+1}, j) - \sum x(\bar{j}, a_{k+1})$, $j \in A(a_{k+1})$, $\bar{j} \in B(a_{k+1})$, is increased by F units. Consider the expression $\sum x(a_i, j) - \sum x(\bar{j}, a_i)$, $j \in A(a_i)$, $\bar{j} \in B(a_i)$. If $1 < i < n$, this expression is increased by F units when $k = i - 1$, and decreased by F units when $k = i$, for a total increase of zero. For $i = 1$, the expression is decreased by F units when $k = 1$; and for $i = n$, the expression is

increased by F when k = n - 1. Since $a_1 = \bar{S}$ and $a_n = S$, the theorem holds at the conclusion of the routine and, consequently, holds initially the next time the routine is entered, with v increased by F units. Since it holds when the routine is entered the first time with v = 0, the theorem follows from induction.

Thus it has been shown that the balance equations always hold at the conclusion of the flow augmenting routine.

ARC FLOW CONSTRAINTS HOLD

Using Lemmas 3 and 4, Theorem 6 will show that the flow augmenting routine preserves the arc constraints, $0 \leq x(i, j) \leq c(i, j)$.

Lemma 3. Let L_k be as defined in Theorem 3 and let $\bar{L}_3(a_k)$ be the value of $L_3(a_k)$ immediately after the value of L_k is assigned. Then $\bar{L}_3(a_k)$ is increasing in k and all $\bar{L}_3(a_k) \geq F$.

Proof: Let $\bar{\bar{L}}_3(a_{k+1})$ be the value of $L_3(a_{k+1})$ immediately prior to assigning the value of L_k. From Step 3 of the algorithm the $L_3(i)$ cannot increase. Thus it follows from Theorem 3 that $\bar{\bar{L}}_3(a_{k+1}) \leq \bar{L}_3(a_{k+1})$. Also from Step 3, $\bar{L}_3(a_k) \leq \bar{\bar{L}}_3(a_{k+1})$. From Step 1 of the flow augmenting routine $F \leq \bar{L}_3(a_1)$, proving the theorem.

Lemma 4. No arc appears more than once in the chain designated by the sequence of nodes $a_1, \ldots, a_n$ and the L_k found by the flow augmenting routine.

Proof: Suppose the proposition is false, and let $a_m = b$ be the last node in the sequence belonging to an arc that appears more than once. Then there is a $k < m$ such that $a_k = b$. Furthermore, from Theorem 3 and Step 3 of the algorithm, $L_m = L_1(b)$ and $L_k = L_2(b)$.

The existence of $L_2(b)$ implies $\bar{c}(b) = 0$, and since $L_2(b)$ could not have existed when L_{m-1} was assigned, $L_{m-1} = b^-$. Thus (a_{m-1}, b) is a repeating arc. Furthermore, a_{m-1} appears only once in the sequence, for if $a_{m-1} = a_r$, then $r < m-1$ would contradict Theorem 3 and $r > m-1$ would contradict our assumption on a_m. Thus $k = m-2$ and $L_k = a^+_{m-1}$, contradicting the fact that $L_k = L_2(a_k)$.

Theorem 6. The flow augmenting routine terminates with all $0 \leq x(i, j) \leq c(i, j)$.

Proof: Suppose it holds at the start of the routine. Define $\bar{L}_3(a_k)$ as in Lemma 3. From Step 3 of the algorithm, if $\bar{L}_k = a^-_{k+1}$, then $\bar{L}_3(a_k) \leq x(a_k, a_{k+1})$; if $\bar{L}_k = a^+_{k+1}$, then $\bar{L}_3(a_k) \leq c(a_{k+1}, a_k) - x(a_{k+i}, a_k)$. Since this holds for all k, it follows from Lemmas 3 and 4 that increasing flow along the chain found by F units preserves the conditions of the theorem, and it holds the next time the routine is entered. Since it holds initially with all $x(i, j) = 0$, the theorem follows from induction.

NODE CAPACITIES HOLD

It will be shown by Lemma 5 and Theorem 7 that the node capacities are not violated.

Lemma 5: These relationships hold: $\bar{c}(S) = c(S) - \sum x(S, j)$, $j \in A(S)$; $\bar{c}(\bar{S}) = c(\bar{S}) - \sum x(k, \bar{S})$, $k \in B(\bar{S})$; and for $i \neq S, \bar{S}$, $\bar{c}(i) = c(i) - \frac{1}{2}[\sum x(i, j) + \sum x(k, i)]$, $j \in A(i)$, $k \in B(i)$.

Proof: Initially the proposition holds with all $x(i, j) = 0$ and all $\bar{c}(i) = c(i)$. Suppose it does hold at one point and $a_1, \ldots, a_n$ is the sequence of nodes that the flow augmenting routine generates. The value of $x(a_n, a_{n-1}) = x(S, a_{n-1})$ is increased by F units since,

by Lemma 2, the value of $x(a_{n-1}, S)$ cannot be decreased by F units. Also, $\bar{c}(S)$ is reduced by F/2 units in Step 2, Case 1, and by another F/2 units in Step 4 for a total of F units, and the first equation is preserved. Similarly, the value of $x(a_2, a_1) = x(a_2, \bar{S})$ is increased by F units, and $\bar{c}(\bar{S})$ is reduced by F/2 units in Step 1 and by another F/2 units in Step 2, Case 1, preserving the second equation. For $k \neq 1, n$, note that $a_k \neq S, \bar{S}$ by Theorem 4. Note also from Step 2 of the flow augmenting routine that if either $x(a_k, a_{k+1})$ or $x(a_{k-1}, a_k)$ is increased by F units, $\bar{c}(a_k)$ is reduced by F/2 units; and that if either $x(a_k, a_{k+1})$ or $x(a_{k-1}, a_k)$ is decreased by F units, $\bar{c}(a_k)$ is increased by F/2 units. Thus the third equation is preserved for a_k. Since the third equation is unaltered for all nodes not in the sequence, $a_1, \ldots, a_n$, the lemma follows from induction.

Theorem 7. At the conclusion of the flow augmenting routine, $\sum x(i, j) \leq c(i)$, $j \in A(i)$ for $i \neq \bar{S}$ and $\sum x(k, \bar{S}) \leq c(\bar{S})$, $k \in B(\bar{S})$.

Proof: From Lemma 5, it is sufficient to show all $\bar{c}(i) \geq 0$. Let $a_1, \ldots, a_n$ be the sequence of nodes that the flow augmenting routine generates and let $\bar{L}_3(a_k)$ be defined as in Lemma 3. Initially all $\bar{c}(i) \geq 0$, since $c(i) > 0$. Assume it holds at the beginning of the flow augmenting routine. The only way to reduce $\bar{c}(a_i)$, $i \neq 1, n$, is for Case 1 to occur in Step 2 for $k = i$ and $k = i-1$; otherwise, the value of $\bar{c}(a_i)$ is raised by F/2 units when k is equal to at least one of these values and the largest net decrease possible would be zero. If Case 1 does occur for k equal to i and i-1, then $\bar{c}(a_i)$ is decreased by F units. It also follows that $L_1(a_{i-1}) = a_i^+$ and $L_1(a_i) = a_{i+1}^+$.

Thus $L_1(a_{i-1})$ must have been assigned its value by Step 3c of the algorithm and $\bar{c}(a_i) \geq \bar{L}_3(a_{i-1})$ and, consequently, $\bar{c}(a_i) \geq F$. No $\bar{c}(i)$ can be decreased more than once, for if so, one $a_k = a_m$, $L_k = L_1(a_m)$ and $L_m = L_1(a_m)$, which would contradict Theorem 3. Thus the theorem holds at the conclusion of the subroutine and hence, if the routine is entered again, at the beginning the next time. The theorem itself follows from induction.

It has now been shown that all network constraints hold after each iteration of the flow augmenting routine. It now remains to show that the algorithm terminates with v maximized.

VI. JUSTIFICATION OF THE ALGORITHM

This section shows that the maximum flow algorithm terminates with v maximized and that the maximum flow equals the minimum cut.

Note that if all flows and capacities are integers, the algorithm will only yield integer values for the $L_3(i)$; hence each iteration of the flow augmenting routine increases v by at least one unit. Thus if all arcs and nodes have finite capacities, the algorithm must eventually terminate. A similar argument holds for rational capacities. Lemma 6 and Theorem 8 will show that, at termination, flow is maximized and is equal to the value of a cut set.

Lemma 6. Suppose the algorithm terminates because the sink cannot be labeled. Consider the cut set where X consists of all labeled nodes, and Y consists of all unlabeled nodes and also all labeled nodes, a, such that there is an unlabeled node, b, with $x(a, b) < c(a, b)$. Then (i) $x(a, b) = c(a, b)$, if $a \in X - Y$ and $b \in Y - X$; (ii) $\bar{c}(a) = 0$ if $a \in X \cap Y$; (iii) $x(a, b) = 0$ if $a \in Y$ and $b \in X$.

Proof: Suppose $a \in X - Y$, $b \in Y - X$, and $x(a, b) < c(a, b)$. Then, since a is labeled and b is unlabeled, $a \in Y$ also for a contradiction. Thus (i) is proved. Suppose now that $a \in X \cap Y$. Let b be such that b is unlabeled and $x(a, b) < c(a, b)$. Then $\bar{c}(a) = 0$; if not, node b could either be labeled by Step 3b or Step 3c of the maximum flow algorithm, and (ii) is proved. Suppose now that $a \in Y$, $b \in X$, and $x(a, b) > 0$. Node $a \notin Y - X$, for if so, a would be unlabeled, b labeled, and consequently a could be labeled by Step 3a. Thus

$a \in X \cap Y$ and, consequently, $\bar{c}(a) = 0$. Also, either $L_1(a) = c^-$ or $L_2(a) = c^-$ for some c, otherwise node a's label could be changed by Step 3d. Let d be an unlabeled node such that $x(a, d) < c(a, d)$. Such a node must exist since $a \in X \cap Y$. Then node d could be labeled by Step 3b for a contradiction. Thus the theorem is proved.

Theorem 8. The maximum flow algorithm terminates with v maximized. Furthermore the maximum flow is equal to the minimum cut separating S and $\bar{S}$.

Proof: Since by Theorem 2 the maximum flow cannot exceed the value of any cut set, it is sufficient to show that, at termination, v is equal to the value of some cut set. Suppose the algorithm terminates with $\bar{c}(\bar{S}) = 0$. Consider the cut set where $X = \{\bar{S}\}$ and $Y = \{N,$ the set of all nodes$\}$. The value of this cut set is $c(\bar{S})$, and by Theorem 1 and Lemma 2 this is equal to v. Suppose the algorithm terminates because the sink cannot be labeled. Let X and Y be defined as in Lemma 6. If $S \in X \cap Y$, then $v = c(S)$ by Lemmas 2, 5, and 6 (ii). If $S \in X - Y$, then by Theorem 1 and Lemma 6 (iii), $v = \sum x(i, j)$, $i \in X - Y$, $j \in Y$. This sum may also be expressed as $v = \sum x(i, j) + \sum x(k, \ell) - \sum x(m, \ell)$ where $i \in X - Y$, $j \in Y - X$, $k \in X$, $\ell \in X \cap Y$, $m \in Y$. The first term in this sum is the total arc capacities of cut (X, Y) by Lemma 6 (i), and the second term is the total node capacities of cut (X, Y) by Lemma 6 (ii) and (iii), Lemma 5, and Theorem 5. The third term is zero by Lemma 6 (iii). Thus v is maximized and is equal to the value of the minimum cut set.

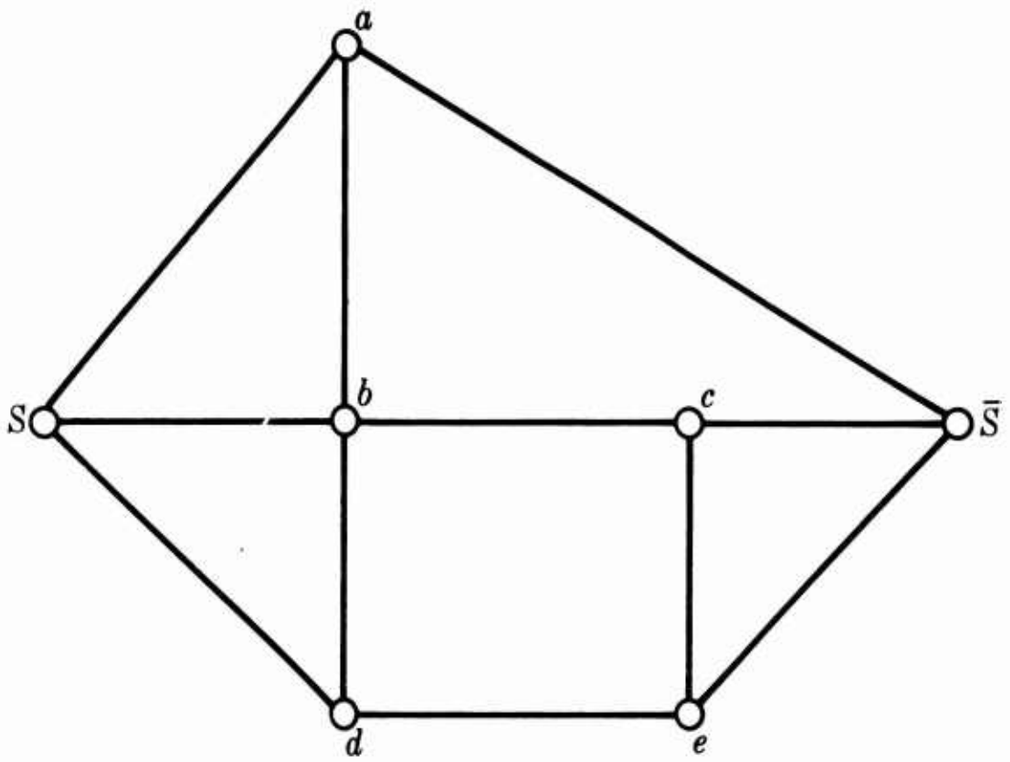
This completes the justification of the algorithm.

VII. EXAMPLE

This section uses the algorithm presented to find the maximum flow from S to $\bar{S}$ for the undirected network of Fig. 2. For purposes of labeling, the arcs are examined in the order of their listing in Fig. 2. Among the results listed for each iteration are the final labels obtained, the path found by the flow augmenting routine, the resulting arc flows, and the final $\bar{c}(i)$ for the nodes. Note that in the fourth iteration (Fig. 6) the chain found contains a cycle. In the fifth iteration, $\bar{S}$ was not reached, terminating the procedure. The minimum cut, determined by the sets $X = \{S, b\}$ and $Y = \{a, b, c, d, e, \bar{S}\}$, consists of arcs (S, a) and (S, d), and node b.

It was mentioned earlier that any flow pattern of value v can be expressed as the sum of flows on directed chains from source to sink and directed cycles, such that flow is non-zero in at most one direction between any pair of nodes and such that the flows on the chains sum up to v. The flow pattern the algorithm finds can be expressed as the sum of directed chains only. It is fairly simple to find such chains.

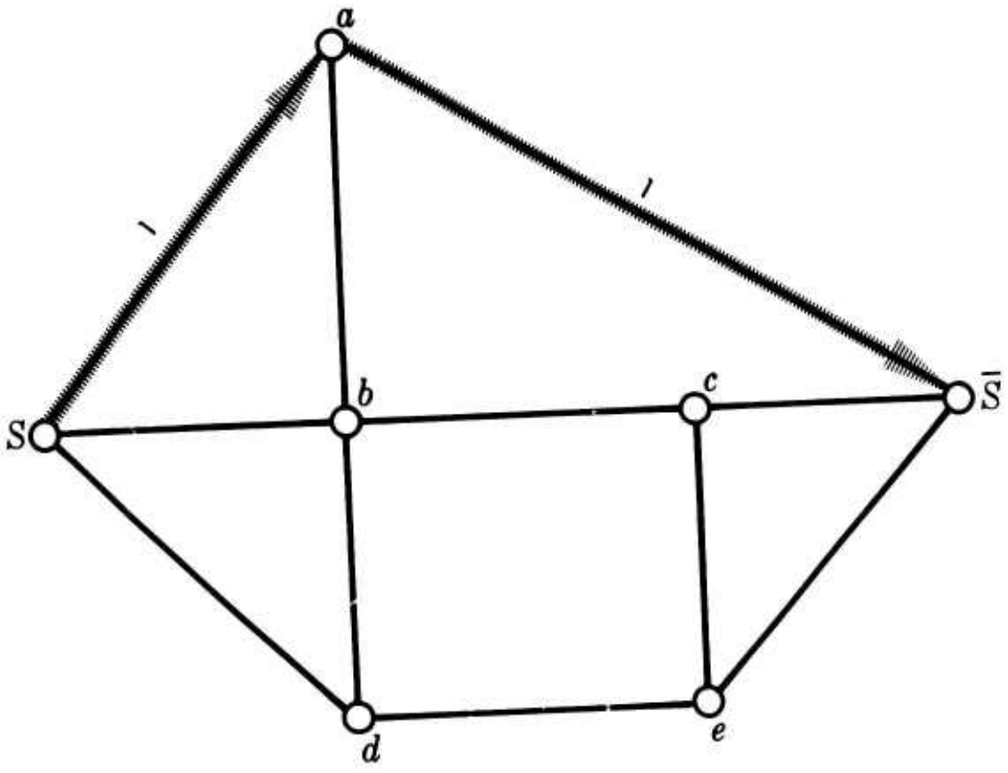
Note that the iterations of the flow augmenting routine yield chains (not necessarily directed) such that the total flow along all of them sums up to a maximum flow pattern if one relaxes the constraints $x(i, j) \leq c(i, j)$ to $x(i, j) - x(j, i) \leq c(i, j)$. It is desired to replace these chains by directed chains such that at most only one of $x(i, j)$ and $x(j, i)$ is non-zero.



Arc	Capacity
(S,a)	1
(a,$\bar{S}$)	10
(S,d)	2
(d,e)	10
(e,$\bar{S}$)	1
(b,d)	10
(b,c)	10
(c,$\bar{S}$)	1
(S,b)	10
(a,b)	10
(c,e)	10

Node	Capacity
S	10
a	2
b	1
c	1
d	10
e	10
$\bar{S}$	10

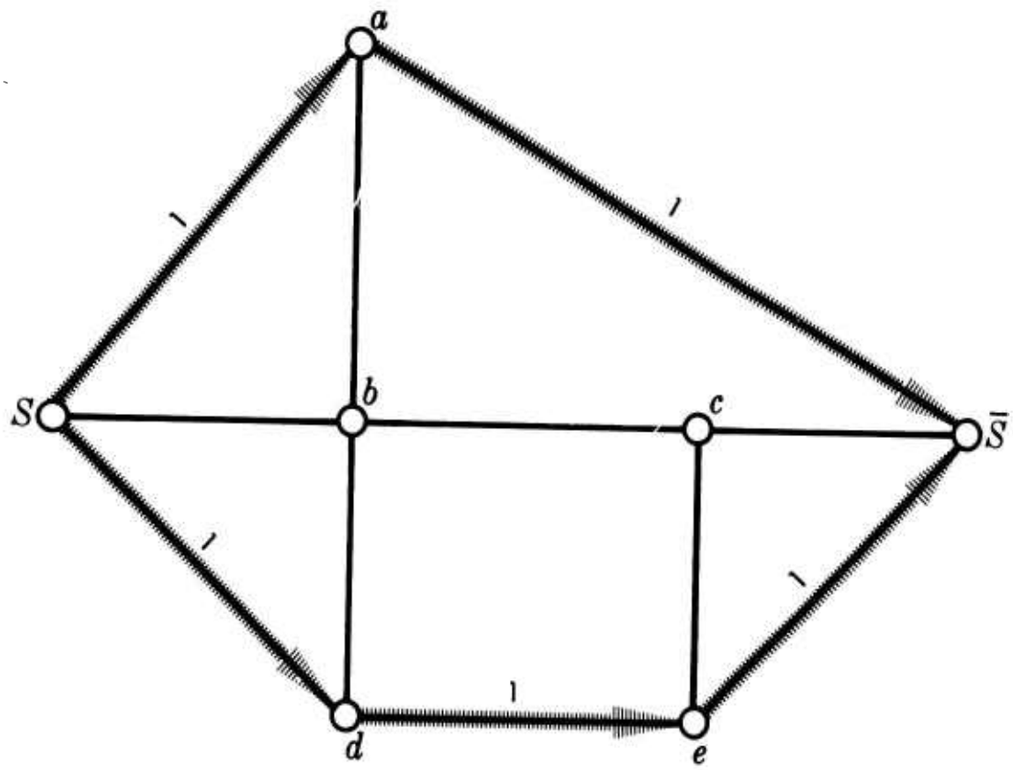
Fig. 2 -- Example of a network with node and arc capacities



Arc	Flow
(S,a)	1
(a,$\bar{S}$)	1
(S,d)	0
(d,e)	0
(e,$\bar{S}$)	0
(b,d)	0
(b,c)	0
(c,$\bar{S}$)	0
(S,b)	0
(a,b)	0
(c,e)	0

Node	Label	$\bar{c}$(a)
S	(-, *, ∞)	9
a	(S^+, *, 1)	1
b		1
c		1
d		10
e		10
$\bar{S}$	(a^+, *, 1)	9

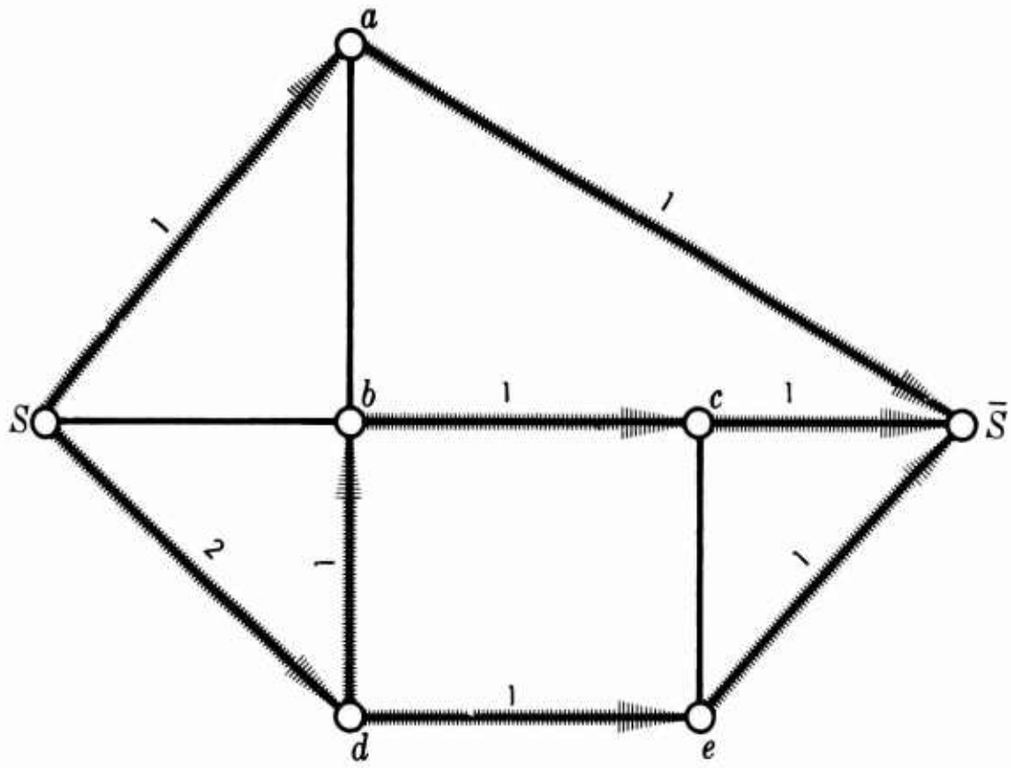
Fig. 3 -- Iteration 1: The chain found is S,a,$\bar{S}$



Arc	Flow
(S,a)	1
(a,$\bar{S}$)	1
(S,d)	1
(d,e)	1
(e,$\bar{S}$)	1
(b,d)	0
(b,c)	0
(c,$\bar{S}$)	0
(S,b)	0
(a,b)	0
(c,e)	0

Node	Label	$\bar{c}(a)$
S	(-, *, ∞)	8
a		1
b		1
c		1
d	(S^+, *, 2)	9
e	(d^+, *, 2)	9
$\bar{S}$	(e^+, *, 1)	8

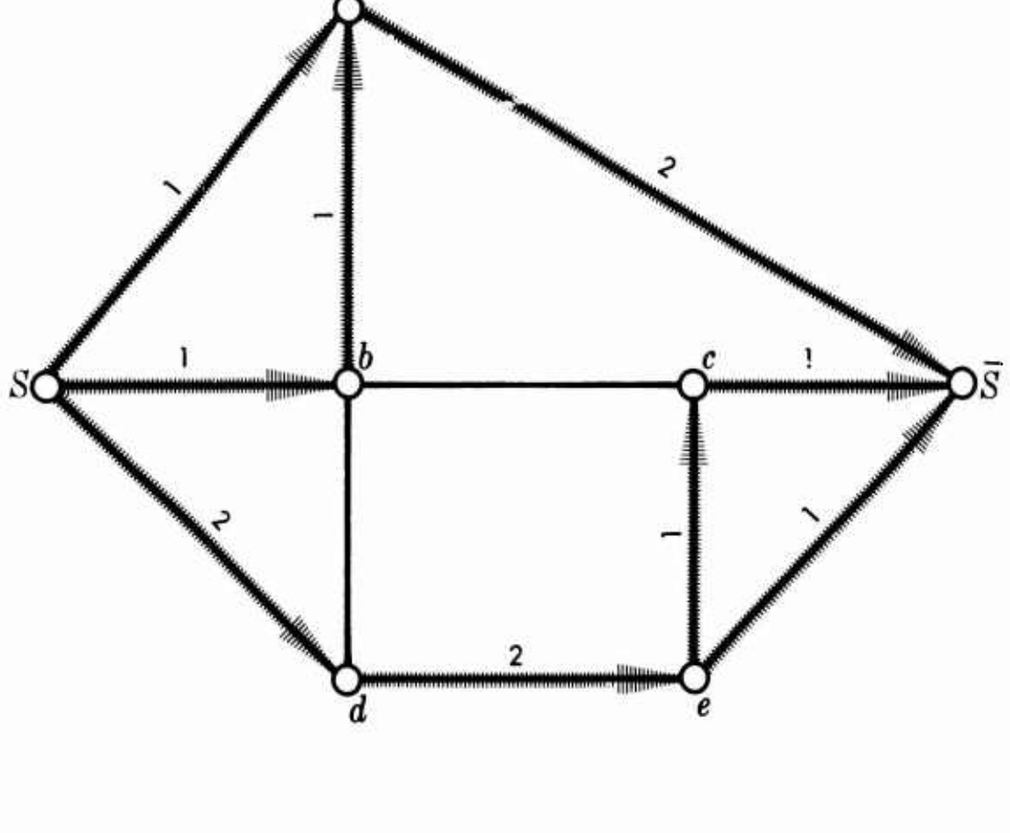
Fig. 4 -- Iteration 2: The chain found is S,d,e,$\bar{S}$



Arc	Flow
(S,a)	1
(a,$\bar{S}$)	1
(S,d)	2
(d,e)	1
(e,$\bar{S}$)	1
(b,d)	1
(b,c)	1
(c,$\bar{S}$)	1
(S,b)	0
(a,b)	0
(c,e)	0

Node	Label	$\bar{c}(a)$
S	(-, *, ∞)	7
a		1
b	(d^+, *, 1)	0
c	(b^+, *, 1)	0
d	(S^+, *, 1)	8
e	(d^+, *, 1)	9
$\bar{S}$	(c^+, *, 1)	7

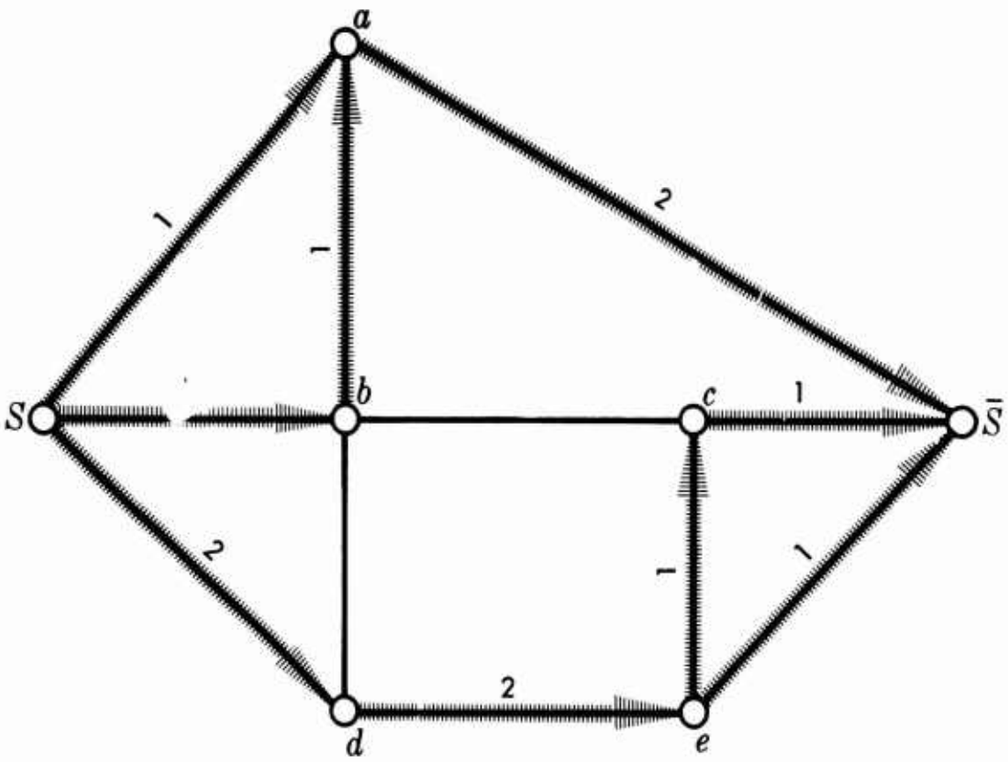
Fig. 5 -- Iteration 3: The chain found is S,d,b,c,$\bar{S}$



Arc	Flow
(S,a)	1
(a,$\bar{S}$)	2
(S,d)	2
(d,e)	2
(e,$\bar{S}$)	1
(b,d)	0
(b,c)	0
(c,$\bar{S}$)	1
(S,b)	1
(a,b)	1
(c,e)	1

Node	Label	$\bar{c}$(a)
S	(-, *, ∞)	6
a	(b^+, *, 1)	0
b	(S^+, c^-, 1)	0
c	(e^+, *, 1)	0
d	(b^-, *, 1)	8
e	(d^+, *, 1)	8
$\bar{S}$	(a^+, *, 1)	6

Fig. 6 -- Iteration 4: The chain found is S,b,d,e,c,b,a,$\bar{S}$



Arc	Flow
(S,a)	1
(a,$\bar{S}$)	2
(S,d)	2
(d,e)	2
(e,$\bar{S}$)	1
(b,d)	0
(b,c)	0
(c,$\bar{S}$)	1
(S,b)	1
(a,b)	1
(c,e)	1

Node	Label	$\bar{c}(a)$
S	(-, *, ∞)	6
a		0
b	(S^+, *, 6)	0
c		0
d		8
e		8
$\bar{S}$		6

Fig. 7 -- Iteration 5: The cut set is determined by the node subsets X = {S,b} and Y = {a,b,c,d,e,$\bar{S}$}. It consists of node b and arcs (S,a) and (S,b)

Suppose the flow pattern consists of a unit of flow along the chain, $a_1, \ldots, a_n$, which reduces the flow on arc (a_{i+1}, a_i) by one unit. Then there must also be a unit of flow along a chain, $b_1, \ldots, b_m$ with $b_j = a_{i+1}$ and $b_{j+1} = a_i$. One may replace these two chains by the two others, $a_1, \ldots, a_{i-1}, b_{j+1}, \ldots, b_m$ and $b_1, \ldots, b_{j-1}, a_{i+1}, \ldots, a_n$, thus reducing each of $x(a_i, a_{i+1})$ and $x(a_{i+1}, a_i)$ by a unit. Repeated application of this technique eventually eliminates all wrong way flows and thereby yields directed chains. Similar reductions on directed chains which use arcs (i, j) and (j, i) yield directed chains summing up to a flow pattern where at most one of x(i, j) and x(j, i) are non-zero for all (i, j).

For the example of this section, the algorithm yielded the four chains:

1. S, a, $\bar{S}$
2. S, d, e, $\bar{S}$
3. S, d, b, c, $\bar{S}$
4. S, b, d, e, c, b, a, $\bar{S}$.

Eliminating a unit of flow on arcs (d, b) and (b, d, from chains 3 and 4 gives:

1. S, a, $\bar{S}$
2. S, d, e, $\bar{S}$
3. S, d, e, c, b, a, $\bar{S}$
4. S, b, c, $\bar{S}$.

Then a unit of flow on arcs (c, b) and (b, c) of chains 3 and 4 yields the desired chains:

BIBLIOGRAPHY

Dantzig, G. B., "Application of the Simplex Method to a Transportation Algorithm," in Activity Analysis of Production and Allocation, Cowles Commission Monograph 13, John Wiley and Sons, New York, 1951, pp. 359-373.

Durbin, E. P., An Interdiction Model of Highway Transportation, The RAND Corporation, RM-4945-PR, May 1966.

Fetter, R. B., and R. C. Steorts, A Model for the Design and Evaluation of Air Cargo Systems, The RAND Corporation, RM-4801-PR, October 1966.

Ford, L. R., and D. R. Fulkerson, Flows in Networks, The RAND Corporation, R-375-PR, December 1960.

-----, "Maximal Flow Through a Network," Can. J. Math,, 8, 1956, pp. 399-404.

Wollmer, R. D., "Removing Arcs From a Network," J. Oper. Res. Soc. of Amer., 12, 1964, pp. 934-940.

-----, Some Methods for Determining the Most Vital Link in a Railway Network, The RAND Corporation, RM-3321-ISA, April 1963.

DOCUMENT CONTROL DATA

1 ORIGINATING ACTIVITY	2a. REPORT SECURITY CLASSIFICATION
THE RAND CORPORATION	UNCLASSIFIED
	2b. GROUP

3. REPORT TITLE

MAXIMIZING FLOW THROUGH A NETWORK WITH NODE AND ARC CAPACITIES

4. AUTHOR(S) (Last name, first name, initial)

Wollmer, R. D.

5. REPORT DATE	6a. TOTAL No. OF PAGES	6b. No. OF REFS.
July 1967	42	---

7. CONTRACT OR GRANT No.	8. ORIGINATOR'S REPORT No.
F44620-67-C-0045	RM-5383-PR

9a. AVAILABILITY / LIMITATION NOTICES	9b. SPONSORING AGENCY
DDC-1	United States Air Force Project RAND

10. ABSTRACT

A method for maximizing the flow from source to destination along a transportation network whose nodes and arcs have limited capacities. An algorithm is derived that eliminates the need for artificial arcs and nodes used in the flow-augmenting method of Fulkerson and Ford. This simplification should provide a substantial savings in computer time and core storage. The problem is formulated as a linear program. For all nodes other than the source and sink, the node capacity may be expressed either as the flow into or out of the node, since these two quantities are equal. For the source and sink, the node capacity must be expressed in terms of the greater of the two quantities. Thus the source node capacity limits flow out of the source, while the sink capacity limits flow into the sink. The solution method used, a cut set approach, is a generalization of Ford and Fulkerson's labeling method.

11. KEY WORDS

Network theory
Mathematics
Linear Programming
Transportation
Logistics