FINDING THE n MOST VITAL LINKS IN FLOW NETWORKS

By

GUY THOMAS SICILIA

A DISSERTATION PRESENTED TO THE GRADUATE COUNCIL OF
THE UNIVERSITY OF FLORIDA IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA
1971

TO

THE THREE MOST IMPORTANT PEOPLE

IN MY LIFE

MY MOTHER, AGATHA MARIE

MY FATHER, PETER

AND

MY WIFE, ANNE

ACKNOWLEDGMENTS

It is only just that when speaking of acknowledgments, Dr. H. D. Ratliff, my supervisory committee chairman, be the first mentioned. His assistance and direction are greatly appreciated.

For assistance in finding a meaningful topic and in carrying out the research, I am deeply grateful to Dr. S. Lubore. I would also like to thank Mr. S. Haus with whom I had many helpful discussions.

The research for this thesis was supported partly by the Army Research Office (Project Themis) and by the Mitre Corporation (as much of the research was accomplished while the author was employed by Mitre). Mr. J. Selinka and Dr. W. Yondorf were largely responsible for this latter support. For this assistance I am very grateful.

Finally, my typist, Mrs. Larrick, and my wife, Anne, deserve a great deal of gratitude as they were very patient while trying to decipher my handwriting and preparing this manuscript.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the
Graduate Council of the University of Florida in Partial Fulfillment
of the Requirements for the Degree of Doctor of Philosophy

FINDING THE n MOST VITAL LINKS
IN FLOW NETWORKS

By

Guy Thomas Sicilia

December, 1971

Chairman: Dr. H. D. Ratliff
Major Department: Industrial and Systems Engineering

The n most vital links of a flow network are defined as those n arcs whose simultaneous removal from the network causes the greatest decrease in the throughput capability of the remaining system between a specified node pair. These n arcs are shown to be the n largest capacity arcs in some cut. A solution procedure is developed which involves sequentially modifying the network in such a manner as to make this cut minimal. An algorithm with computational results is presented.

CHAPTER 1

INTRODUCTION

The research presented in this dissertation is concerned with the problem of optimally removing components of a flow network. There are many problems embedded in this part of network theory. This is due to the multitude of considerations which are involved in stating and solving a network problem. For instance, one must note the type of network involved; which and how many network elements are candidates for removal; and what is the measure of optimality (i.e., what was the purpose of posing the problem).

The problem of removing arcs from a network so as to stop all flow from a set of sources, S, to a set of sinks, T, at a minimal cost (where each arc requires an assigned cost for removal) is referred to in the literature as the Attacker's Problem. The related problem of reinforcing a network so as to most increase the cost of the optimal solution to the Attacker's Problem is referred to as the Defender's Problem. In this latter problem there is an assigned cost which is required to improve any link and a budget consideration which limits the "improvements" that can be made. Thus, there are several sets of improvements which the defender could make and remain within his budget. Any such set of improvements is called cost feasible and each of these, if implemented, would lead to a different reinforced network.

The Defender's Problem is to find that cost-feasible set of improvements which gives a reinforced network with the property that the optimal solution to the Attacker's Problem used against this reinforced network is at least as large as the optimal attacker's solution against any of the other feasible reinforced networks. Jarvis [19] solves both the Attacker's and Defender's Problems for the single-commodity case by using the Generalized Lagrange Multiplier Method of Everett [10] and Nunn [24], and the transshipment model of either Busacker and Gowen [6] or Ford and Fulkerson [11]. In the same paper, Jarvis solves the multi-commodity Attacker's Problem and formulates, but does not solve, the multi-commodity Defender's Problem. Ratliff [26] solves the multi-commodity Defender's Problem for a network by solving a series of multi-commodity minimal disconnecting set problems and a series of set-covering problems. Ratliff uses a method developed by Bellmore, Greenberg, and Jarvis [3] to solve the former series of problems and a method of Bellmore and Ratliff [4] to solve the latter series of problems. Greenberg [15] defines a more general Attacker's Problem than that of Jarvis. His method of solution for the single-commodity case involves the Generalized Lagrange Multiplier Method and the maximum flow algorithm [11]. He solves the multi-commodity Attacker's Problem by modifying a 0-1 integer programming method described by Geoffrion [14] and Balas [1].

It can be noted that by setting the cost of destroying an arc equal to its capacity, the Attacker's Problem becomes synonymous with the minimal disconnecting set problem. Bennington [5] gives a method to solve the latter problem for an undirected network. He also shows that the multi-commodity minimal disconnecting set problem can be posed as a node isolation problem. A discussion of the isolation problem and a method to solve it are given by Bellmore, Bennington, and Lubore [2].

The notion of assigning values to different elements of the network has also been employed by various people. Hobbs [18] assigns numbers to the links and nodes of a communication network to reflect their relative importance. The determination of these "importance numbers" which are based on the configuration of the network and the assigned value of each commander--subordinate pair, is equal to a sum of products. The sum is taken over the possible commander--subordinate pairs, where each element of the sum is a product of the value of the particular commander--subordinate pair and the percent of chains connecting the commander and subordinate which contain the link in question. Krokowski [20] also suggests that importance values be assigned to links and nodes, but he proposes that a good measure of the value of a network component (in a war situation) would be the replacement cost of the element. He recommends a linear programming approach but does not develop the solution beyond the conceptual stage.

The effect of changing the length of a link on some or all of the shortest paths between nodes in a network is addressed by Halder [17]. In fact, he locates the shortest path distances that will be affected by increasing the length of an arc for a symmetric distance network (a network where d(a,b) = d(b,a) for all nodes a, b in the set N).

It should also be noted that the general subject of network vulnerability and synthesis can be viewed strictly as a graph theory problem. For instance, Frank and Frisch in [13] present analysis methods for problems similar to, and yet different from, those previously mentioned. In particular, they consider an Attacker-like Problem in which the objective is to determine the minimum number of nodes, arcs, or arcs and nodes that must be removed to: (1) stop all possible flow between designated node sets, and (2) partition the network into two disjoint sets (i.e., source and sink nodes are not specified and breaking the communication between any pair of nodes is acceptable). They also consider the problem of constructing a network, subject to cost constraints, for which the number of elements that must be removed to partition the network into two components is maximal.

Problem Definition

The problem to be addressed in this paper is to determine those n arcs (or links) whose simultaneous removal from the network causes the greatest decrease in the maximal possible flow between some specified node pair. This problem will be referred to as the n Most Vital Links Problem and the best set of n arcs to remove will be called the n most vital links. Note that if removing n arcs stops all the flow between a source and sink, then this problem becomes the Attacker's Problem where the cost to destroy any arc is the same.

The following definitions and notation will be used throughout this dissertation.

A network will be denoted by $G(N,A)$, where $N = \{x_1, x_2, \ldots\}$ is a finite collection of elements x_i, each representing a node in the network, and A is a subset of the pairs (x,y) of elements from N representing arcs in the network. (Another characterization of the arcs which will be employed is $a \in A$.) The number of nodes in N will be denoted by $|N|$ and the number of arcs in A by m. If the arc (x,y) is an ordered pair, then it is called a directed arc; otherwise, it is an undirected arc. For the results presented in this dissertation, arcs may be considered as either directed or undirected.

Given the network $G(N,A)$, if a non-negative number denoted by either $c(x,y)$ or $c(a)$ is assigned for each arc in A, then this value will be defined as the capacity of the arc (x,y), as it can be thought of as the maximal amount of some commodity that can arrive at node y from node x per unit time.

One of the fundamental notions in network theory is the concept of the maximal flow between a designated pair of nodes (s and t). Given the network $G(N,A)$ and a capacity $c(x,y)$ for each $(x,y) \in A$, a question which arises is: what is the maximum amount of steady state flow that can travel from node s to node t over the network $G(N,A)$, subject to the condition that the flow over any arc is no larger than the capacity of the arc? Ford and Fulkerson [11] formulate this problem as a linear programming problem with a special structure. This structure allows the use of very efficient methods (the maximum flow algorithm) for solving the problem. Let $f(x,y)$ be a variable which corresponds to the flow over arc (x,y) and v the net flow out of node s, or equivalently, the net flow into node t (i.e., $v = \sum_{y \in N} f(s,y) - \sum_{y \in N} f(y,s)$

for all pairs which are arcs in the network). A statement of the maximal flow problem is:

Max v

such that

$$\sum_{y \in N} f(x,y) - \sum_{y \in N} f(y,x) = \begin{cases} v & \text{if } x = s \\ 0 & \text{if } x \neq s,t \\ -v & \text{if } x = t \end{cases}$$

$$f(x,y) \leq c(x,y) \quad \text{for all } (x,y) \in A$$

$$f(x,y) \geq 0 \quad \text{for all } (x,y) \in A$$

Finally, two related sets of arcs, disconnecting sets and cuts, will be discussed. A cut, $(X,\bar{X})$, separating a designated pair of nodes s and t is a set of arcs defined by:

$$(X,\bar{X}) = \{(x,y) \mid (x,y) \in A,\ x \in X,\ \text{and}\ \ y \in \bar{X}\}$$

where the sets X and $\bar{X}$ are any partition of the set of nodes, N, such that $s \in X$, $t \in \bar{X}$, $X \cap \bar{X} = \phi$, and $X \cup \bar{X} = N$. (A cut, on occasion, will also be denoted by A^*.) A disconnecting set separating a designated pair of nodes s and t is a set of arcs which partitions the nodes of N into p sets; $X_1, X_2, \ldots, X_p$ such that if $s \in X_i$, then $t \notin X_i$ for $i = 1,2,\ldots,p$; $X_i \cap X_j = \phi$ for $i \neq j$; and $\bigcup_{i=1}^{p} X_i = N$. Thus, a cut is a special type of disconnecting set, i.e., a disconnecting set for which $p = 2$. The capacity or value of a cut and a disconnecting set will be taken as the sum of the capacities of the arcs in the cut and disconnecting set, respectively.

Letting a cut be defined as minimal if its value is at least as small as the value of every cut of G(N,A), Ford and Fulkerson [11] prove that the maximal flow value of G(N,A) is equal to the minimal cut value of the same network.

Using the notation and results just discussed, the n most vital links problem may now be restated as:

> Let E_i for i = 1,2,...,p be all subsets of A that contain exactly n elements (the cardinality of the set E_i will be denoted by $|E_i|$), and let $C_o(N,A)$ be the capacity of a minimum cut in G(N,A). Find the arc set E_k such that $C_o(N,A-E_k) \leq C_o(N,A-E_j)$ for j = 1,2,...,p.

Applications

Some applications of the most vital links problem include:

(1) A conflict situation where there is a logistics or communications network, a defender or user of this system and an interdictor of the system. The user wishes to know which arcs are most vital to him so that he can reinforce these arcs against attack, while the attacker, naturally, wants to destroy those arcs which most decrease the efficiency of the system.

(2) A traffic control situation where, during evening rush hour, most motorists go from the city (source) to the suburbs (sinks). A question which arises is: If n road segments were to become blocked what effect would this have on the rush hour flow and the blockage which

n road segments would most decrease flow? One use for this information arises in the scheduling of work crews to the transportation network. For example, given there are n different repair crews, simultaneously assigning them to the n most vital links would not be advisable.

Previous Contributions

The most vital links problem and other very similar problems previously have been addressed. These findings are mentioned in this section with the most significant results being outlined in more detail.‡

Wollmer [29] has devised an algorithm for finding the single most vital link in a network. Also, by solving shortest path problems [11], for a dual graph [7], in [27] he has been able to solve for the n simultaneous most vital links for a planar undirected network. Durbin [9] uses Wollmer's single most vital link approach to target strikes for an attacker of a given network. Each arc is given a repair time (in time periods) and n strikes are made for each time period on that part of the network which is currently operative. Ondrasek and Wollmer [25] address a problem, similar to, but more general than that of Durbin as they add the consideration of repair time and cost of an arc in determining which arcs to attack.

‡The single most vital link problem allows solution methods which cannot be efficiently applied to the more general problem. For this reason only results concerning the more general problem are presented here. Chapter 2 will be solely devoted to the single most vital link problem.

The most significant contributions in this area are due to Wollmer [27,28], and McMasters and Mustin [22]. Their results can be summarized as follows:

<u>Results due to Wollmer</u>

Letting a network G(N,A) be defined as <u>planar</u> if it can be drawn on a plane such that no two arcs intercept except at nodes, and as <u>source-sink planar</u>, if the network generated by adding to G(N,A) an arc from the source to sink is planar, the problem proposed by Wollmer can be stated as:

> Consider a source-sink undirected planar network in which each arc is subject to, at most, one breakdown: a breakdown resulting in a reduction in the capacity of that arc by a known quantity. It is desired to find the greatest reduction in maximum flow possible from, at most, n breakdowns and where these breakdowns must occur, to give this reduction.

If a given network, the primal network, is source-sink planar there exists a related dual network. Wollmer's solution technique uses these two networks and a relationship developed by Ford and Fulkerson in [12]; i.e., the result is:

> A one to one correspondence exists between the cut sets of the original or primal network and the loopless path through the dual network (in the dual network the arcs have lengths rather than capacities). The problem of finding the minimum cut set in the primal network is equivalent to finding the shortest path through the dual.

Thus, finding the maximal flow value of a source-sink planar network is equivalent to solving a shortest path problem through the dual. Also, the problem proposed by Wollmer can be stated in terms of the dual network as:

Let each arc of the dual network have two lengths assigned to it, i.e., the original and the reduced length (where in the primal network these correspond to the original and reduced capacities, respectively). Assuming that the lengths of at most n arcs can be reduced, the problem is: the length reduction of which n arcs will lead to the shortest possible path through the network?

Wollmer, using the fact that shortest path problems are in general easier to solve than other network problems, developed an algorithm which solves $n+1$ network problems. The first or 0^{th} problem is a shortest path problem for the dual network. The other n problems, for $i = 1,\ldots,n$, solve the following: Given that the shortest path problem in which at most $i-1$ arcs may have their length decreased has been solved, what is the optimal solution to a shortest path problem which allows at most i arcs to have a reduced length? Wollmer solves the i^{th} subproblem by finding the shortest i-arc paths between the source and all the other nodes of the network, where an i-arc path to node a is any path from the source to node a along which at most i arcs have their length reduced. The algorithm uses information gained by solving the $i-1^{st}$ subproblem to solve the i^{th} subproblem, i.e., all the shortest $i-1$ arc path lengths are used in finding the set of the shortest i-arc paths.

Results due to McMasters and Mustin

The interdiction problem posed by McMasters and Mustin in [22] is not exactly the most vital links problem as defined in this paper.

However, the problem which they solve is very closely related to the stated problem and is a natural parallel problem or extension to the most vital links problem. Thus, the problem statement and a brief description of the solution method aregiven in this section.

The problem in the notation of its originators is:

Consider an undirected planar network, [N;A]. Each arc (i,j) in the network is assumed to have a maximum flow capacity, $u_{ij} \geq 0$, and a minimum flow capacity, $l_{ij} \geq 0$, and it is assumed that at least one arc has $l_{ij} > 0$. After the interdiction has taken place, the actual capacity of an arc, m_{ij}, will satisfy $0 \leq l_{ij} \leq m_{ij} \leq u_{ij}$.

Assuming that the interdictor incurs a cost, C_{ij}, per unit of capacity destroyed, then this total cost for reducing an arc's capacity from u_{ij} to m_{ij} is $C_{ij}\,[u_{ij} - m_{ij}]$. If the interdiction has a budget of K, then the interdictor's problem is to find a set of m_{ij} which minimizes the maximum flow value of the network subject to:

$$\sum_{(i,j)\in A} C_{ij}\,[u_{ij} - m_{ij}] \leq K$$

$$l_{ij} \leq m_{ij} \leq u_{ij}, \text{ for all } (i,j) \in A.$$

The authors, like Wollmer, rely on the original network being planar and undirected. Also, the dual network is used in this method as it was in Wollmer's. A very brief description of the method will now be given. The actual statement of the algorithm is rather lengthy and will not be included.

The first step of the algorithm is to set the capacities of each arc to its lower bound and to find a minimum cut for this "reduced" network by solving a shortest path problem in the dual network. If this solution is feasible to the above interdictor's problem, then it is optimal, since this is as good as even an infinitely wealthy attacker could do. If the solution is not feasible, the best feasible solution allowed by interdicting only the arcs of the minimum cut is found and another minimum cut to the reduced problem is chosen and tested. This procedure ends in one of two states: either a minimum cut solution is feasible and thus is optimal or no minimum cut solution is feasible but the current "best solution" has been noted. If the latter condition holds, the cuts of the reduced network are tested in increasing value (actually the r^{th} shortest path problem is found in the dual graph). This testing and updating of the current "best solution" is continued until, for the "k^{th} smallest cut," the best possible solution for this cut, i.e., the solution with all the capacities equal to the lower bound, is not better than the current best feasible solution. At this point, the algorithm terminates as all the remaining cuts to the reduced problem will have values greater than a feasible solution of another cut.

The remainder of this paper is divided as follows. Chapter 2 treats the single most vital links problem; a solution method for this special case is given and an example is used to show how the solution may be found. Chapters 3 and 4 discuss the more general n most vital links problem. Algorithms are developed and the computational experience using them is described in Chapter 5. In Appendix A, a linear zero-one and a zero-one quadratic formulation of the n most vital links problem are presented and discussed.

CHAPTER 2

THE SINGLE MOST VITAL LINK PROBLEM

Introduction

The problem addressed by this chapter is concerned with finding the most vital link in a single commodity flow network $G(N,A)$ (directed, undirected, or mixed). An arc $(x,y) \in A$ is declared to be the most vital link if its value $v(x,y)$ is at least as large as the value of every other arc in the network. The value of arc (x,y) is defined as the difference in the maximal flow values in networks $G(N,A)$ and $G(N,A-(x,y))$ between some source node and some sink node. Thus, $v(x,y)$ reflects the reduction in the maximal flow attainable if arc (x,y) is removed from the network.

Wollmer has developed an algorithm [29] for determining the most vital network link. This algorithm has been employed by Durbin [9] to determine the single most critical link in a highway system.

The following sections of this paper briefly present Wollmer's method for finding the most vital link in a network and develop an improved algorithm for solving the most vital link problem. An example using the improved method is included.

Wollmer's Algorithm

Wollmer's algorithm for finding the most vital link in a network follows as a consequence of the following theorem. The proof of this theorem is given in reference [29].

Theorem (Wollmer):

Suppose the network $G(N,A)$ has a maximal flow value of $v(F^*)$ while the network $G(N,A-(x,y))$ has a maximal flow value of $v(F^*_{xy})$. Then, every maximal flow pattern of $G(N,A)$ has at least $v(F^*)-v(F^*_{xy})$ units of flow through the arc (x,y). Moreover, there is a maximal flow pattern of $G(N,A)$ which has exactly $v(F^*)-v(F^*_{xy})$ units of flow over the arc (x,y).

As Wollmer points out, "the above theorem reduces the problem to one of finding that link whose minimal flow among all maximal flow patterns is greatest." Wollmer's iterative procedure for finding this link is as follows:

STEP 0:

Find a maximal flow pattern, F^*, in the network $G(N,A)$ and let $f^*(x,y)$ be the corresponding flow in each arc $(x,y) \in A$. Set the "least flow" of each arc (x,y) equal to $f^*(x,y)$. Let $f^*(p,q) = \max\limits_{(x,y) \in A} f^*(x,y)$ and go to STEP 1.

STEP 1:

Solve a maximal flow problem for the network $G(N,A-(p,q))$. Let the corresponding maximal flow pattern be denoted as F^*_{pq} and go to STEP 2.

STEP 2:

(a) Set the capacity $c(p,q)$ of arc (p,q) equal to $v(F^*) - v(F^*_{pq})$ and solve a maximal flow problem for this network (i.e., the network $G(N,A)$ with $c(p,q) = v(F^*) - v(F^*_{pq})$). Call the corresponding maximal flow pattern F'. (Note F' is also a maximal flow pattern of $G(N,A)$.)

(b) Next, compare the least flow of each arc (x,y) with $f'(x,y)$ and if $f'(x,y) <$ least flow of arc (x,y), replace the least flow of (x,y) with $f'(x,y)$. Reset $c(p,q)$ to its original value and go to STEP 3.

STEP 3:

Let $U = \max_{(x,y) \in A} \{\text{least flow of arc } (x,y)\}$.

(a) If $U \leq v(F^*) - v(F^*_{pq})$ terminate; (p,q) is a most vital link;

(b) If $U > v(F^*) - v(F^*_{pq})$; find an arc (p,q) such that the least flow of (p,q) equals U, and go to STEP 1.

An Improved Algorithm

Wollmer's algorithm considers each arc as a candidate for the most vital link. However, a necessary condition is employed in the improved algorithm which reduces the number of arcs that must be considered explicitly as candidates.

Theorem 1:

A necessary condition for an arc (a,b) to be a most vital link is that for any maximal flow pattern in the network $G(N,A)$, the flow in arc (a,b) is at least as great as the flow over every arc in a minimal cut.

The following lemma will be useful in the proof of Theorem 1.

<u>Lemma 2.1</u>:

If $(X,\bar{X})$ is a minimal cut containing at least two arcs in a network $G(N,A)$ and if arc (x,y) is in $(X,\bar{X})$, then $(X,\bar{X}) - (x,y)$ is a minimal cut in the network $G(N,A-(x,y))$.

Proof: (Lemma 2.1)

Suppose that $(Y,\bar{Y})$ is a minimal cut in $G(N,A-(x,y))$ and that $C(Y,\bar{Y}) < C(X,\bar{X}) - c(x,y)$. Note that $(Y,\bar{Y}) \cup (x,y)$ has to be a disconnecting set for $G(N,A)$ and that $C(Y,\bar{Y}) + c(x,y) < C(X,\bar{X})$: but $(X,\bar{X})$ is a minimal cut of $G(N,A)$ and $(Y,\bar{Y}) \cup (x,y)$ is a disconnecting set of $G(N,A)$. Thus, it must be true that $C(Y,\bar{Y}) = C(X,\bar{X}) - c(x,y)$ and, hence, $(X,\bar{X}) - (x,y)$ is a minimal disconnecting set and thus a minimal cut in $G(N,A-(x,y))$.

Proof: (Theorem 2.1)

Let $(X,\bar{X})$ be a minimal cut in $G(N,A)$ and note that by hypothesis arc (a,b) is a most vital link. Assume that $f^*(a,b) < f^*(p,q) = \max\limits_{(x,y)\in(X,\bar{X})} f^*(x,y)$ for some maximal flow pattern F^* defined in $G(N,A)$. It will be shown that this assumption leads to a contradiction.

(1) By Lemma 2.1: $v(F^*_{pq}) = v(F^*) - f^*(p,q)$; by assumption $f^*(a,b) < f^*(p,$ therefore, $v(F^*_{pq}) = v(F^*) - f^*(p,q) < v(F^*) - f^*(a,b)$.

(2) Further, there exists a flow pattern in $G(N,A-(a,b))$ with the value $v(F_{ab}) = v(F^*) - f^*(a,b)$, and, hence, the maximal flow value $v(F^*_{ab})$, in $G(N,A-(a,b))$, must satisfy: $v(F^*_{ab}) \geq v(F^*) - f^*(a,b)$.

(3) Conditions (1) and (2) imply that $v(F^*_{ab}) > v(F^*_{pq})$; but this leads to a contradiction since by assumption (a,b) is a most

vital link and by the definition of a most vital link it follows that: $v(F^*_{ab}) \leq v(F^*_{xy})$, for all $(x,y) \in A$.

Q. E. D.

Theorem 1 guarantees that those arcs whose flow is less than the largest flow through an arc in a minimal cut for some maximal flow pattern in G(N,A) need not be considered as candidates for the most vital link.

A Labeling Scheme

A labeling scheme is employed in the algorithm for finding a most vital link from the set of candidate arcs. At the outset of the labeling, it is assumed that there is a maximal flow pattern, and minimal cut defined for G(N,A). Suppose that arc (a,b) is an arc from the candidate set (this set is made up of those arcs that satisfy the necessary condition of Theorem 1). Next, an attempt is made to label from node a to node b without using the arc (a,b). Initially node a is labeled and all other nodes are unlabeled. The labeling scheme systematically searches for a flow path from node a to node b which does not include arc (a,b) such that flow may be diverted from arc (a,b) over this path. If node b is labeled (breakthrough), such a path exists. The labeling (and backtracking) rules to be used are similar to those employed by Ford and Fulkerson [11] in their algorithm for the solution of the maximal flow problem. The labeling and flow changing rules are:

Let node a be labeled with $(-, \epsilon(a) = \infty)$. At a general step, suppose the node x is labeled $(z\pm, \epsilon(x))$ and that the nodes x and y are connected by some arc. Then, node y may be labeled if either of the following situations occur:

(a) $f(y,x) > 0$; then node y is labeled $(x^-, \epsilon(y))$ where $\epsilon(y) = \min(\epsilon(x), f(y,x))$; or

(b) $f(x,y) < c(x,y)$; then node y is labeled $(x^+, \epsilon(y))$ where $\epsilon(y) = \min(\epsilon(x), c(x,y) - f(x,y))$.

The labeling process is continued until either node b is labeled (breakthrough), or node b is unlabeled in which case no more labeling is possible (non-breakthrough). If breakthrough occurs, node b must have a label of the form $(q^{\pm}, \epsilon(b))$ for some node q. Likewise, node q has a label $(r^{\pm}, \epsilon(q))$ and node r has a label $(p^{\pm}, \epsilon(r))$, etc. Thus a series of nodes (and a path of arcs) starting with node b and ending with node a is defined.

Let $\epsilon = \min[\epsilon(b), f(a,b)]$. For each arc (x,y) on this path, change the flow as follows:

(a) If node y has a label $(x^+, \epsilon(y))$, replace $f(x,y)$ by $f(x,y) + \epsilon$;

(b) If node y has a label $(x^-, \epsilon(y))$, replace $f(y,x)$ by $f(y,x) - \epsilon$.

Finally, decrease the flow in arc (a,b) by ϵ units.

As Ford and Fulkerson have shown, the labeling scheme must end in one of two mentioned states: breakthrough is achieved or breakthrough is not achieved. The following results indicate what can be deduced when either of these states occur at the end of the labeling process.

<u>Theorem 2</u>:

Let F^* be a maximal flow pattern in the network G(N,A). Let (a,b) be an arc in G(N,A) and use the labeling scheme to label from node a to node b without using arc (a,b). If breakthrough occurs, the value of arc (a,b) is less than or equal to $f^*(a,b) - \epsilon$. If non-breakthrough occurs, the value of arc (a,b) is equal to $f^*(a,b)$.

Proof:

If breakthrough occurs an alternate maximal flow pattern is found by making an ϵ change of flow in the path established by the labeling and decreasing the flow in (a,b) by ϵ. Then,

$$v(F^*_{ab}) \geq v(F^*) - f^*(a,b) + \epsilon, \text{ and, since}$$

$$v(a,b) = v(F^*) - v(F^*_{ab}) \leq v(F^*) - [v(F^*) - f^*(a,b) + \epsilon] \text{ and}$$

$\epsilon > 0$, it follows that $v(a,b) \leq f^*(a,b) - \epsilon < f^*(a,b)$.

Assume that non-breakthrough occurs, then by the nature of the labeling rules, no maximal flow pattern exists in G(N,A) which has less than $f^*(a,b)$ units of flow over arc (a,b) and, hence, by Wollmer's theorem $v(a,b) = f^*(a,b)$.

Q. E. D.

Utilizing these results, the following algorithm can be employed to locate a most vital link in a network:

STEP 0:

(a) Find a maximal flow pattern, F^*, in the network G(N,A) and let $(X,\bar{X})$ be a minimal cut.‡ Let $U^* = \max_{(x,y)\epsilon(X,\bar{X})} c(x,y)$ or, alternately,

$$U^* = \max_{(x,y)\epsilon(X,\bar{X})} f^*(x,y).$$

‡The Ford and Fulkerson maximal flow algorithm [11] may be used since it identifies a minimal cut as well as a maximal flow pattern.

(b) Note those arcs $(x,y) \in A$ for which $f^*(x,y) \geq U^*$ and store these arcs in a list; these arcs form the candidate set, S. For each arc in this set define an upper bound as $U(x,y) = f^*(x,y)$.

STEP 1:

Let $U(a,b) = \max\limits_{(x,y) \in S} U(x,y)$ and set $f(x,y) = f^*(x,y)$ for all arcs in $G(N,A)$.

STEP 2:

Use the labeling rules to label from node a to node b without using arc (a,b).

STEP 3:

(a) If breakthrough occurs, use the backtracking and flow changing rules, replace $f(x,y)$ by the resultant flow in each $(x,y) \in A$, and if $f(a,b) > 0$, repeat STEP 2.‡

(b) Otherwise, replace $U(a,b)$ with $f(a,b)$ and if $U(a,b) \geq \max\limits_{(x,y) \in S} U(x,y)$ terminate; arc (a,b) is a most vital link.

Otherwise, replace $f^*(x,y)$ with $f(x,y)$ for all $(x,y) \in A$ and go to STEP 1.

By the use of this algorithm, a maximal flow pattern is always maintained and once non-breakthrough results for the arc (a,b) and

‡An alternative to STEP 3a is to test if $f(x,y) < U(x,y)$, $(x,y) \in S$, and if $f(x,y) \geq U^*$, to replace $U(x,y)$ with $f(x,y)$; if $f(x,y) < U^*$, then arc (x,y) may be dropped from the set S. The computations are continued by either repeating STEP 2 if arc (a,b) has not been dropped or returning to STEP 1 if arc (a,b) has been dropped.

$U(a,b) \geq U(x,y)$ for all arcs (x,y) of the candidate set S, the arc (a,b) can be declared a most vital link, i.e., non-breakthrough implies that $v(a,b) = f^*(a,b) = U(a,b)$ and, thus, $v(a,b) \geq U(x,y) \geq v(x,y)$ for $(x,y) \in S$ and by Theorem 1 the most vital link of G(N,A) is an element of S.

The other aspects of the method that must be considered are: (1) is the method finite and (2) is there a way to locate alternative optimal solutions? The finiteness of the procedure follows since there can be only a finite number of arcs in the candidate set and there is a finite number of labelings that can be made for each arc of the set. Alternative optima are readily identified as follows: after an optimal solution has been found and other candidate arcs exist, reapply the algorithm to any arc in the candidate set whose upper bound is equal to the value of the most vital link (a,b), i.e., set $U^* = U(a,b)$, delete arc (a,b) from the candidate set, and reapply the algorithm. If upon reapplication of the algorithm the most vital link has a value equal to U^*, then an alternative solution has been found. In the latter case, the algorithm is reapplied using a further reduced candidate set and the procedure is continued until all alternative optimal solutions have been found.

An Example

Consider the flow network G(N,A) shown in Figure 2.1.

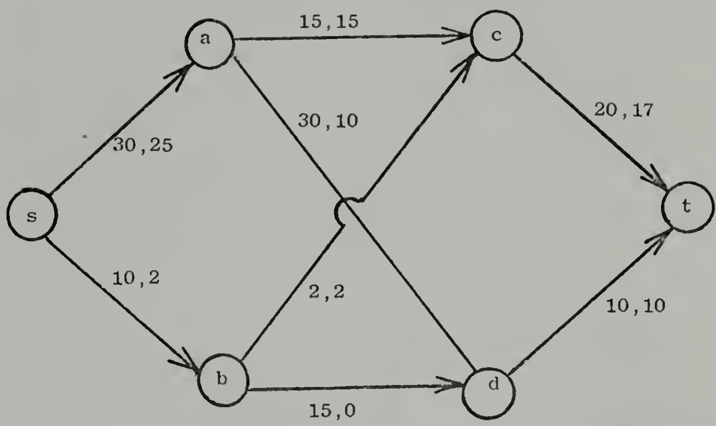

Figure 2.1. Maximal Flow in Sample Network 2.1 with Source Node s and Sink Node t.

Associated with each arc in the network is an ordered pair of numbers. The first number corresponds to the capacity of the arc and the second number corresponds to the amount of flow over the arc in some maximal flow pattern.

The improved algorithm will be employed to find all most vital links in the network G(N,A).

STEP 0:

(a) Figure 2.1 presents a maximal flow pattern F^* in the network with 27 units of flow from node s to node t. The individual arc flows are:

$f^*(s,a) = 25$ $\quad$ $f^*(a,c) = 15$

$f^*(s,b) = 2$ $\quad$ $f^*(a,d) = 10$

$f^*(c,t) = 17$ $\quad$ $f^*(b,c) = 2$

$f^*(d,t) = 10$ $\quad$ $f^*(b,d) = 0$

The corresponding minimal cut (which is unique) contains the arcs (a,c), (b,c), and (d,t). Hence,

$$U^* = \max_{(x,y)\in(X,\bar{X})} f^*(x,y) = f^*(a,c) = 15$$

(b) The set of candidate arcs is:

$$S = \{(s,a),\ (c,t),\ (a,c)\}$$

STEP 1:

Start with arc (s,a), since $U(s,a) = f^*(s,a) = \max_{(x,y)\in S} U(x,y)$.

STEP 2:

Labeling from node s to node a without using arc (s,a) is possible over the path containing the arcs (s,b), (b,d), and (a,d). The value of $\epsilon = 8$.

STEP 3:

(a) Since breakthrough occurred, the ϵ units of flow are removed from arc (s,a) and added to each arc in the path found in STEP 2. The revised maximal flow pattern is shown in Figure 2.2.

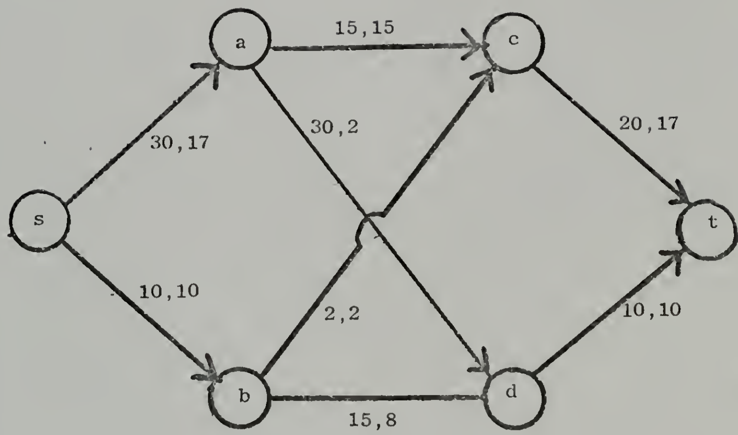

Figure 2.2. Revised Maximal Flow in Sample Network 2.1.

STEP 2:

Starting with the maximal flow pattern of Figure 2.2, it is not possible to label from node s to node a and non-breakthrough has occurred.

STEP 3:

(b) Replace $U(s,a)$ with the flow $f(s,a) = 17$. Since

$$U(s,a) \geq \max_{(x,y) \in S} U(x,y) = U(c,t),$$ arc (s,a) is a most vital link.

Since $U(c,t) = U(s,a) = 17$, arc (c,t) may also be a most vital link. In order to test this hypothesis, arc(s,a) is dropped from the set S and the algorithm is reapplied starting with the maximal flow pattern in Figure 2.2.

Employment of the labeling rules from node c to node t results in non-breakthrough and arc (c,t) is an alternative most vital link.

CHAPTER 3

THE n MOST VITAL LINKS PROBLEM

Introduction

As stated in Chapter 1, determining the n most vital links of a network involves locating that set of n arcs whose removal most impairs the throughput capability of the remaining network. In Chapter 2, results pertaining to the special case where $n = 1$ were given. While the algorithm developed in that chapter is, with minor changes, valid for the more general problem it is not computationally feasible for large problems with $n > 1$. Furthermore, as Wollmer in [29] notes, the single most vital link may not be one of the n most vital links. Thus, the single most vital link problem will be treated as a special case and another approach will be employed for the more general problem.

A Cut Problem Approach

In this chapter the n most vital links problem is formulated as a cut problem and an algorithm, which uses the maximum flow algorithm of Ford and Fulkerson [11], is developed to solve this problem.

Theorem 3.1 and Corollary 3.1.1 characterize the n most vital links of G(N,A) as elements in a cut between the source and sink of G(N,A).

Theorem 3.1:

If the arcs in A'_n are the n most vital links in the network, $G(N,A)$, then there exists a cut A' such that $A'_n \subseteq A'$ and $A' - A'_n$ is a minimum cut in $G(N,A-A'_n)$.

Proof:

Find the minimum cut in $G(N,A-A'_n)$, say A''_n with capacity $C(A''_n)$. Now consider the arc set $A''_n \cup A'_n$. This is certainly a disconnecting set for $G(N,A)$. Suppose it is not a cut. Then there exists at least one arc $a_i \in A''_n \cup A'_n$ such that $A''_n \cup A'_n - \{a_i\}$ is also a disconnecting set for $G(N,A)$. If $a_i \in A''_n$, then A''_n is not a minimum cut for $G(N,A-A'_n)$. If $a_i \in A'_n$, then $A'_n \cup \{a_k\} - \{a_i\}$, where a_k is any arc in A''_n, is a better set of n links to remove from $G(N,A)$. In either case a contradiction results. Hence, by letting $A' = A'_n \cup A''_n$, the result follows.

Q. E. D.

Corollary 3.1.1:

$C(A'_n) \geq C(A^*_n)$ for all arc sets, $A^*_n \subseteq A'$, which contain at most n arcs.

Proof:

Suppose $C(A^*_n) > C(A'_n)$ for some $A^*_n \subseteq A'$ containing at most n arcs. Note that for any $A^*_n \subseteq A'$

$$C(A') = C(A^*_n) + C(A'-A^*_n) = C(A'_n) + C(A'-A'_n),$$

hence, $C(A'-A^*_n) < C(A'-A'_n)$, but this implies that A'_n is not the n most vital links in $G(N,A)$ which is a contradiction.

Q. E. D.

Corollary 3.1.2:

$C(A'_n) \geq C(A'''_n)$ where A''' is any minimum cut of $G(N,A)$ and A'''_n contains the n largest arcs of A'''.

Proof:

This result follows since $C(A'-A'_n) \leq C(A'''-A'''_n)$ (or $C(A') - C(A'_n) \leq C(A''') - C(A'''_n)$) and $C(A') \geq C(A''')$.

Q. E. D.

Note that due to Theorem 3.1 and Corollary 3.1.1, the n most vital links problem of $G(N,A)$ can be stated as the following cut problem:

Find the n largest arcs, A^*_n, of the cut A^* which satisfies:

$$C(A^*-A^*_n) = \min C(A'-A'_n)$$

for all cuts A'

where A'_n denotes the n largest arcs of A'.

Development of a Cut Algorithm

Let $G^u(N,A)$ be the network $G(N,A)$ with the capacity of each arc $a_i \in A$ defined as:

$$c^u(a_i) = \begin{cases} c(a_i) & \text{if } c(a_i) < u \\ u & \text{if } c(a_i) \geq u. \end{cases}$$

Note that $G^\infty(N,A) \equiv G(N,A)$. For any $A_k \subseteq A$ let

$$c^u(A_k) = \sum_{a_i \in A_k} c^u(a_i).$$

<u>Theorem 3.2:</u>

If A' is a minimum cut in $G^u(N,A)$ with $A'_u \equiv \{a_i \mid a_i \in A' \text{ and } c^u(a_i) = u\}$, and $\alpha = |A'_u|$ is defined as the number of arcs contained in A'_u, then the arcs in A'_u are the α most vital links in $G^\infty(N,A)$.

Proof:

A'_u contains the α largest arcs in A' for both $G^u(N,A)$ and $G^\infty(N,A)$. Suppose A'_u does not contain the α most vital arcs in $G^\infty(N,A)$. Then, from Theorem 3.1 there exists a cut A^* such that:

$$c^\infty(A^*-A^*_u) < c^\infty(A'-A'_u) \qquad (1)$$

where A^*_u are the α largest arcs of A^* in $G^\infty(N,A)$.

Note that $c^\infty(a_i) < u$ for all $a_i \in A' - A'_u$, since

$$c^\infty(a_i) \geq u \Rightarrow c^u(a_i) = u \Rightarrow a_i \in A'_u .$$

Hence,

$$c^\infty(A'-A'_u) = c^u(A'-A'_u) \ . \qquad (2)$$

Equations (1) and (2) imply that

$$c^\infty(A^*-A^*_u) < c^u(A'-A'_u)$$

but

$$c^u(A^*-A^*_u) \leq c^\infty(A^*-A^*_u) \ .$$

Hence,

$$c^u(A^*-A^*_u) < c^u(A'-A'_u)$$

or

$$c^u(A^*) - c^u(A^*_u) < c^u(A') - c^u(A'_u) \ .$$

Now, since $c^u(a_i) \leq u$ for all $a_i \in A^*_u$ and $c^u(a_i) = u$, for all $a_i \in A'_u$, $C^u(A^*_u) \leq C^u(A'_u)$ which implies that $C^u(A^*) < C^u(A')$ which is a contradiction, since A' is a minimum cut of $G^u(N,A)$.

Q. E. D.

Corollary 3.2.1:

The maximum flow through the network $G(N,A-A'_u)$ is equal to $C(A'-A'_u)$.

Proof:

$A' - A'_u$ is a minimum cut in $G^u(N,A-A'_u)$ and, since $C^u(A'-A'_u) = C^\infty(A'-A'_u)$, $A' - A'_u$ is also a minimum cut in $G^\infty(N,A-A'_u)$.

Q. E. D.

Corollary 3.2.2:

Let $A'_x = \{a_i \mid a_i \in A'$ and $c^\infty(a_i) > u\}$ where $|A'_x| = \alpha$ and $A'_p = \{a_i \mid a_i \in A'$ and $c^\infty(a_i) = u\}$ where $|A'_p| = \beta$. Then $A'_x \bigcup_{i=1}^{k} a_i$ are the $\alpha + k$ most vital links of $G(N,A)$ for $a_i \in A'_p$ and any $k = 0,1,2,\ldots,\beta$.

Proof:

Same as for Theorem 3.2.

Q. E. D.

Let a minimum cut of $G^u(N,A)$ be denoted by A^u and define A_u to be the set of arcs defined by $A_u \equiv \{a \mid a \in A^u$ and $c^u(a) = u\}$.

Theorem 3.3:

Let η be defined as the smallest number of arcs contained in any cut (i.e., η is the minimum number of arcs necessary to disconnect the network $G^{\infty}(N,A)$), then

$$\eta = |A_1| \geq |A_2| \geq \ldots \geq |A_u| \geq \ldots \geq |A_{\infty}| \geq 0.$$

Proof:

From the definition of A_u it follows that $|A_{\infty}| \geq 0$. Also, by noting that in $G^1(N,A)$ all arcs have a capacity of 1, it follows that the minimum cut of $G^1(N,A)$ must be that cut which contains the fewest arcs (i.e., $\eta = |A_1|$).

It must now be shown that $p < q$ implies $|A_p| \geq |A_q|$. Assume that this is not the case, i.e., assume that $p < q$ and $|A_p| < |A_q|$.

Note that by Theorems 3.1 and 3.2 the arcs in A_p are the $|A_p|$ most vital links of $G(N,A)$ and the maximal flow value of $G(N,A-A_p)$ is $C(A^p-A_p)$. Similarly, the arcs in A_q are the $|A_q|$ most vital links of $G(N,A)$ and removing A_q from $G(N,A)$ leaves a maximal flow value of $C(A^q-A_q)$ in $G(N,A^q-A_q)$. Also, since it is assumed that $|A_q| > |A_p|$, it follows that

$$C^{\infty}(A^p-A_p) \geq C^{\infty}(A^q-A_q) \ . \tag{3}$$

Since for any arc $a \in A^p$, $c^{\infty}(a) \geq p$ if and only if $a \in A_p$, it follows that

$$C^{\infty}(A^p-A_p) = C^p(A^p-A_p),$$

(where $C^p(A^p-A_p) = C^p(A^p) - p|A_p|$).

Similarly, it follows that

$$c^{\infty}(A^q - A_q) = c^q(A^q - A_q) = c^q(A^q) - q|A_q| .$$

Using these identities the inequality (3) can be written as

$$c^p(A^p) - p|A_p| \geq c^q(A^q) - q|A_q| . \tag{4}$$

Finally, note that

$$c^q(A^q - A_q) \geq c^p(A^q - A_q),$$

or equivalently

$$c^q(A^q) - q|A_q| \geq c^p(A^q) - p|A_q| .$$

Thus, using this last relationship with (4) and the assumption that $|A_q| > |A_p|$, it follows that

$$c^p(A^p) > c^p(A^q) .$$

But this is a contradiction, since A^p is a minimum cut of $G^p(N,A)$ and, thus, $|A_q| \leq |A_p|$.

Q. E. D.

The Modified Cut Method

Theorems 3.2 and 3.3 provide the basis for a very simple algorithm to solve the n most vital links problem for some values of n. The algorithm consists of defining a graph as in Theorem 3.2 and then systematically changing the value of u. Stated formally, the algorithm, "the modified cut method," becomes:

STEP 0:

Let $u = 1$ and go to STEP 1.

STEP 1:

$$\text{Set } c^u(a_i) = \begin{cases} c^\infty(a_i) & \text{if } c^\infty(a_i) < u \\ u & \text{if } c^\infty(a_i) \geq u . \end{cases}$$

Solve for the minimum cut of $G^u(N,A)$.

Note A^u, A_u, and $|A_u|$. Go to STEP 2.‡

STEP 2:

If $|A_u| > n$ let $u = u + 1$. Go to STEP 1.

If $|A_u| = n$ terminate as the n most vital links are the arcs contained in A_u.

If $|A_u| < n$ terminate, a gap has occurred at the n most vital links problem.

‡Note that, due to Corollary 3.2.2, if k of the $|A_u|$ ceiling arcs have $c^\infty(a_i) = u$, then the $|A_u| - k$ through the $|A_u|$ most vital links have been found. This addition to the algorithm has proved useful in the examples that have been tried to date.

If the algorithm terminates with $|A_u| = n$, then from Theorem 3.2 the n most vital links have been found. If the algorithm terminates with $|A_u| < n$, then there is a "gap" at the value of interest (i.e., the algorithm obtains some sets of most vital links but not the n most vital). This problem will be treated in detail in Chapter 4. The algorithm clearly terminates in a finite number of steps, since for $u = \max c(a_i) + 1$ for all $a_i \in A$, $|A_u| = 0$ (for every minimum cut of $G^u(N,A)$). It should be noted that for $u = 1$ all arcs in A^u are at capacity $u = 1$. Hence, at least for this value the algorithm finds a set of most vital links. In particular, it finds the fewest arcs necessary to disconnect the network. Also, from Theorem 3.3, $|A_u|$ is non-increasing; hence, if $|A_u| < n$, there is no point in investigating a larger value of u.

Finally, observe that the maximum flow pattern found for some $G^u(N,A)$ is feasible for $G^{u+1}(N,A)$ and can be used as an initial flow in finding the maximum flow in $G^{u+1}(N,A)$.

An Example Using the Modified Cut Method

The modified cut method will be illustrated by determining the two most vital links in Example Network 3.1 as defined in Figure 3.1.

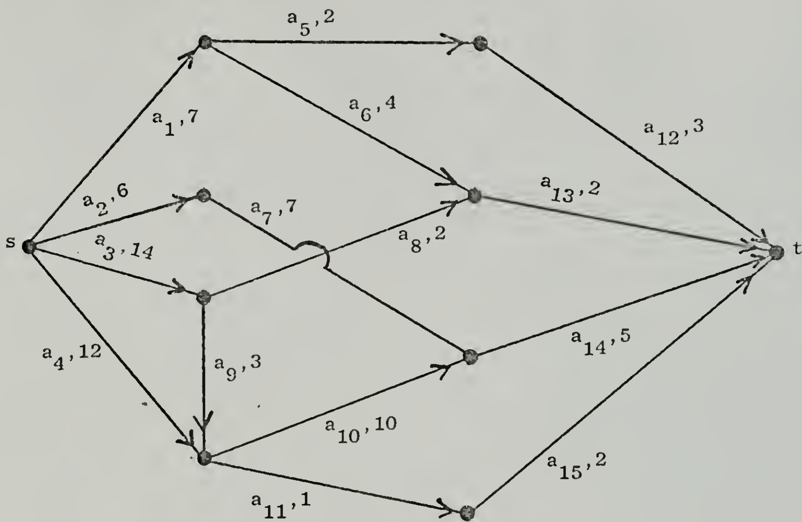

Figure 3.1. Example Network 3.1.

Note that $G^1(N,A)$ is defined as in Figure 3.2.

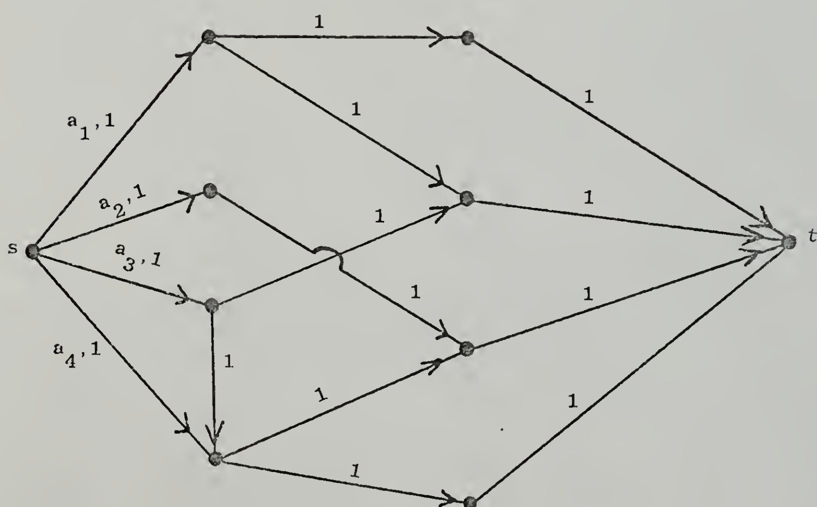

Figure 3.2. $G^1(N,A)$ of Example Network 3.1.

The minimum cut value of $G^1(N,A)$ is 4 with $A^1 = \{a_1, a_2, a_3, a_4\}$ a minimum cut of $G^1(N,A)$ and $|A_1| = 4$.

For $G^2(N,A)$ (see Figure 3.3), the desired minimum cut is $A^2 = \{a_{11}, a_{12}, a_{13}, a_{14}\}$ and $|A_2| = 3$. The minimum cut value of $G^2(N,A)$ is 7.

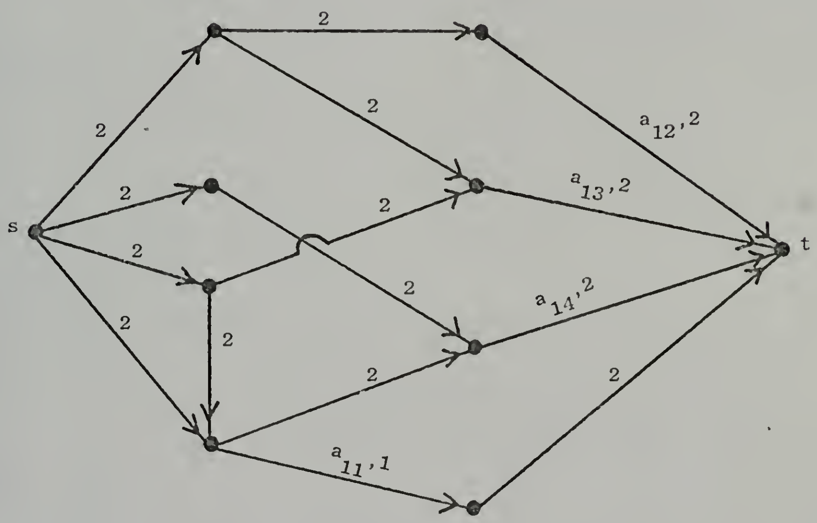

Figure 3.3. $G^2(N,A)$ of Example Network 3.1.

For $G^3(N,A)$, as shown in Figure 3.4, the desired minimum cut is again $A^3 = \{a_{11}, a_{12}, a_{13}, a_{14}\}$, $|A_3| = 2$ and the minimum cut value is 9. Thus, a_{12} and a_{14} are the two most vital links and the problem is solved. It may be noted that if the result of Corollary 3.2.2 were invoked at $G^2(N,A)$, the same conclusions could have been made, since a_{13} was in the minimum cut of $G^2(N,A)$; $|A_2| = 3$; and, $c^\infty(a_{13}) = 2$.

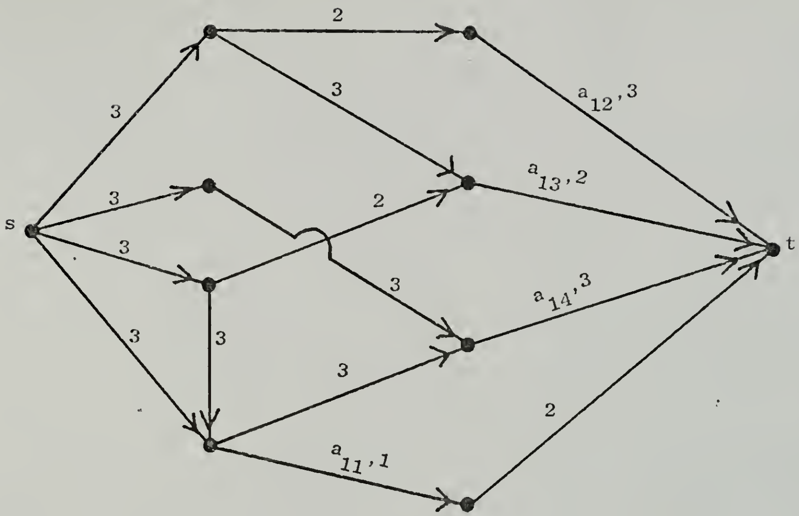

Figure 3.4. $G^3(N,A)$ of Example Network 3.1.

An Example Where a Gap Occurs

In stating the modified cut method it was noted that the algorithm may end in a gap, that is, $|A_u| > \alpha$ and $|A_u+1| < \alpha$. In Chapter 5 the frequency of occurrence of these gaps in the sample problems to date is noted and discussed. In this section a network will be used for which the modified cut method fails to find the n most vital links.

Consider the Example Network 3.2 as defined in Figure 3.5 and use the modified cut method to find the two most vital links of that network.

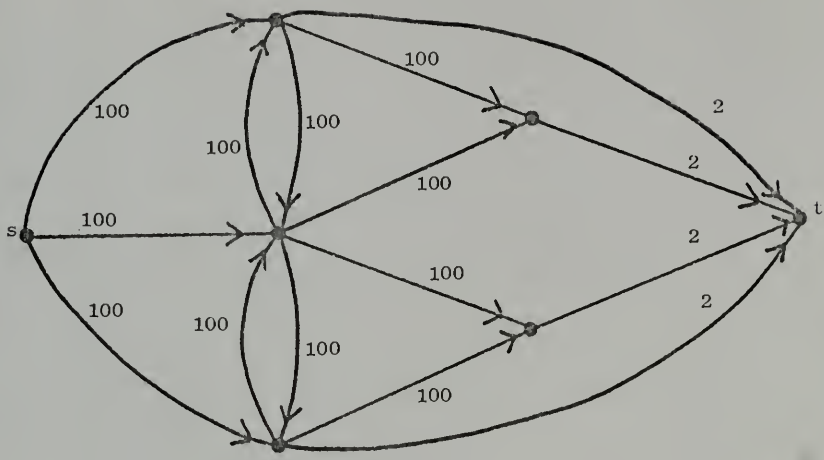

Figure 3.5. Example Network 3.2.

Note that for $G^1(N,A)$ the arcs leaving node s form the minimum cut. (See Figure 3.6).

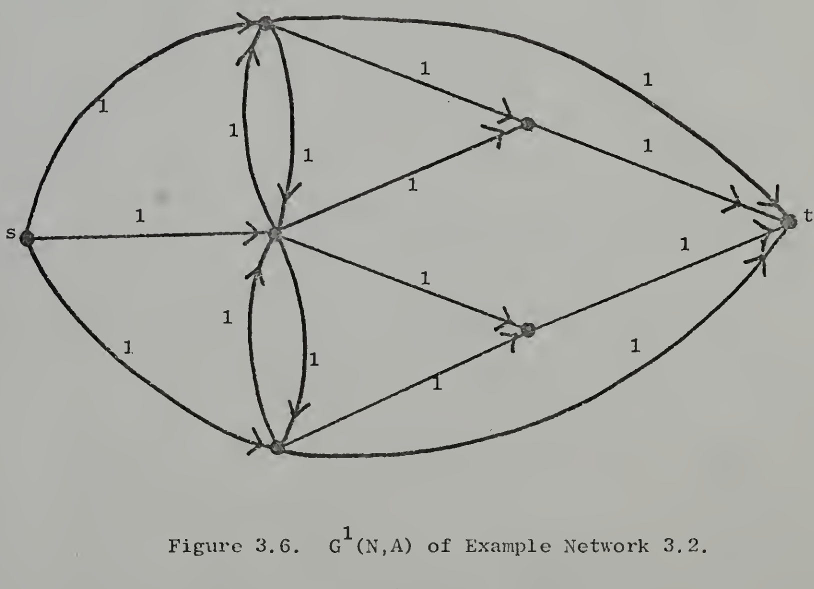

Figure 3.6. $G^1(N,A)$ of Example Network 3.2.

Thus, the minimum cut value of $G^1(N,A)$ is 3 and $|A_u| = 3$.

$G^2(N,A)$ is noted in Figure 3.7. Again the arcs leaving node s form the minimum cut, $|A_2| = 3$, and the minimum cut value is 6.

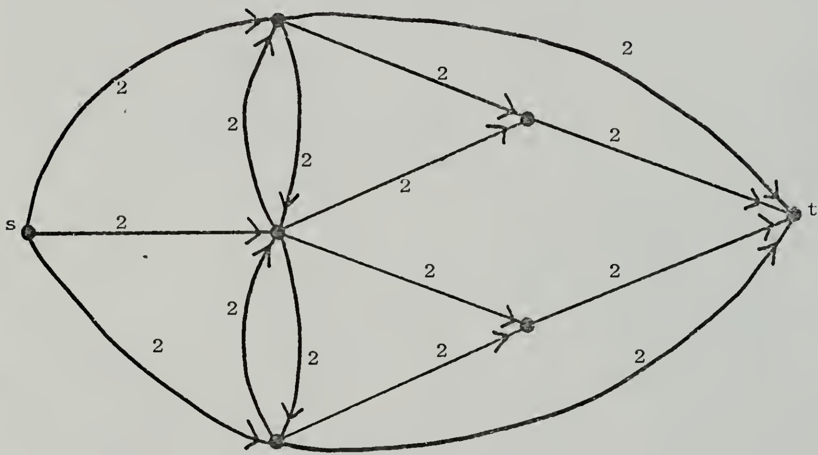

Figure 3.7. $G^2(N,A)$ of Example Network 3.2.

The arcs that enter node t form a minimum cut for $G^3(N,A)$, as shown in Figure 3.8. This minimum cut has a cut value of 8. Each of the arcs in this cut has a capacity of 2 in the original network, so $|A_3| = 0$, and a gap exists at the desired problem.

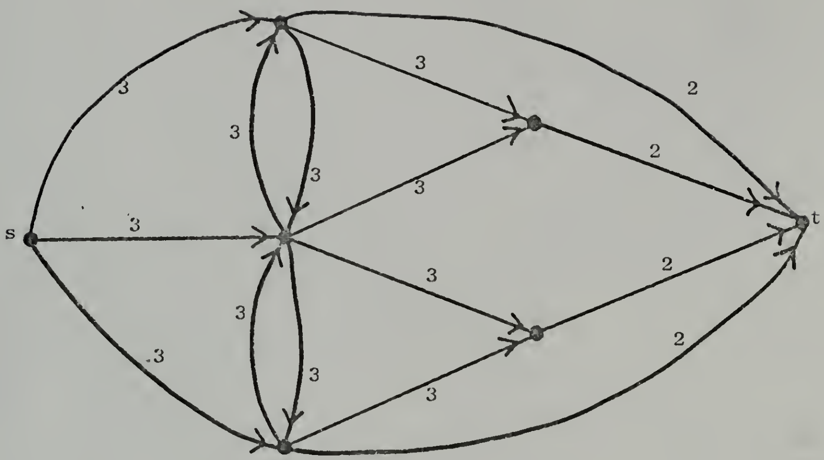

Figure 3.8. $G^3(N,A)$ of Example Network 3.2.

In Chapter 4 the gap problem will be treated and the resulting algorithm will be applied to Example Network 3.2 given in this chapter.

CHAPTER 4

RESOLUTION OF THE GAP PROBLEM

Introduction

In this chapter the gap problem introduced in Chapter 3 is addressed. As shown in Example 3.2, at termination the modified cut method may not determine the desired n most vital links, i.e., the method may end in a gap. Results will now be developed and an algorithm presented which will resolve a gap if it occurs.

It will be assumed in this chapter that the arcs of a network are ordered such that if a_i and a_j are arbitrary arcs of G(N,A) and if $i < j$, then $c(a_i) \geq c(a_j)$. This leads to no loss of generality, since, if the condition does not hold in the original network, the arcs can be renumbered without changing the network. Also, since in what follows only disconnecting sets between the source node s and sink node t are of interest, the term disconnecting set will imply a disconnecting set between s and t. If F is a disconnecting set, then $C(F - F_n)$ will be called the n-reduced value of F, where F_n denotes the n largest arcs of F. Finally, if $C(F-F_n) \leq C(H-H_n)$ for all disconnecting sets in some set $\mathcal{H}$, then F will be called the weakest n-disconnecting set of $\mathcal{H}$.

Some Preliminary Results

Recall that in Chapter 3 the n most vital links were shown to be the n largest arcs A_n^* of a disconnecting set‡ A^* which satisfies: $C(A^*-A_n^*) \leq C(A'-A_n')$ for all disconnecting sets A' in G(N,A). Thus, solving for the optimal solution of this disconnecting set problem determines the n most vital links. Initially, every disconnecting set of G(N,A) is a candidate for the optimal solution to the most vital links cut problem. The following theorem can be used to eliminate from further consideration many of the disconnecting sets of G(N,A).

Theorem 4.1:

Let a_i, a_j be arcs of G(N,A) such that $c(a_i) \geq c(a_j)$, and define the set $\mathcal{H}$ to be:

$$\mathcal{H} \equiv \{H \mid H \text{ is a disconnecting set of } G(N,A) \text{ and } a_i \in H_n\}.$$

If H^* is a disconnecting set of $\mathcal{H}$ which satisfies:

$$C(H^*-H_n^*) \leq C(H-H_n) \text{ for all disconnecting sets } H \in \mathcal{H}$$

and if there exists a disconnecting set F such that $a_j \in F_n$ and

$$C(F-F_n) < C(H^*-H_n^*), \text{ then } a_i \notin F.$$

‡Actually only cuts were considered in Chapter 3, but since the weakest n-disconnecting set of G(N,A) is a cut and every cut is a disconnecting set, this generalization follows.

Proof:

Given that $a_j \in F_n$, there are three possible cases

Case 1 -- $a_j \in F_n$ and $a_i \in F_n$

Case 2 -- $a_j \in F_n$ and $a_i \in F-F_n$

Case 3 -- $a_j \in F_n$ and $a_i \notin F$.

If Case 1 holds, then $F \in \mathcal{H}$, but by the hypothesis $C(H^*-H_n^*) \leq C(H-H_n)$ for all $H \in \mathcal{H}$ so this case cannot occur. Case 2 is not possible, since, if $c(a_i) > c(a_j)$, then $a_i \in F-F_n$ implies that $a_j \in F-F_n$, and if $c(a_i) = c(a_j)$, then $F \in \mathcal{H}$ and the comments made about Case 1 are valid and lead to a contradiction for this case. Thus, only Case 3 is possible and the result follows.

Q. E. D.

Corollary 4.1.1:

Given the network $G(N,A)$, let $\Gamma = \{a_1, a_2, \ldots, a_p\}$ be a set of arcs such that $\Gamma \subseteq A$ and let $a_k \in A$ where $k > p$ (i.e., $c(a_k) \leq c(a_i)$ for $i = 1,2,\ldots,p$). If among all disconnecting sets H which satisfy $H_n \cap \Gamma \neq \Phi$, H^* is the weakest n-disconnecting set and if there exists a disconnecting set F for which $a_k \in F_n$ and $C(F-F_n) < C(H^*-H_n^*)$, then $F \cap \Gamma = \Phi$.

Proof:

Since $c(a_i) \geq c(a_k)$ for $i = 1,2,\ldots,p$, the result follows from Theorem 4.1.

Let $\mathcal{H}$ and $\mathcal{H}^*$ be sets defined by:

$$\mathcal{H} = \{H \mid H \text{ is a disconnecting set of } G(N,A) \text{ and } a_1 \in H_n\}$$

$$\mathcal{H}^* = \{H^* \mid H^* \text{ is a disconnecting set of } G(N,A) \text{ and } a_1 \notin H^*\}.$$

If the weakest n-disconnecting sets in both $\mathcal{H}$ and $\mathcal{H}^*$ are found, then, by Theorem 4.1, the n most vital links of G(N,A) are determined. The problem of determining these arc sets will now be investigated.

The Extended Modified Cut Method

Observe that in defining the modified cut method it was only necessary to find the weakest n-cut of G(N,A), i.e., only cuts of G(N,A) and not all disconnecting sets were considered. In this section an extended version of this method will be employed, and it will be necessary to consider disconnecting sets rather than cuts of G(N,A). That is, for a fixed set of arcs Λ the n-$|\Lambda|$ most vital links will be found in G(N,A-Λ). Thus, while the weakest (n-$|\Lambda|$)-disconnecting set of G(N,A-Λ) will be a cut of G(N,A-Λ), the set together with Λ may only be a disconnecting set of G(N,A). This will cause no difficulty, however, as all the results which have been stated are valid for both disconnecting sets and cuts.

Observe that the weakest n-disconnecting set of the set $\mathcal{H}$ may be characterized as the set whose n largest arcs consist of a_1 and the n-1 best arcs to remove from G(N,A), given a_1 is to be removed. But, this is exactly the disconnecting set which would be found by setting $\Lambda = a_1$ and solving for the n-$|\Lambda|$ most vital links of G(N,A-Λ). Thus, the modified cut method can be used to find the weakest n-disconnecting set of $\mathscr{U}$. Assuming that the desired set of $\mathscr{U}$ is found (the case where a gap occurs in this problem will be treated later in this section), the set $\mathscr{U}^*$ will now be considered, and it will be demonstrated that the modified cut method can also be extended to find the weakest n-disconnecting set among those sets which do not contain arcs from some fixed set Γ. (Thus, for $\Gamma = a_1$, the method to be presented will locate the weakest n-disconnecting set of $\mathscr{U}^*$.)

It is assumed in what follows that there exist disconnecting sets in G(N,A) which do not contain arcs from Γ, since, if this is not the case, then in the context of the last section $\mathscr{U}^* = \Phi$.

Given the fixed set of arcs Γ and G(N,A), let $\mathscr{B}^u(N,A)$ be G(N,A) with a capacity function C^u defined by:

$$C^u(a_i) = \begin{cases} \infty & \text{if} \quad a_i \in \Gamma \\ u & \text{if} \quad a_i \notin \Gamma \quad \text{and} \quad c(a_i) \geq u \\ c(a_i) & \text{if} \quad a_i \notin \Gamma \quad \text{and} \quad c(a_i) < u \ . \end{cases}$$

An arc a_i will be called a u-ceiling arc of $c(a_i) = u$ in $\mathscr{B}^u(N,A)$.

<u>Theorem 4.2:</u>

If a minimum cut A^* of $\mathscr{B}^u(N,A)$ contains α u-ceiling arcs, then $C(A^*-A_n^*) \leq C(B-B_n^*)$ for all cuts B such that $B \cap \Gamma = \Phi$.

Proof:

A^* is a minimum cut of $\mathscr{G}^u(N,A)$, therefore $C^u(A^*) \leq C^u(B)$ for all cuts B. Also, for all cuts B that satisfy $B \cap \Gamma = \Phi$, $C^u(B_\alpha) \leq u\alpha$. But, by the hypothesis $C^u(A^*_\alpha) = u\alpha$; therefore,

$$C^u(A^*-A^*_\alpha) = C^u(A^*) - u\alpha \leq C^u(B) - u\alpha \leq C^u(B-B_\alpha) \qquad (1)$$

for all cuts B such that $B \cap \Gamma = \Phi$.

Next, observe that for cuts B which satisfy $B \cap \Gamma = \Phi$, $C(B-B_\alpha) \geq C^u(B-B_\alpha)$ and that, since A^* contains exactly α u-ceiling arcs, $C(A^*-A^*_\alpha) = C^u(A^*-A^*_\alpha)$. Combining these last conclusions with (1) gives the desired result.

Q. E. D.

Recalling Theorem 3.3, which is valid for the $\mathscr{G}^u(N,A)$ as well as the $G^u(N,A)$ networks and Theorem 4.2, the extended modified cut method may be stated as:

STEP 0:

Note the fixed set Γ.

Let $u = 1$ and go to STEP 1.

STEP 1:

$$\text{Set } c^u(a_i) = \begin{cases} \infty & \text{if } a_i \in \Gamma \\ c^\infty(a_i) & \text{if } c^\infty(a_i) < u, \text{ and } a_i \notin \Gamma \\ u & \text{if } c^\infty(a_i) \geq u, \text{ and } a_i \notin \Gamma \end{cases}$$

Solve for the minimum cut of $\mathcal{G}^u(N,A)$; if the minimum cut value is not finite stop (all cuts contain arcs of Γ), otherwise note A^u, A_u, and $|A_u|$. Go to STEP 2.

STEP 2:

If $|A_u| > n$ let $u = u+1$. Go to STEP 1.

If $|A_u| = n$ terminate as the n most vital links are the arcs contained in A_u.

If $|A_u| < n$ terminate, a gap has occurred at the n most vital links problem.

Finally, the problem of resolving gaps that arise in using the extended modified cut method to find the weakest n-disconnecting set in either $\mathcal{N}$ or $\mathcal{N}^*$ must be addressed. Suppose a gap occurs in solving for the $n-|\Lambda|$ most vital links in $G(N,A-\Lambda)$ where the disconnecting sets containing arcs from some fixed set Γ are not to be considered. This is a well-defined problem exactly like that posed at the beginning of this chapter; therefore, the extended modified cut method can again be employed as follows: place the largest arc in $A - \Lambda - \Gamma$, say a_p, into the set Λ and solve for the weakest $(n-|\Lambda|)$-disconnecting set in both of the following sets:

$$\mathcal{H} = \{H \mid H \text{ is a disconnecting set of } G(N,A-\Lambda),\ \Gamma \cap H = \Phi, \text{ and, } a_p \in H_n\}$$

and

$$\mathcal{H}^* = \{H^* \mid H^* \text{ is a disconnecting set of } G(N,A-\Lambda),\ \Gamma \cap H = \Phi, \text{ and, } a_p \notin H_n^*\}.$$

The General Algorithm

The general algorithm consists of 3 routines. Routine A is the extended modified cut method with the set Γ of infinite capacitied arcs and a set Λ of arcs to be removed from G(N,A) (i.e., the network $G(N,A-\Lambda)$ is considered). Routine B is employed when Routine A ends in a success; i.e., no gap occurred or when all the disconnecting sets have been considered. Routine C is used to define a new set of sub-problems if a gap occurs.

It is assumed at the outset that Λ and Γ are both empty. Initially the current best solution value, denoted by V_o, is set to infinity or to the value of the best solution known.

The General Algorithm can be stated as:

ROUTINE A

STEP 0:

If $n - |\Lambda| = 0$ let $u = \infty$, otherwise let $u = 1$.

STEP 1:

$$\text{Let } c^u(a) = \begin{cases} c(a) & \text{if } c(a) < u, \text{ and } a \notin \Gamma \\ \infty & \text{if } a \in \Gamma \\ u & \text{if } c(a) \geq u, \text{ and } a \notin \Gamma \end{cases}$$

Step 2:

Find a minimum disconnecting set A^* in $\mathscr{G}^u(N, A-\Lambda)$. Let $U = \{a \mid a \in A^* \text{ and } c(a) = u\}$.

STEP 3:

If $c^u(A^*) = \infty$, go to STEP 2, ROUTINE B.

STEP 4:

a. If $|U| > n-|\Lambda|$, let $u = u+1$ and go to STEP 1.

b. If $|U| = n-|\Lambda|$, go to STEP 1, ROUTINE B.

c. If $|U| < n-|\Lambda|$, go to ROUTINE C.

ROUTINE B

STEP 1:

If $C(A^*-U) < V_o$ let $A^o = A^*$ and $V_o = C(A^*-U)$.

STEP 2:

If $\Lambda = \Phi$ terminate, A^o is the optimal solution. If $\Lambda \neq \Phi$ continue.

STEP 3:

Let $\Lambda = \Lambda - \{a_p\}$ where $p \geq k$ for all $a_k \in \Lambda$ (i.e., a_p is the smallest capacity arc in Λ).

Let $\Gamma = \Gamma \cup \{a_p\} - (\Gamma \cap \{a_{p+1}, a_{p+2}, \ldots, a_m\})$ (i.e., add a_p to Γ and remove from Γ all arcs having indices greater than p). Go to ROUTINE A.

ROUTINE C

STEP 1:

If $A - \Lambda \cup \Gamma = \Phi$ go to STEP 2, ROUTINE B.

STEP 2:

Let $\Lambda = \Lambda \cup \{a_p\}$ where $p \leq k$ for all $a_k \in A-\Lambda \cup \Gamma$ (i.e., a_p is the largest capacity arc not in Λ or Γ). Go to ROUTINE A.

An Improved General Algorithm

In this section results which improve the computational efficiency of the general algorithm are discussed and an improved general algorithm is developed.

Lemma 4.1:

If a cut B^* is optimal for the $n-k$ but not the n most vital links problem of G(N,A) and if the cut H^* is optimal for the n most vital links problem, then $C(H^*_n - H^*_k) > C(B^*_n - B^*_k)$.

Proof:

Note, by the hypothesis of Lemma 4.1,

$$C(B^*-B^*_k) \leq C(H^*-H^*_k) \tag{2}$$

and

$$C(B^*-B^*_n) > C(H^*-H^*_n). \tag{3}$$

Combining (2) and (3) yields:

$$C(B^*-B^*_k) - C(B^*-B^*_n) < C(H^*-H^*_k) - C(H-H^*_n) \tag{4}$$

But (4) reduces to $C(B^*_n-B^*_k) < C(H^*_n-H^*_k)$ which is the desired result.

Q. E. D.

Lemma 4.1 is most useful for the case where $k=1$; that is, consider the situation where, in using the modified cut method, a gap occurs at the n but not at the $n-1$ most vital links problem. If the cut B^* is found to be optimal for this latter problem and the n^{th} largest arc of B^* is a_x, then cuts whose n^{th} largest arc is no larger than a_x do not have a smaller n-reduced value than that of B^* (the optimal cut of the $n-1$ most vital links problem).

The final result of this chapter concerns a lower bound on the optimal solution of the α most vital links problem which is found as a by-product of the modified cut method.

<u>Lemma 4.2</u>:

If B^* is a minimum cut of $G^u(N,A)$, then the optimal cut, A^*, to the α most vital links problem satisfies:

$$C(A^* - A^*_\alpha) \geq C^u(B^*) - u\alpha .$$

Proof:

Verification of this result only requires the observation that if B^* contains α u-ceiling arcs, then B^* is the optimal cut for the α most vital links problem (Theorem 3.2) and $C(B^* - B^*_\alpha) = C^u(B^*) - u\alpha$. Thus, the optimal solution A^* can be no better than $C^u(B^*) - u\alpha$.

<u>Q. E. D.</u>

Note that Lemma 4.2 also holds for any $\mathcal{G}^u(N,A)$ where one is considering all cuts but those which contain arcs from a set Γ (this follows, since Theorem 4.2 and Theorem 3.2 prove parallel results for $\mathcal{G}^u(N,A)$ and $G^u(N,A)$, respectively).

Although given gap occurs, the bound found in Lemma 4.2 is always a strict lower bound, the last result can be employed in two ways. The first use is utilizing the bound as a stopping rule, i.e., if the current best solution is "close" to the lower bound, terminate.

The second use is as follows: suppose a gap occurs at some subproblem. If the lower bound on the optimal of this subproblem is no better than a solution which already has been found, then the subproblem need not be considered any further, even if it ended in a gap.

Since the results of this section apply to Routine C, an improved Routine C will now be noted. It is assumed that $c(a_x)$ will be set to $c(a_m)$ if the $n-1$ most vital links are not found when solving for the n most vital links in $G(N,A)$; otherwise, it will represent the capacity of the n^{th} largest arc of the optimal cut to this latter problem. To use the improved algorithm the best lower bound (BLB) of the current subproblem as well as the current best solution and the current best solution's value (CBSV) must be maintained.

The Improved Routine C becomes:

STEP 1:

If BLB $\geq$ CBSV, go to STEP 2, ROUTINE B.

STEP 2:

Let $\Lambda = \Lambda \cup \{a_p\}$ where $p \leq k$ for all $a_k \in A - \Lambda \cup \Gamma$.

If $c(a_q) \leq c(a_x)$ for any $a_q \in \Lambda$, go to STEP 2, ROUTINE B, otherwise, go to ROUTINE A.

An Example

Recall that in Chapter 3 it was established that a gap occurs in solving for the two most vital links of Example Network 4.1, defined by Figure 4.1. The Improved General Algorithm will now be applied to this network to resolve this gap and locate the two most vital links.

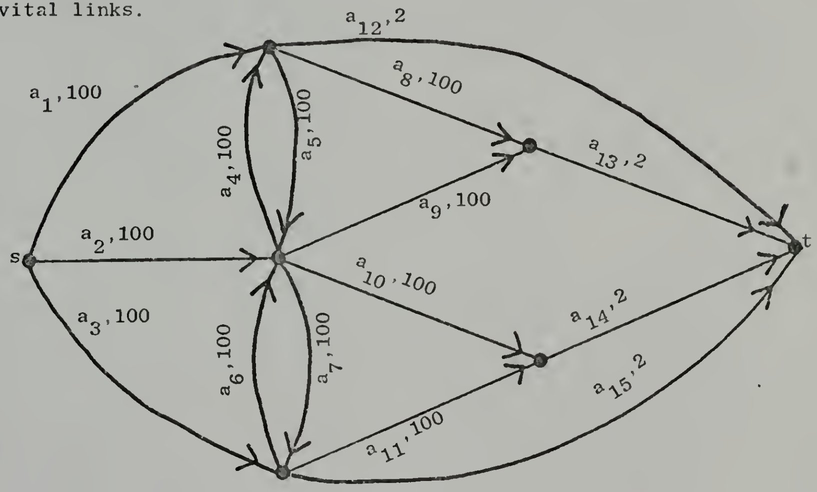

Figure 4.1. Example Network 4.1.

The algorithm starts in Routine A with $\Gamma = \Phi$, $\Lambda = \Phi$ (i.e., the original network G(N,A)). As noted in Chapter 3, the modified cut method ends in a gap with the following minimal cuts of $G^u(N,A)$, $u = 1,2,3,\ldots,$ being located $A^1 = A^2 = \{a_1, a_2, a_3\}$ and $A^3 = \{a_{12}, a_{13}, a_{14}, a_{15}\}$. The best lower bound is found at G^2 or G^3 and has value $C^3(A^3) - (2)(3) = 8 - 6 = 2$. The current best solution is the cut $\{a_{12}, a_{13}, a_{14}, a_{15}\}$ with a 2-reduced cut value of 4.

Since a gap occurred, Routine C is employed to define the network depicted in Figure 4.2.

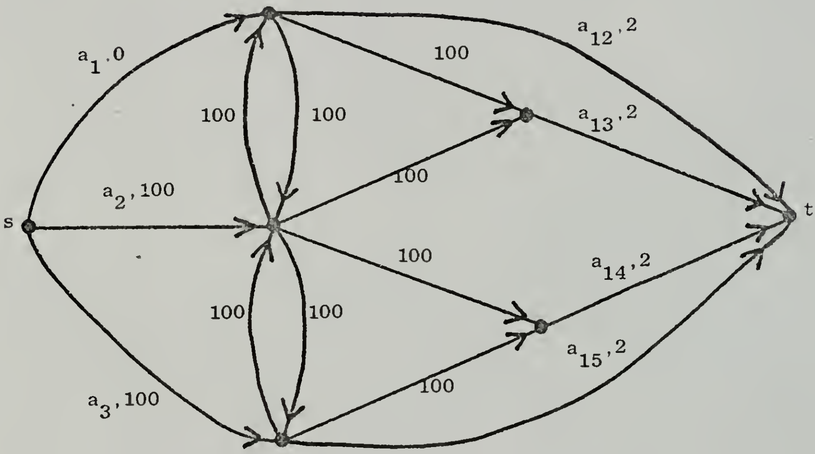

Figure 4.2. The First Subnetwork of Example Network 4.1.

Upon definition of the $G(N,A-a_1)$, Routine A is again employed to locate the one most vital link of this network. In this problem $A^1 = A^2 = A^3 = \{a_2, a_3\}$ and $A^4 = \{a_{12}, a_{13}, a_{14}, a_{15}\}$. Thus a gap occurs and the current best solution value remains 4. The best lower bound of this subproblem has a value of 4 and is found at $\mathscr{B}^3(N,A)$. Therefore, since the lower bound of the subproblem is not better than the best current solution, the subproblem does not have to be considered and Routine B is called.

Upon applying Routine B, the network shown in Figure 4.3 is realized.

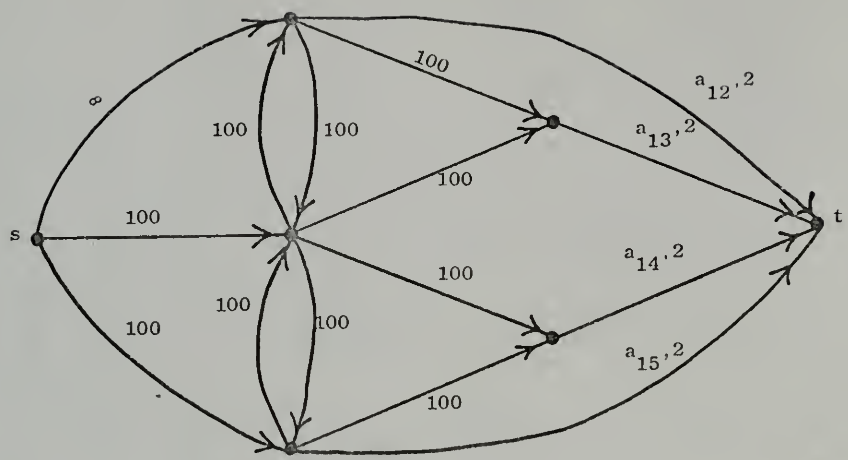

Figure 4.3. The Second Subnetwork of Example Network 4.1.

For this problem the cut $\{a_{12}, a_{13}, a_{14}, a_{15}\}$ is the minimum cut for every $\mathcal{G}^u$. Thus, at $\mathcal{G}^2$, the two most vital links problem is solved for the above network and Routine B is called.

Since $\Lambda = \Phi$, the algorithm terminates with the current best solution being declared the optimal solution (i.e., any pair of arcs from the set $\{a_{12}, a_{13}, a_{14}, a_{15}\}$ are the two most vital links of this network).

CHAPTER 5

COMPUTATIONAL RESULTS AND CONCLUSIONS

The modified cut method of Chapter 3 and the improved general algorithm of Chapter 4 were coded in Fortran IV and problems were solved using an IBM 360/65 computer. These algorithms for a given network require approximately the same amount of core space that the maximum flow algorithm of Ford and Fulkerson [12] would require‡ to solve the maximum flow-minimum cut problem for the same network. For the example networks referred to in this chapter, the capacity of the arcs ranged from 1 to 50.‡‡

Several basic network configurations were used to test the algorithms and, as can be seen in Table 5.1, the test problems ranged in size from small (14 arcs, 6 nodes) to larger networks (250 arcs, 50 nodes). Also, since changing the capacity function of a network affects the solution of the n most vital links problem, several of the example networks have the same structures with different capacity functions.

---

‡Actually the improved general algorithm which contains the modified cut method requires four $|A|$- component vectors more than the maximum flow algorithm.

‡‡Naturally, if the range of the capacity values were significantly larger than 50, then, for a given large problem, one might expect slightly longer execution times than are cited in this chapter.

The results of applying the modified cut method to the example problems are presented in Table 5.1. The symbol η which has been used to indicate the smallest number of arcs in any cut between the source and sink nodes in G(N,A) also represents the maximum number of most vital links problems that could be of interest for a given network, i.e., n = 1,2,...,η. Those values of n for which a gap occurred are noted in the last column of Table 5.1 for each test problem. Observe that for the sample problems which were solved, about 5 percent of the possible most vital links problems ended in a gap.

In Table 5.2 the results of applying the improved general algorithm to those gaps noted in Table 5.1 are presented. Since the number of times the modified cut method (Routine A) is employed is a key factor in determining the efficiency of the improved general algorithm, this statistic is included in column 6 of the Table 5.2.

TABLE 5.1

RESULTS OF APPLYING MODIFIED CUT METHOD TO SAMPLE NETWORKS

Problem Number	$\|A\|$	$\|N\|$	η	Execution Time (minutes)	Values of n for Which Gaps Occurred
1	28	12	4	.03	-
2	28	12	4	.03	-
3	28	12	4	.03	-
4	35	16	5	.04	-
5	35	16	5	.03	-
6	35	16	5	.04	-
7	14	6	4	.03	-
8	14	6	4	.02	-
9	14	6	4	.03	-
10	52	17	8	.07	2
11	52	17	8	.06	-
12	52	17	8	.11	4
13	41	16	8	.06	4,5
14	41	16	8	.05	-
15	41	16	8	.07	-
16	41	12	7	.05	5
17	41	12	7	.03	-
18	41	12	7	.04	-
19	48	16	7	.04	-
20	48	16	7	.05	-
21	48	16	7	.08	-

TABLE 5.1 (Continued)

Problem Number	$\|A\|$	$\|N\|$	η	Execution Time (minutes)	Values of n for Which Gaps Occurred
22	26	9	7	.05	-
23	26	9	7	.03	-
24	26	9	7	.05	-
25	57	20	7	.03	-
26	57	20	7	.05	-
27	57	20	7	.05	-
28	35	11	8	.04	-
2	35	11	8	.05	-
30	35	11	8	.05	4,6
31	15	7	3	.02	1,2
32	250	50	25	.30	-
33	250	50	25	.38	6,18,19
34	250	50	25	.35	-
35	250	50	25	.43	6,7,8,9,12,13
36	250	50	25	.36	4,14
37	250	50	25	.26	-
38	250	50	25	.50	7,8,9,10,11
39	250	50	25	.27	-
40	250	50	25	.29	-

TABLE 5.2

RESULTS OF APPLYING IMPROVED GENERAL ALGORITHM TO GAPS THAT OCCURRED IN SAMPLE NETWORKS

Problem Number in Table 5.1	$\lvert A \rvert$	$\lvert N \rvert$	Value of n Solved for	Execution Time (minutes)	Times Modified Cut Method Employed
10	52	17	2	.48	31
12	52	17	4	.12	3
13	41	16	4	.09	3
13	41	16	5	.09	3
16	41	12	5	.07	5
30	35	10	4	.16	11
30	35	10	6	.08	3
31	15	7	1	.02	3
31	15	7	2	.02	3
33	250	50	6	2.72	14
33	250	50	18	1.10	5
33	250	50	19	1.06	.5
35	250	50	6	2.28	7
35	250	50	7	2.50	7
35	250	50	8	2.46	7
35	250	50	9	2.46	8
35	250	50	12	.92	3
35	250	50	13	.92	3
36	250	50	4	8.50	61
36	250	50	14	.69	3
38	250	50	7	12.15	43
38	250	50	8	12.20	43
38	250	50	9	12.20	43
38	250	50	10	12.56	43
38	250	50	11	12.04	43

Conclusions

To find the n most vital links of an arbitrary network by enumeration would involve solving either $\binom{|A|}{n}$ maximum flow problems (one for each possible set of n arcs of A) or locating all the cuts between the source and sink nodes. As noted in this chapter for the problems investigated to date, a gap occurred for less than 5 percent of the possible n-values. Thus, 95 percent of the cases required one use of the modified cut method (which is effectively the same as solving a maximum flow problem) and the solution found included a family of most vital links of which the n-value of interest was a member. Furthermore, even in situations where gaps occurred, on the average it has only been required to employ the modified cut method sixteen times per gap problem. This is approximately equivalent to sixteen maximum flow problems.

Next, recall that two bounds have been identified (Lemma 4.1 and Lemma 4.2) in connection with the improved general algorithm. While each of these has been employed in particular problems, the lower bound on the value of the optimal solutions (Lemma 4.2) has proved to be much stronger than the bound noted in Lemma 4.1. However, since both bounds are easy to find and maintain, it is suggested that both be used when applying the methods developed in this paper.

In conclusion, while the above observations were made after considering only a small sample of problems, the partitioning scheme of the improved general algorithm, while similar to implicit enumeration (e.g., Balas [1], Geoffrion [14]), is much more efficient than the latter approach. This is due to the infrequency of gaps occurring; the bounds which truncate subproblems; and the fact that each time an arc is placed in Γ whole classes of disconnecting sets are eliminated from further consideration.

APPENDIX A

FORMULATIONS OF THE n MOST VITAL LINKS PROBLEM

A Linear Formulation

In Chapter 3, it was demonstrated that the n most vital links problem could be posed as a cut problem where the optimal cut A^* satisfies

$$C(A^* - A_n^*) = \min \quad C(B - B_n) \qquad \text{for all cuts } B \tag{A.1}$$

and A_n^* contains the n most vital links. This cut problem will now be formulated as a linear zero-one integer programming problem. First observe that, instead of only considering cuts in equation (A.1), the B's could represent all disconnecting sets separating the source and sink. This is valid, since the weakest disconnecting set will always be a cut.

The minimum cut problem as formulated by Ford and Fulkerson [11] may be posed as a zero-one integer programming problem, i.e.,

$$\text{Min} \sum_{(i,j)} c(i,j)\ \gamma(i,j)$$

$$\text{such that} \quad -\pi(s) + \pi(t) \geq 1$$

I

$$\pi(i) - \pi(j) + \gamma(i,j) \geq 0 \qquad (i,j) \in A$$

$$\pi(i),\ \gamma(i,j) = 0,1 \qquad (i,j) \in A \text{ and } (i) \in N$$

Furthermore, a solution is feasible for I, if and only if, the set of arcs for which $\gamma(i,j) = 1$ forms a disconnecting set of the given network.

Now define a new set of zero-one variables $\underline{u}$ such that for each arc (i,j) there is a variable $u(i,j)$. These $u(i,j)$'s are restricted to satisfy $u(i,j) \leq \gamma(i,j)$. Thus, for any solution only $u(i,j)$'s corresponding to arcs in the identified disconnecting set may be non-zero. Adding the constraint $\sum_{(i,j)} u(i,j) \leq n$ insures that no more than n of the $u(i,j)$ variables will be non-zero for any feasible solution of the problem to be defined. The n most vital links problem can then be formulated as:

$$\text{Min} \sum_{(i,j)} c(i,j)\,\gamma(i,j) - \sum_{(i,j)} c(i,j)\,u(i,j)$$

such that

$$\begin{array}{llll} & -\pi(s) + \pi(t) \geq 1 & (i,j) \in A & \\ \text{II} & \pi(i) - \pi(j) + \gamma(i,j) \geq 0 & (i,j) \in A & \\ & \sum_{(i,j)} u(i,j) \leq n & (i,j) \in A & \\ & u(i,j),\ \gamma(i,j),\ \pi(i) = 0,1 & (i,j) \in A & \text{and } (i) \in N \\ & u(i,j) - \gamma(i,j) \leq 0 & (i,j) \in A & \end{array}$$

Note that only disconnecting sets, B, are feasible for II and for each such B the optimal $u(i,j)$ values are one if $(i,j) \in B_n$ and zero otherwise. Thus, the optimal solution to II is that disconnecting set A^* which satisfies equation (A.1).

By making the change of variable $\gamma'(i,j) = \gamma(i,j) - u(i,j)$, problem II becomes:

$$\text{Min} \sum_{(i,j)} c(i,j)\, \gamma'(i,j)$$

such that

$$\text{III} \quad \begin{aligned} -\pi(s) + \pi(t) &\geq 1 & & \\ \pi(i) - \pi(j) + \gamma'(i,j) + u(i,j) &\geq 0 & & (i,j) \in A \\ \sum_{(i,j)} u(i,j) &\leq n & & (i,j) \in A \\ u(i,j),\ \gamma'(i,j),\ \pi(i) &= 0,1 & & (i,j) \in A \text{ and } (i) \in N \end{aligned}$$

A Quadratic Formulation

In this section a zero-one quadratic formulation with special structure is developed for solving the n most vital links problem. Basically, the formulation is Formulation II presented in the last section with the constraints $\gamma(i,j) - u(i,j) \geq 0$ removed from the constraint set. These conditions, i.e., $u(i,j) \leq \gamma(i,j)$, are guaranteed at optimality in the new formulation by introducing a quadratic part to the objective function which is zero if $u(i,j) \leq \gamma(i,j)$ and ∞ if the constraint is violated. The objective function will be shown to be concave and the constraint coefficient matrix unimodular. This structure will allow the optimal solution to this new formulation to be found by ranking the extreme points of a related continuous quadratic programming problem.

Consider the following zero-one quadratic programming problem, where M is some very large positive number.

$$\text{Min} \sum_{(i,j)} c(i,j)\,\gamma(i,j) - \sum_{(i,j)} c(i,j)\,u(i,j)$$

$$+ \sum_{(i,j)} M\left\{1 - \frac{1}{4}\Big(u(i,j) - \gamma(i,j) - 1\Big)^2 - \frac{3}{4}\Big(u(i,j) + \gamma(i,j) - 1\Big)^2\right\}$$

such that

IV

$$-\pi(s) + \pi(t) \geq 1$$

$$\pi(i) - \pi(j) + \gamma(i,j) \geq 0 \qquad (i,j) \in A$$

$$\sum_{(i,j)} u(i,j) \leq n \qquad (i,j) \in A$$

$$u(i,j),\ \gamma(i,j),\ \pi(i) = 0,1 \qquad (i,j) \in A \text{ and } (i) \in N$$

$$\text{Let } f\Big(u(i,j),\gamma(i,j)\Big) \equiv \left\{1 - \frac{1}{4}\Big(u(i,j) - \gamma(i,j) - 1\Big)^2 - \frac{3}{4}\Big(u(i,j) + \gamma(i,j) - 1\Big)^2\right\} M.$$

Note that $f\Big(u(i,j),\gamma(i,j)\Big) = 0$ whenever $\gamma(i,j) \geq u(i,j)$ in IV and $f\Big(u(i,j),\gamma(i,j)\Big) = M$ if $\gamma(i,j) = 0$ and $u(i,j) = 1$. Thus, for M sufficiently large, the optimal solutions to Formulation II of the last section and IV, given above, will be the same and IV will solve the n most vital links problem.

Next, $\sum_{(i,j)} f\Big(u(i,j),\gamma(i,j)\Big)$ will be shown to be a concave function of $\underline{u}$ and $\underline{\gamma}$. Note that $f\Big(u(i,j),\gamma(i,j)\Big)$ is a sum of a constant and two quadratic forms, each of which has a value of less than or equal to zero for all values of $\gamma(i,j)$ and $u(i,j)$; thus, each of the forms are concave. Furthermore, the sums of concave functions are concave so $f\big(u(i,j),\gamma(i,j)\big)$ is concave as is the function $\sum_{(i,j)} f\big(\gamma(i,j),u(i,j)\big)$.

Consider Formulation V which is really a continuous version of Formulation IV.

$$\text{Min} \sum_{(i,j)} c(i,j)\ \gamma(i,j) - \sum_{(i,j)} c(i,j)\ \gamma(i,j) + \sum_{(i,j)} f\big(u(i,j),\gamma(i,j)\big)$$

such that

$$\text{V} \qquad \begin{array}{rl} -\pi(s) + \pi(t) \geq 1 & \\ \pi(i) - \pi(j) + \gamma(i,j) \geq 0 & (i,j) \in A \\ \sum_{(i,j)} u(i,j) \leq n & (i,j) \in A \\ 0 \leq \gamma(i,j) \leq 1 & (i,j) \in A \\ 0 \leq u(i,j) \leq 1 & (i,j) \in A \\ 0 \leq \pi(i) \leq 1 & (i) \in N \end{array}$$

Hadley in [16] notes that when minimizing a concave function over a convex set, there is an optimal to the problem which occurs at an extreme point of the convex set. The convex constraint set of V will now be shown to be unimodular. Thus, since, if this is the case, each extreme point solution will be zero-one, the optimal zero-one solution to IV can be found by ranking the extreme points of the constraint set of V. Murty [23], and Cabot and Francis [8] demonstrate how to rank the extreme points of quadratic programming problems.

It only remains to verify that the coefficient matrix of V has the unimodularity property. The first two sets of constraints of V are just the constraints of the minimum cut problem and it is a known fact that the coefficient matrix associated with the constraint set of the minimum cut problem is unimodular. Thus, by letting the coefficient matrix associated with these constraints be denoted by R the problem reduces to showing that if R is a unimodular matrix, then R' is also unimodular. Where R' is defined in partitioned form as:

$$R' = \left[\begin{array}{c|c} R & 0 \\ \hline \underline{0} & \underline{1} \\ \hline 0 & I_\alpha \end{array}\right]$$

where $\underline{0}$, $\underline{1}$ are row vectors and α is equal to $2|A| + |N|$ for $G(N,A)$

The matrix R' is unimodular if and only if every minor determinate is equal to 0, +1, or -1. To show this result holds for R', the matrices R_1, R_2, R_3 will be defined where $R_1 = [R:0]$, $R_2 = [\underline{0}:\underline{1}]$ and $R_3 = [0:I_\alpha]$.

Let A be an arbitrary minor of R'. If $A \cap R_3 \neq \Phi$, then there is some row, r, of A which contains elements of R_3. By expanding A by this row a matrix, B, satisfying $B \subseteq A$ and $|A| = \pm |B|$ will be found. Also, B will not contain the row r. Likewise, it is possible to expand B by any row which contains elements of R_3 and this expansion process can be continued until for some matrix, C, such that $|A| = \pm |C|$, and $C \cap R_3 = \Phi$.

Next, consider the case where a minor, C, of R' has $C \cap R_3 = \Phi$. If $C \cap R_2 = \Phi$, then $C \subseteq R_1$ and thus $|C| = 0$, +1 or -1, since R_1 is unimodular. If $C \cap R_2 \neq \Phi$, then there are columns of C which contain elements of R_2. By a sequential expansion about this set of columns, it will be possible to equate $|C| = \pm |D|$ where $D \subseteq C$ and $D \cap R_2 = \Phi$. But, $D \subseteq C$ implies that $D \cap R_3 = \Phi$ which in turn implies that $D \subseteq R_1$. Thus, $|C| = \pm |D|$ which has a value of 0, +1, or -1.

The above discussion leads to the following method for showing that if A is a minor of R', then $|A| = 0$, +1, or -1. First A is expanded by any row containing elements of R_3 and the method described above is used to get a matrix B where: $B \subseteq A$, $|A| = \pm |B|$, and $B \cap R_3 = \Phi$. Next, the columns of B are inspected to find if $B \cap R_2 = \Phi$. If this is the case, $B \subseteq R_1$, and the desired result has been found. If $R \cap R_2 \neq \Phi$, then the above expansion scheme will locate a new matrix, C, such that $|A| = \pm |B| = \pm |C|$ and $C \subseteq R_1$. So in either case $|B|$, and thus $|A|$, has a value of 0, +1, or -1.

REFERENCES

1. Balas, E., "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," ORSA, Vol. 13, No. 4, 1965.

2. Bellmore, M., Bennington, G., and Lubore, S., "A Network Isolation Algorithm," MITRE Technical Paper, MTP-30, December, 1969.

3. Bellmore, M., Greenberg, H., and Jarvis, J., "Multi-commodity Disconnecting Sets," Management Science, Vol. 16, No. 6, 1970.

4. Bellmore, M., and Ratliff, H., "Set Covering and Involuntary Bases", to appear in Management Science.

5. Bennington, G. E., "Multi-commodity Disconnecting Sets in Undirected Graphs by Node Partitioning," Ph.D. Dissertation, The Johns Hopkins University, 1969.

6. Busacker, R. G., and Gowen, P. G., "A Procedure for Determining a Family of Minimal-Cost Network Flow Patterns," O.R.O. Technical Paper 15, 1961.

7. Busacker, R. G., and Saaty, T. L., Finite Graphs and Networks, McGraw-Hill, Inc., New York, N. Y., 1965.

8. Cabot, A. V., and Francis, R. L., "Solving Certain Quadratic Minimization Problems by Ranking the Extreme Points," ORSA, Vol. 18, No. 1, 1970.

9. Durbin, E. P., "An Interdiction Model of Highway Transportation," RAND Memorandum, RM-4945 -PR, May, 1966.

10. Everett, R., "Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources," ORSA, Vol. 11, No. 3, 1963.

11. Ford, L. R., and Fulkerson, D. R., Flows in Networks, Princeton University Press, Princeton, New Jersey, 1962.

12. Ford, L. R., and Fulkerson, D. R., "Maximal Flow Through a Network," Canad. J. Math., Vol. 8, 1956, pp. 399-404.

13. Frank, H., and Frisch, I. T., Communication, Transmission, and Transportation Networks, Addison-Wesley, Reading, Mass., 1971.

14. Geoffrion, A. M., "Integer Programming by Implicit Enumeration and Balas' Method," SIAM Review, Vol. 9, No. 2, 1967.

15. Greenberg, H. J., "Optimal Attack of a Command and Control Communications Network," Ph.D. Dissertation, The Johns Hopkins University, 1968.

16. Hadley, G., Nonlinear and Dynamic Programming, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1964.

17. Halder, A. K., "The Method of Competing Links," Transportation Science (ORSA), Vol. 4, No. 1, 1970.

18. Hobbs, A. M., "Programmer's Manual for the Pathfinding Importance Number Generator," U. S. Department of Commerce, National Bureau of Standards, Washington, D. C., 1962.

19. Jarvis, J., "Optimal Attack and Defense of a Command and Control Communications Network," Ph.D. Dissertation, The Johns Hopkins University, 1968.

20. Krokowski, M., "Some Considerations in Assignment of Values to Elements of Communications Network," Working Paper, National Bureau of Standards, Project 4556455, July, 1967.

21. Laughhunn, D. J., "Quadratic Binary Programming with Applications to Capital-Budgeting Problems," ORSA, Vol. 18, No. 3, 1970.

22. McMasters, A. W., and Mustin, T. M., "Optimal Interdiction of a Supply Network," to appear in the Naval Research Logistics Quarterly.

23. Murty, Katta G., "An Algorithm for Ranking all the Assignments in Order of Increasing Cost," ORSA, Vol. 16, No. 3, 1968.

24. Nunn, W., "The Lagrange Multiplier--Heuristic Presentation," OEG Research Report, Number 76, CNA, Washington, D. C., 1966.

25. Ondrasek, M. J., and Wollmer, R. D., "A Model for Targeting Strikes in a LOC Network," RAND Memorandum, RM-5940-PR, September, 1969.

26. Ratliff, H., "Optimal Defense of Multi-Commodity Network," Ph.D. Dissertation, The Johns Hopkins University, 1970.

27. Wollmer, R. D., "Removing Arcs from a Network," ORSA, Vol. 12, No. 6, 1964.

28. Wollmer, R., "Sensitivity Analysis in Networks," Ph.D. Dissertation, University of California, Berkeley, 1965.

29. Wollmer, R. D., "Some Methods for Determining the Most Vital Link in a Railway Network," RAND Memorandum, RM-3321-ISA, April, 1963.

VITA

Guy Thomas Sicilia was born in Frederick, Maryland, on May 16, 1944. He attended Saint Joseph's High School, Emmitsburg, Maryland, from 1958 to 1962, and enrolled at The Johns Hopkins University in September 1962.

At Johns Hopkins he studied Operations Research and in 1966 received a Bachelor of Engineering Science (Operations Research).

In September 1966, Mr. Sicilia began graduate studies in the Industrial and Systems Engineering Department, University of Florida. He was a research assistant from 1966 through September 1969, and was employed by the Mitre Corporation while pursuing his degree from September 1969 to June 1971.

In December 1969, Mr. Sicilia received a Master of Science in Systems Engineering (Operations Research) and completed all requirements for the degree of Doctor of Philosophy in September 1971. He is a member of Alpha Pi Mu, Honorary Industrial Engineering Society.

Mr. Sicilia is married. He and his wife, Anne, have one son, Peter.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

H. D. Ratliff, Chairman
Assistant Professor of Industrial and
Systems Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

M E Thomas

M. E. Thomas
Professor of Industrial and Systems Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

T. J. Hodgson
Assistant Professor of Industrial and
Systems Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Arun Kumar Varma

A. K. Varma
Associate Professor of Mathematics

This dissertation was submitted to the Dean of the College of Engineering and to the Graduate Council, and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December, 1971

Robert E Uhrig

Dean, College of Engineering

Dean, Graduate School