In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

7/25/68

MAXIMAL FUNNEL-NODE FLOWS
IN AN UNDIRECTED NETWORK

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

by

Duane David Miller

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in the School of Industrial and Systems Engineering

Georgia Institute of Technology

June, 1970

MAXIMAL FUNNEL-NODE FLOWS

IN AN UNDIRECTED NETWORK

Approved:

Chairman

Date approved by Chairman: 5/18/70

ACKNOWLEDGMENTS

I gratefully acknowledge the assistance of all the members of my thesis reading committee. I especially wish to recognize the encouragement, assistance, advice and friendship of my faculty advisor, Dr. J. J. Jarvis.

A special debt of gratitude is due my wife and children who have often become strangers to my thoughts during the period of preparation of this thesis.

Finally I wish to acknowledge the excellent typing job done by Mrs. Joene Owen.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

Consider the following military problem. A logistical commander in a recently established theater of operations is receiving ammunition at an ocean terminal for immediate shipment to a forward area. The only trucks available to transport the ammunition are being received at another terminal for ultimate shipment forward. Since there exists a road network of limited capacity, the logistical commander is faced with the problem of determining the routing of trucks which will allow the maximum number of trucks to proceed to the ammunition supply point and then to deliver both the trucks and the ammunition to the forward area. Of course, road capacities may not be violated. The road network can be represented as shown in Figure 1. The lines represent roads with the capacity of each road indicated beside its corresponding line. The circles represent road junctions or terminal points. Traffic may flow in both directions simultaneously along any road; however, the sum of the flows in both directions may not exceed the capacity of the road in question. The problem then is to identify that flow pattern, represented by arrows, which maximizes the flow of trucks from the source of trucks through the source of ammunition to the delivery point.

Now consider a second problem. A communication system designer has been told to establish a message center for an existing communication

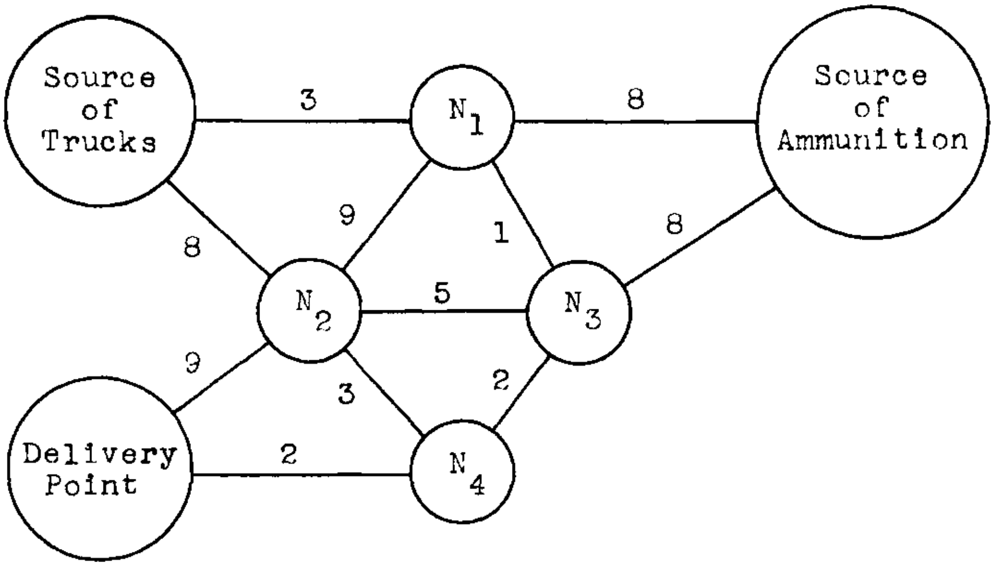

Figure 1. Road Network

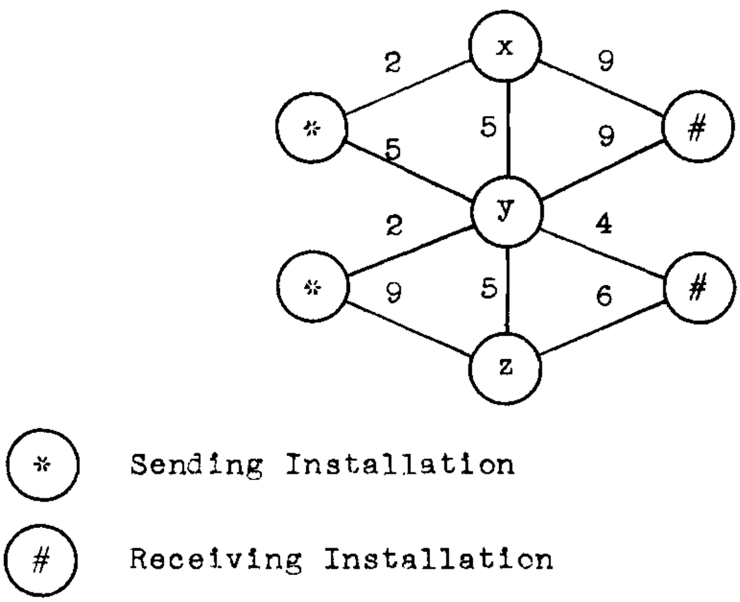

Figure 2. Communication Network

network. All messages are to pass through the message center and the message center must be located at an existing installation. Of course, it is desirable to maintain the maximum possible message flow under the given conditions. The communication network can be represented as shown in Figure 2. The lines represent communication links with the capacity of each link indicated beside its corresponding line. The circles represent communication installations. As in the case of the first problem, two-way communication flow is possible along any link; however, for any link the sum of the flows in both directions may not exceed its capacity. The problem then is to identify that existing installation which, if all flow is required to pass through it, allows the maximum communication flow between the sending installations and the receiving installations.

These two problems are examples of the type of problem with which this paper will be concerned. We will return later to these problems as illustrations of the more general problem to be developed. Let us now begin to define some basic ideas and to survey the appropriate results of others which will lead to a more precise statement of our problem and its solution.

Definition

An undirected network, $G = (N;E)$, consists of a finite set, N, of n elements, N_i, $i = 1 \ldots n$, and a subset, E, of the unordered pairs, (N_i, N_j), of the elements in N.

Interpreting G as a graph, N is a set of nodes (vertices, points) and E is a set of undirected edges (arcs, links) connecting the nodes.

In the two example problems, the circles are the nodes and the lines are the edges. Without loss of generality, we will eliminate self loops from consideration. That is, we will not consider edges which are only incident with a single node.

Definition

Associated with every edge is a non-negative real number, $c(N_i,N_j)$, which will be interpreted as the capacity of edge, (N_i,N_j). Capacity may be thought of as the ability of an edge to transport c units of a certain commodity during a unit of time.

Definition

Let N_s, $N_t \in N$, $s \neq t$, be special nodes called respectively the source and sink. Let us also refer to these nodes unambiguously as simply s and t.

Definition

A cut separating s and t is a subset of E such that its removal will disconnect s from t and no proper subset of it will have the same property.

Definition

A minimum cut separating s and t, denoted (s;t), is a cut such that the sum of the capacities of the edges in the cut is minimal.

Definition

Let c(s;t) be the sum of the capacities of the edges in the minimum cut separating s and t, or simply the capacity of the minimum cut.

Definition

A flow from s to t in an undirected network of value, v(s;t), is a two-dimensional vector mapping, $[f(N_i,N_j), f(N_j,N_i)]$, from E into the non-negative reals that satisfies:

$$\sum_k f(N_j,N_k) - \sum_i f(N_i,N_j) = \begin{cases} v(s;t), & \text{if } j = s \\ 0, & \text{if } j \neq s,t \\ -v(s;t), & \text{if } j = t \end{cases} \qquad (1)$$

$$|f(N_i,N_j) - f(N_j,N_i)| \leq c(N_i,N_j), \text{ all } i, j \qquad (2)$$

$$f(N_i,N_j) \geq 0, \text{ all } i, j, \qquad (3)$$

where $f(N_i,N_j)$ represents flow in edge (N_i,N_j) from N_i to N_j and $f(N_j,N_i)$ represents flow in edge (N_i,N_j) from N_j to N_i.

If

$$f'(N_i,N_j) = f(N_i,N_j)-f(N_j,N_i), \text{ all } i, j, \qquad (4)$$

then (2) and (3) may be rewritten simply as

$$|f'(N_i,N_j)| \leq c(N_i,N_j), \text{ all } i, j.$$

Note that $f'(N_i,N_j)>0$ implies that the direction of flow is from N_i to N_j. Also, since $f'(N_i,N_j) = -f'(N_j,N_i)$, $f'(N_j,N_i)<0$ implies flow from N_i to N_j.

We may now formalize a "single-commodity, max-flow" problem for undirected networks. This problem is

Maximize: $v(s;t)$

Subject to:

$$\sum_k f'(N_j,N_k) - \sum_i f'(N_i,N_j) = \begin{cases} v(s;t), & \text{if } j = s \\ 0, & \text{if } j \neq s,t \\ -v(s;t), & \text{if } j = t \end{cases}$$

$$|f('(N_i,N_j)| \leq c(N_i,N_j), \text{ all } i,j.$$

This is the well-known node-arc formulation of the "single-commodity, max-flow" problem developed by Ford and Fulkerson (1). An equivalent formulation also applies to the case of directed networks.

Definition

Let $v^*(s;t)$ be the maximal value of $v(s;t)$.

Definition

A chain flow from s to t of value, $h(s;t)$, is a flow of value, $h(s;t)$, directed along an uninterupted sequence of nodes and edges beginning at s and terminating at t. Such a sequence is called a chain from s to t.

Ford and Fulkerson have solved the "single-commodity, max-flow" problem by the use of a node labeling algorithm which determines both $v^*(s;t)$ and the appropriate routing of flows through the network. This algorithm is given in Appendix A. They have also shown that:

(i) $v^*(s;t) = c(s;t)$

(ii) If the edge capacities are integral, then $v^*(s;t)$ will be integral.

(iii) If a flow of value, $v(s;t)$, exists, then it may be decomposed into chain flows of value, $h_i(s;t)$, $i = 1 \ldots m$, where m is the number of chains in the decomposition, such that

$$\sum_{i=1}^{m} h_i(s;t) = v(s;t).$$

They have also given an algorithm to accomplish this chain decomposition. A modification of this algorithm is given in Appendix B.

Now an obvious generalization of the "single-commodity, max-flow" problem is to allow more than one commodity to flow between appropriate pairs of sources and sinks. Specifically, we will concern ourselves with the "two-commodity, max-flow" problem.

Definition

Let N_{s_ℓ}, $N_{t_\ell} \varepsilon N$, $S_\ell \neq t_\ell$, $\ell = 1,2$ be respectively the source and sink for commodity ℓ. Again we will refer to these nodes unambiguously as S_ℓ and t_ℓ, $\ell = 1,2$.

Definition

A proper disconnecting set for two pairs of nodes is a subset of E such that its removal will disconnect S_ℓ from t_ℓ, $\ell = 1,2$, and no proper subset of it will have the same property.

Definition

A minimum proper disconnecting set separating S_ℓ from t_ℓ, $\ell = 1,2$, is a proper disconnecting set such that the sum of the capacities of the edges in the proper disconnecting set is minimal over all disconnecting sets. Let this set be denoted by $(s_1,s_2;t_1,t_2)$. Rothschild and Whinston (4) refer to this set as a minimum double-cut.

Definition

Let $c(s_1,s_2;t_1,t_2)$ be the sum of the capacities of the edges in the minimum proper disconnecting set which disconnects s_ℓ from t_ℓ, $\ell = 1,2$. More simply we will refer to this sum as the capacity of the minimum proper disconnecting set or as the capacity of the minimum double-cut.

Definition

Let N_i-N_j indicate a single node formed by combining nodes N_i and N_j. That is, by connecting nodes N_i and N_j with an edge of infinite capacity.

Definition

A two-commodity flow in an undirected network of value, $v(s_1;t_1) + v(s_2;t_2)$, is a four-dimensional vector mapping, $[f_1(N_i,N_j),\ f_1(N_j,N_i),\ f_2(N_i,N_j),\ f_2(N_j,N_i)]$, from E into the non-negative reals that satisfies:

$$\sum_k f_1(N_j,N_k) - \sum_i f_1(N_i,N_j) = \begin{cases} v(s_1;t_1), & \text{if } j = s_1 \\ 0 \quad , & \text{if } j \neq s_1,t_1 \\ -v(s_1;t_1), & \text{if } j = t_1 \end{cases}$$

$$\sum_k f_2(N_j,N_k) - \sum_i f_2(N_i,N_j) = \begin{cases} v(s_2;t_2), & \text{if } j = s_2 \\ 0 \quad , & \text{if } j \neq s_2,t_2 \\ -v(s_2;t_2), & \text{if } j = t_2 \end{cases}$$

$$|f_1(N_i,N_j)-f_1(N_j,N_i)|+|f_2(N_i,N_j)-f_2(N_j,N_i)|\leq c(N_i,N_j), \text{ all } i,j \quad (5)$$

$$f_1(N_i,N_j),\ f_2(N_i,N_j) \geq 0, \text{ all } i,j. \quad (6)$$

If we define $f_1'(N_i,N_j)$ and $f_2'(N_i,N_j)$ analogously to $f'(N_i,N_j)$ for single-commodity flows, then (5) and (6) may be rewritten simply as

$$|f_1'(N_i,N_j)|+|f_2'(N_i,N_j)|\leq c(N_i,N_j), \text{ all } i,j$$

Note the critical fact that opposing flows of different commodities may not be cancelled.

Now the "two-commodity, max-flow" problem may be formalized as follows:

$$\text{Maximize: } v(s_1;t_1) + v(s_2;t_2)$$

Subject to:

$$\sum_k f_1'(N_j,N_k) - \sum_i f_1'(N_i,N_j) = \begin{cases} v(s_1;t_1), & \text{if } j = s_1 \\ 0, & \text{if } j \neq s_1,t_1 \\ -v(s_1;t_1), & \text{if } j = t_1 \end{cases}$$

$$\sum_k f_2'(N_j,N_k) - \sum_i f_2'(N_i,N_j) = \begin{cases} v(s_2;t_2), & \text{if } j = s_2 \\ 0, & \text{if } j \neq s_2,t_2 \\ -v(s_2;t_2), & \text{if } j = t_2 \end{cases}$$

$$|f_1'(N_i,N_j)|+|f_2'(N_i,N_j)|\le c(N_i,N_j), \text{ for all } i,j$$

Definition

Let $MAX[v(s_1;t_1) + v(s_2;t_2)]$ be the maximal value of $v(s_1;t_1) + v(s_2;t_2)$ in the previous definition.

Hu (2) has solved the "two-commodity, max-flow" problem using an algorithm involving flow exchanges which determines both $MAX[v(s_1;t_1) + v(s_2;t_2)]$ and the appropriate routing of flows through the network. He has also shown that:

(i) A solution to the "two-commodity, max-flow" problem always exists and there are, at least, two solutions (possibly identical) of which one has the property that

$$v(s_1;t_1) = v^*(s_1;t_1)$$

and the other has the property that

$$v(s_2;t_2) = v^*(s_2;t_2).$$

(ii) $MAX[v(s_1;t_1)+v(s_2;t_2)]=MIN[c(s_1-s_2;t_1-t_2),c(s_1-t_2;t_1-s_2)]$.

This result has also been shown by Rothfarb, Shein and Frisch (3).

(iii) A solution to the "two-commodity, max-flow" problem is feasible only if

$$v(s_1;t_1) \leq c(s_1;t_1)$$

$$v(s_2;t_2) \leq c(s_2;t_2)$$

$$v(s_1;t_1) + v(s_2;t_2) \leq c(s_1,s_2;t_1,t_2)$$

(iv) If the capacities of all edges are even integers, then $\mathrm{MAX}[v(s_1;t_1) + v(s_2;t_2)]$ will be an integer.

<u>Definition</u>

Let

$$v^+(s_1;t_1) = \mathrm{MAX}[v(s_1;t_1)+v(s_2;t_2)] - v^*(s_2;t_2)$$

and

$$v^+(s_2;t_2) = \mathrm{MAX}[v(s_1;t_1)+v(s_2;t_2)] - v^*(s_1;t_1).$$

It should be noted that Hu's first result guarantees the existence of $v^+(s_1,t_1)$ and $v^+(s_2,t_2)$. Now using this definition, it follows that Hu's first result may be rewritten: At least two solutions (possibly identical) to the "two-commodity, max-flow" problem exist such that

$$\begin{aligned} \mathrm{MAX}[v(s_1;t_1)+v(s_2;t_2)] &= v^*(s_1;t_1) + v^+(s_2;t_2) \\ &= v^+(s_1;t_1) + v^*(s_2;t_2). \end{aligned}$$

Rothschild and Whinston (4) have also solved the "two-commodity, max-flow" problem and have shown that $\mathrm{MAX}[v(s_1;t_1) + v(s_2;t_2)]$ will be an integer if the capacities of the edges are integer (not necessarily even) and at each node the sum of the capacities of all incident edges is an even integer. This is called an Euler network. Arinal (5) has recently presented another algorithm for finding maximal, two-commodity flows. Rothschild and Whinston (6) have developed a method for solving a specialization of the "two-commodity, max-flow" problem, where $s_1 = t_2$ and $s_2 = t_1$ in an Euler network.

Now let us turn to the task of precisely defining our problem.

Definition

A funnel-node flow is a single-commodity flow which passes through a specified node other than s or t. Such a node will be called a funnel-node and will be represented by a. More specifically a funnel-node flow in an undirected network of value, v(s;a;t), is a four-dimensional vector mapping, $[f_1(N_i,N_j), f_1(N_j,N_i), f_2(N_i,N_j), f_2(N_j,N_i)]$, from E into the non-negative reals which satisfies:

$$\sum_k f_1(N_j,N_k) - \sum_i f_1(N_i,N_j) = \begin{cases} v(s;a), & \text{if } j = s \\ 0\quad, & \text{if } j \neq s,a \\ -v(s;a), & \text{if } j = a \end{cases}$$

$$\sum_k f_2(N_j,N_k) - \sum_i f_2(N_i,N_j) = \begin{cases} v(a;t), & \text{if } j = a \\ 0\quad, & \text{if } j \neq a,t \\ -v(a;t), & \text{if } j = t \end{cases}$$

$$|f_1(N_i,N_j)-f_1(N_j,N_i)|+|f_2(N_i,N_j)-f_2(N_j,N_i)| \leq c(N_i,N_j), \text{ all } i,j$$

$$f_1(N_i,N_j), f_2(N_i,N_j) \geq 0, \text{ all } i,j$$

$$v(s;a;t) = v(s;a) = v(a;t).$$

Now defining $f_1'(N_i,N_j)$ and $f_2'(N_i,N_j)$ as before we obtain a precise statement of the "funnel-node, max flow" problem.

Maximize: $v(s;a;t)$

Subject to:

$$\sum_k f_1'(N_j,N_k) - \sum_i f_1'(N_i,N_j) = \begin{cases} v(s;a), & \text{if } j = s \\ 0, & \text{if } j \neq s,a \\ -v(s;a), & \text{if } j = a \end{cases}$$

$$\sum_k f_2'(N_j,N_k) - \sum_i f_1'(N_i,N_j) = \begin{cases} v(a;t), & \text{if } j = a \\ 0, & \text{if } j \neq a,t \\ -v(a;t), & \text{if } j = t \end{cases}$$

$$|f_1'(N_i,N_j)| + |f_2'(N_i,N_j)| \leq c(N_i,N_j), \text{ all } i, j$$

$$v(s;a;t) = v(s;a) = v(a;t).$$

<u>Definition</u>

Let $v^*(s;a;t)$ be the maximal value of $v(s;a;t)$.

The similarity of the "funnel-node-max flow" problem and the "two-commodity, max-flow" problem is evident. The constraint sets are identical with the exception that in the funnel-node problem

(i) $t_1 = s_2$

(ii) $v(s;a) = v(a;t)$

Now with the addition of constraint (ii), the function to be maximized is exactly one-half of that which is to be maximized in the "two-commodity, max-flow" problem. We will investigate the effect of these differences in the next chapter.

CHAPTER II

MAXIMAL FUNNEL-NODE FLOWS

In this chapter we will establish the principal result of this thesis by the statement and proof of a theorem. We will then use the theorem to develop and demonstrate an algorithm for the construction of maximal funnel-node flows in undirected networks.

1. "Funnel-Node, Max-Flow" Theorem

If $G = (N,E)$ is an undirected network, then

$$v^*(s;a;t) = \mathrm{MIN}\{v^*(s;a),\ v^*(a;t),\ \tfrac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]\}.$$

Proof. The strategy of proof will be:

(i) Establish two upper bounds on $v^*(s;a;t)$. (Lemmas 1 and 2).

(ii) Using lemmas 1 and 2 and the results of Hu, determine $v^*(s;a;t)$ for all possible values of $v^*(s;a)$, $v^*(a;t)$, $v^+(s;a)$ and $v^+(a;t)$. (Lemmas 3a, 3b, and 4).

(iii) Show that

$$\mathrm{MIN}\{v^*(s;a),\ v^*(a;t),\ \tfrac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]\}$$

yields the appropriate value of $v^*(s;a;t)$. (Lemmas 5a, 5b, and 6).

Lemma 1

$$v^*(s;a;t) \leq \mathrm{MIN}[v^*(s;a),\ v^*(a;t)]$$

Proof. Suppose the lemma is not true. Then,

$$v^*(s;a;t) > \mathrm{MIN}[v^*(s;a),\ v^*(a;t)].$$

Consider two cases:

Case 1. If $v^*(s;a) \geq v^*(a;t)$, then $v^*(s;a;t) > v^*(a;t)$. But since $v^*(a;t) \geq v(a;t)$, this clearly violates the definition of $v(s;a;t)$ which requires that for any funnel-node flow to be feasible, $v(s;a;t) = v(s;a) = v(a;t)$.

Case 2. If $v^*(s;a) < v^*(a;t)$, then $v^*(s;a;t) > v^*(s;a)$. As in Case 1 this again violates the definition of $v(s;a;t)$.

Q.E.D.

<u>Lemma 2</u>

$$v^*(s;a;t) \leq \tfrac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)].$$

Proof:

$$\begin{aligned}
\tfrac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)] &\geq \tfrac{1}{2} \underset{v(s;a)=v(a;t)}{\mathrm{MAX}} [v(s;a)+v(a;t)] \\
&\geq \tfrac{1}{2}\mathrm{MAX}[2v(s;a;t)] \\
&\geq \mathrm{MAX}[v(s;a;t)] \\
&\geq v^*(s;a;t)
\end{aligned}$$

Q.E.D.

We now determine $v^*(s;a;t)$ for all possible values of $v^*(s;a)$, $v^*(a;t)$, $v^+(s;a)$ and $v^+(a;t)$.

Lemma 3a

$v^*(s;a;t) = v^*(s;a)$ if and only if $v^*(s;a) \leq v^+(a;t)$.

Proof. We first prove the sufficiency of the assumption. Let $v^*(s;a) \leq v^+(a;t)$. By Lemma 1, $v^*(s;a;t) \leq v^*(s;a)$. Now consider two cases.

Case 1. $v^*(s;a) = v^+(a;t)$. In this case the flow corresponding to $v^*(s;a)$ and $v^+(a;t)$ is a feasible funnel-node flow with value $v^*(s;a)$. Thus it is optimal.

Case 2. $v^*(s;a) < v^+(a;t)$. Now, from the flow solution yielding $v^*(s;a)$ and $v^+(a;t)$, successively reduce flow along chains from a to t up to an amount $\delta = [v^+(a;t) - v^*(s;a)]$. This new flow is a feasible funnel-node flow and has value $v^*(s;a)$. Therefore, it must be optimal.

We now prove the necessity of the assumption. Let $v^*(s;a;t) = v^*(s;a)$. Recalling the previously discussed work of Hu, if we let $s_1 = s$, $t_2 = t$ and $s_2 = t_1 = a$, then the "two-commodity, max-flow" problem becomes:

$$\text{Maximize:} \quad v(s;a) + v(a;t) \tag{8}$$

$$\begin{aligned} \text{Subject to:} \quad & v(s;a) \leq c(s;a) \\ & v(a;t) \leq c(a;t) \\ & v(s;a) + v(a;t) \leq c(s,a:a,t). \end{aligned}$$

At least two optimal solutions (possibly identical) exist. They are

$$\begin{bmatrix} v^*(s;a) \\ v^+(a;t) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} v^+(s;a) \\ v^*(a;t) \end{bmatrix}.$$

Thus $MAX[v(s;a)+v(a;t)] = v^*(s;a)+v^+(a;t) = v^+(s;a)+v^*(a;t)$. Now by Lemma 2, $v^*(s;a;t) \leq \frac{1}{2}MAX[v(s;a)+v(a;t)]$ or $2v^*(s;a;t) \leq [v^*(s;a)+v^+(a;t)]$. But $v^*(s;a;t) = v^*(s;a)$. Therefore, $2v^*(s;a) \leq [v^*(s;a)+v^+(a;t)]$ or $v^*(s;a) \leq v^+(a;t)$.

Q.E.D.

Lemma 3b

$v^*(s;a;t) = v^*(a;t)$ if and only if $v^*(a;t) \leq v^+(s;a)$.

Proof. As in Lemma 3a.

It should be noted that there is no problem concerning the existence of $v^*(s;a;t)$ under the conditions imposed by Lemmas 3a and 3b. We have assumed the existence of the appropriate single-commodity flows and in Case 2 of each lemma we have reduced one of the flows which does not violate feasibility.

Lemma 4

$v^*(s;a;t) = \frac{1}{2}MAX[v(s;a)+v(a;t)]$ if and only if $v^*(s;a) \geq v^+(a;t)$ and $v^*(a;t) \geq v^+(s;a)$.

Proof. We first prove the sufficiency of the assumptions. Let $v^*(s;a) \geq v^+(a;t)$ and $v^*(a;t) \geq v^+(s;a)$. By Lemma 2 we know that $v^*(s;a;t) \leq \frac{1}{2}MAX[v(s;a)+v(a;t)]$. Then, if we can show the existence of a funnel-node flow such that $v(s;a;t) = \frac{1}{2}MAX[v(s;a)+v(a;t)]$, we will have proven the desired result.

Case 1. $v^*(s;a) = v^+(a;t)$. By Lemma 3a, $v^*(s;a;t) = v^*(s;a) = \frac{1}{2}[v^*(s;a)+v^+(a;t)] = \frac{1}{2}MAX[v(s;a)+v(a;t)]$.

Case 2. $v^*(a;t)=v^+(s;a)$. The argument proceeds as in Case 1.

Case 3. $v^*(s;a)>v^+(a;t)$ and $v^*(a;t)>v^+(s;a)$.

We begin by stating a well-known property of linear programs: If $\bar{v}_1$ and $\bar{v}_2$ are two different optimal solutions to a linear programming problem, then

$$\alpha\bar{v}_1 + (1-\alpha)\bar{v}_2,\ 0\leq\alpha\leq 1,$$

is also an optimal solution.

Now using this property and again recalling the work of Hu,

$$\alpha \begin{bmatrix} v^*(s;a) \\ v^+(a;t) \end{bmatrix} + (1-\alpha) \begin{bmatrix} v^+(s;a) \\ v^*(a;t) \end{bmatrix}$$

is also an optimal solution to (8) if $0\leq\alpha\leq 1$. But we wish to impose the additional constraint that $v(s;a) = v(a;t)$. Therefore, if we can demonstrate for

$$\alpha \begin{bmatrix} v^*(s;a) \\ v^+(a;t) \end{bmatrix} + (1-\alpha) \begin{bmatrix} v^+(s;a) \\ v^*(a;t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\{v^*(a;t)+v^+(s;a)\} \\ \frac{1}{2}\{v^*(a;t)+v^+(s;a)\} \end{bmatrix}, \tag{9}$$

$$v^*(s;a)\geq v^+(a;t),$$

$$v^*(a;t)\geq v^+(s;a),$$

that α is contained in the closed interval [0,1], then we will have shown the existence of a solution such that

$$v(s;a;t) = \tfrac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]$$

and the sufficiency of the assumptions will have been proven and the existence of the flow shown. Solving either equation of (9) for α, since both yield the same value, we obtain

$$\alpha = \frac{v^*(a;t) - v^+(s;a)}{2[v^*(s;a) - v^+(s;a)]} .$$

Now suppose α is not contained in the closed interval [0,1], then either $\alpha<0$ or $\alpha>1$.

Case 1. If

$$\frac{v^*(a;t) - v^+(s;a)}{2[v^*(s;a) - v^+(s;a)]} < 0,$$

then $v^*(a;t)<v^+(s;a)$. But this is a contradiction.

Case 2. If

$$\frac{v^*(a;t) - v^+(s;a)}{2[v^*(s;a) - v^+(s;a)]} > 1,$$

then $v^*(a;t)+v^+(s;a)>2v^*(s;a)$. But since

$v^*(a;t)+v^+(s;a) = v^*(s;a)+v^+(a;t)$, it follows that $v^*(s;a)+v^+(a;t)>2v^*(s;a)$ or $v^*(s;a)<v^+(a;t)$. But this also is a contradiction. We therefore conclude that $0\leq\alpha\leq 1$ and that $v^*(s;a;t) = \frac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]$.

We now prove the necessity of the assumptions. Let $v^*(s;a;t) = \frac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]$ or

$$2v^*(s;a;t) = v^*(s;a)+v^+(a;t) = v^+(s;a)+v^*(a;t). \tag{10}$$

Assume the result is not true, that is $v^*(s;a)<v^+(a;t)$ or $v^*(a;t)<v^+(s;a)$ or both.

Case 1. $v^*(s;a)<v^+(a;t)$. Substituting in (10) we obtain $2v^*(s;a;t)>2v^*(s;a)$ or $v^*(s;a;t)>v^*(s;a)$. But this contradicts Lemma 1.

Case 2. $v^*(s;a)<v^+(s;a)$. The argument proceeds as in Case 1.

Case 3. $v^*(s;a)<v^+(a;t)$ and $v^*(a;t)<v^+(s;a)$. This case is a subset of cases 1 and 2.

Q.E.D.

Let us summarize our results to this point in Table 1. It can be seen that we have established the value of $v^*(s;a;t)$ for all possible values of $v^*(s;a)$, $v^*(a;t)$, $v^+(s;a)$ and $v^+(a;t)$. It should be noted that if $v^*(s;a) = v^+(a;t)$, we have determined two expressions for $v^*(s;a;t)$. From Lemma 3a we obtained $v^*(s;a;t) = v^*(s;a)$. Since $v^*(s;a) = v^+(a;t)$ implies that $v^*(a;t)\geq v^+(s;a)$, we obtain from Lemma 4, $v^*(s;a;t) = \frac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]$ or $v^*(s;a;t) = \frac{1}{2}[v^*(s;a)+v^+(a;t)]$.

Table 1. Summary of Results

Relationship of $v^*(s;a)$ to $v^+(a;t)$ and $v^*(a;t)$ to $v^+(s;a)$	$v^*(s;a;t)$
$v^*(s;a) \leq v^+(a;t)$	$v^*(s;a)$
$v^*(a;t) \leq v^+(s;a)$	$v^*(a;t)$
$v^*(s;a) \geq v^+(a;t)$ and $v^*(a;t) \geq v^+(s;a)$	$\frac{1}{2}MAX[v(s;a)+v(a;t)]$

But since $v^*(s;a) = v^+(a;t)$, $v^*(s;a;t) = v^*(s;a)$ and we have shown that the two expressions for $v^*(s;a;t)$ are identical. A similar argument can be made in the case of $v^*(a;t) = v^+(s;a)$.

What remains to be done is to show that the proper value of $v^*(s;a;t)$ is given by the expression

$$\mathrm{MIN}\{v^*(s;a),\ v^*(a;t),\ \tfrac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]\}$$

This will be done in Lemmas 5a, 5b and 6.

Lemma 5a

$v^*(s;a;t) = v^*(s;a)$ if and only if

$$\mathrm{MIN}\{v^*(s;a),\ v^*(a;t),\ \tfrac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]\} = v^*(s;a).$$

Proof. First we prove the sufficiency of the assumption. Let $v^*(s;a) \leq v^*(a;t)$ and $v^*(s;a) \leq \frac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]$. Now $v^*(s;a) \leq \frac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]$ implies that $v^*(s;a) \leq v^+(a;t)$. By Lemma 3a, $v^*(s;a;t) = v^*(s;a)$.

We now prove the necessity of the assumption. Let $v^*(s;a;t) = v^*(s;a)$. By Lemma 3a, $v^*(s;a) \leq v^+(a;t)$. Since $v^*(a;t) \geq v^+(a;t)$, $v^*(s;a) \leq v^*(a;t)$. Now by Lemma 2, $v^*(s;a;t) \leq \frac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]$ or $v^*(s;a) \leq \frac{1}{2}\mathrm{MAX}[v(s;a)+v(a;t)]$.

Q.E.D.

Lemma 5b

$v^*(s;a;t) = v^*(a;t)$ if and only if

$$\text{MIN}\{v^*(s;a),\ v^*(a;t),\ \tfrac{1}{2}\text{MAX}[v(s;a)+v(a;t)]\} = v^*(a;t).$$

Proof. As in Lemma 5a.

Lemma 6

$v^*(s;a;t) = \frac{1}{2}\text{MAX}[v(s;a)+v(a;t)]$ if and only if

$$\text{MIN}\{v^*(s;a),\ v^*(a;t),\ \tfrac{1}{2}\text{MAX}[v(s;a)+v(a;t)]\} = \tfrac{1}{2}\text{MAX}[v(s;a)+v(a;t)].$$

Proof. First we prove the sufficiency of the assumption. Let $\frac{1}{2}\text{MAX}[v(s;a)+v(a;t)]\leq v^*(s;a)$ and $\frac{1}{2}\text{MAX}[v(s;a)+v(a;t)]\leq v^*(a;t)$. Therefore, $v^*(s;a)+v^+(a;t)\leq 2v^*(s;a)$ or $v^+(a;t)\leq v^*(s;a)$. Similarly, $v^+(s;a)\leq v^*(a;t)$. By Lemma 4, $v^*(s;a;t) = \frac{1}{2}\text{MAX}[v(s;a)+v(a;t)]$.

We now prove the necessity of the assumption. Let $v^*(s;a;t) = \frac{1}{2}\text{MAX}[v(s;a)+v(a;t)]$. Then by Lemma 4, $v^*(s;a)\geq v^+(a;t)$ and $v^*(a;t)\geq v^+(s;a)$. Therefore,

$$\tfrac{1}{2}\text{MAX}[v(s;a)+v(a;t)] = \tfrac{1}{2}[v^*(s;a)+v^+(a;t)]\leq\tfrac{1}{2}[v^*(s;a)+v^*(s;a)] = v^*(s;a).$$

$$= \tfrac{1}{2}[v^+(s;a)+v^*(a;t)]\leq\tfrac{1}{2}[v^*(a;t)+v^*(a;t)] = v^*(a;t).$$

Q.E.D.

Thus the theorem is true.

2. Algorithm

The "funnel-node, max-flow" theorem leads directly to the following algorithm for the construction of maximal funnel-node flows in an undirected network:

Step 1. Solve for $v^*(s;a)$ and $v^*(a;t)$ using the algorithm given in Appendix A.

Step 2. Construct a new network G' by the addition to G of a node s' and edges (s',s) and (s',t) each with infinite capacity. Now solve for $v^*(s';a)$ using the algorithm given in Appendix A.

Step 3. Determine

$$v^*(s;a;t) = \mathrm{MIN}[v^*(s;a),\ v^*(a;t),\ \tfrac{1}{2}v^*(s';a)].$$

If $v^*(s;a;t) = 0$, stop. No flow is possible.

Step 4. Construct a new network G" by the addition to G of a node s" and edges (s",s) and (s",t) each with capacity equal to $v^*(s;a;t)$. Solve for $v^*(s'';a)$ using the algorithm given in Appendix A. Decompose the flow pattern obtained into a flow from s" through s to a and a flow from a through t to s" using the algorithm given in Appendix B. Remove node s" and edges (s",s) and (s",t). The result is a "funnel-node, max-flow" from s through a to t. See Figure 3 for flow chart.

The use of $v^*(s';a)$ to determine the value of $\mathrm{MAX}[v(s;a)+v(a;t)]$ in Step 2 of the algorithm is a consequence of the second result of Hu. That result is

$$\mathrm{MAX}[v(s_1;t_1)+v(s_2;t_2)] = \mathrm{MIN}[c(s_1-s_2;t_1-t_2),\ c(s_1-t_2;t_1-s_2)].$$

But for our problem $s_1=s$, $s_2=t_1=a$ and $t_2=t$. Making these substitutions, Hu's result is

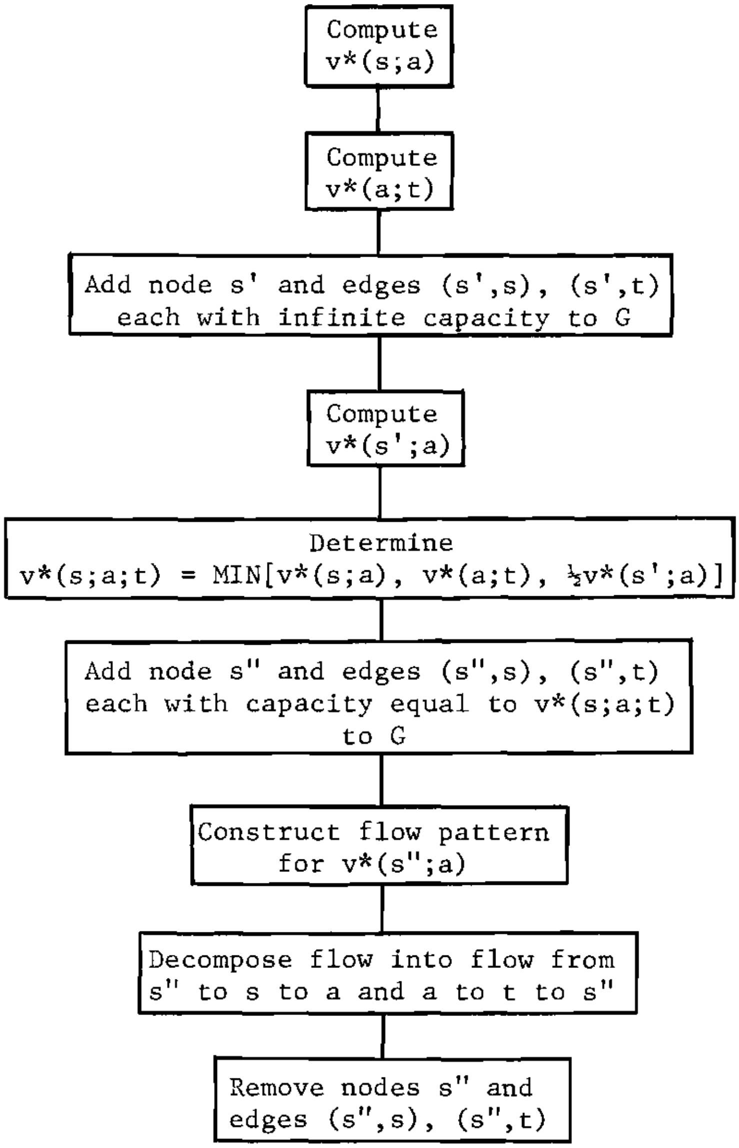

Figure 3. Flow Chart

$$MAX[v(s;a)+v(a;t)] = MIN[c(s-a;a-t), c(s-t: a].$$

But (s-a; a-t) implies that a is in both components of G which violates the definition of a cut. Therefore this case cannot exist and

$$MAX[v(s;a)+v(a;t)] = c(s-t; a).$$

Thus we can determine MAX[v(s;a)+v(a;t)] by computing v*(s-t; a) or, equivalently, by computing v*(s';a).

Since the algorithm involves only the use of finite processes, it is itself finite.

3. Example Problems

We now return to the first problem presented at the beginning of Chapter I. Let s represent the terminal at which trucks are being received, a represents the terminal at which ammunition is being received and t the point of delivery for both trucks and ammunition. Figure 4 is a graphical representation of the road network. Note that the road capacities are indicated by the number beside each edge. The reader will recall that we wish to determine the routing of trucks through the network such that the flow of trucks from s through a to t is maximal. Using the "single-commodity, max-flow" algorithm, we determine v*(s;a) = 11, v*(a;t) = 11 and v*(s';a) = 8. Appending node s" and edges (s";s) and (s";t) both with a capacity of 8 and

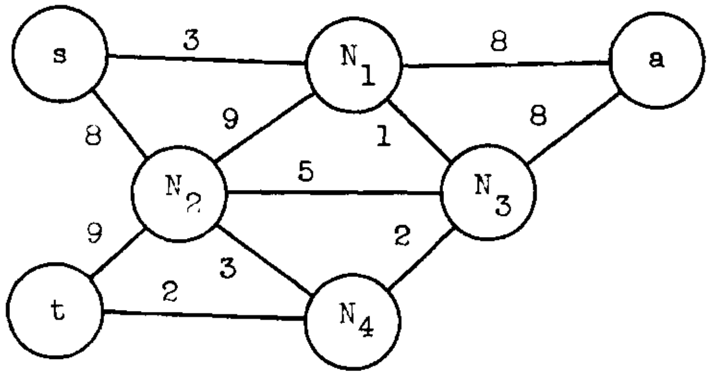

Figure 4. Road Network

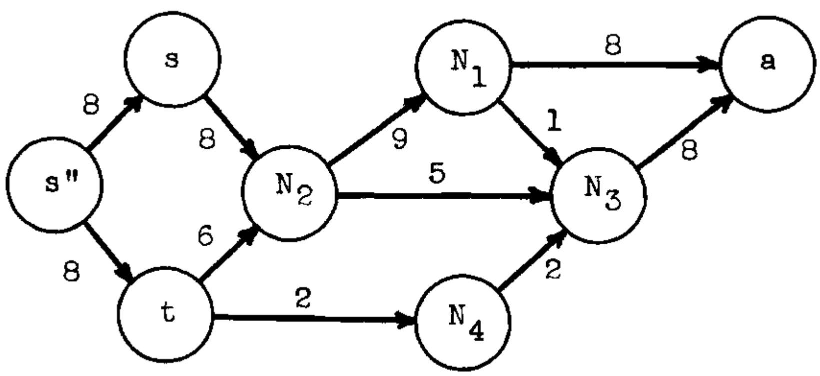

Figure 5. Flow Pattern for v*(s";a)

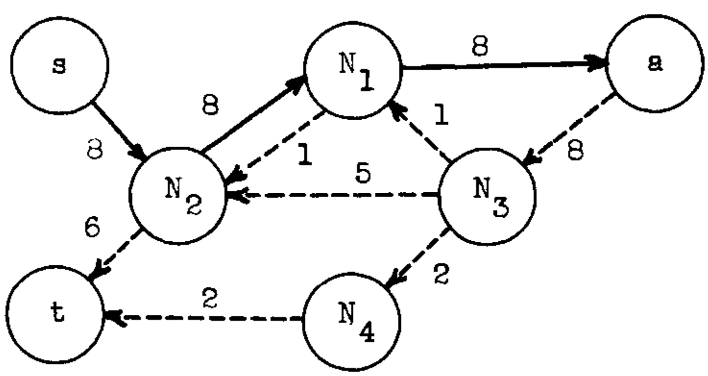

Figure 6. Flow Pattern for v*(s;a;t)

solving for v*(s";a), we obtain the flow pattern shown in Figure 5. Note that the number beside each edge indicates the flow through the edge. The direction of flow is indicated by a bold-faced arrow. Now, decomposing the flow into flow from s" through s to a and flow from a through t to s", we obtain the maximal funnel-node flow pattern shown in Figure 6. Note that flow directed from s to a is indicated by a bold-faced arrow and flow directed from a to t is indicated by a broken arrow.

A simple extension to the first example problem may be formed by requiring that the logistical commander not only send forward the maximum amount of ammunition but that, given that requirement, he also send forward the maximum number of trucks. That is, he desires to determine the flow pattern corresponding to the problem:

$$\text{Maximize:} \quad [v(s;t)+v^*(s;a;t)]$$

$$\text{Subject to:}$$

$$\sum_k f_1'(N_j,N_k) - \sum_i f_1'(N_i,N_j) = \begin{cases} v^*(s;a), & \text{if } j = s \\ 0, & \text{if } j \neq s,a \\ -v^*(s;a), & \text{if } j = a \end{cases}$$

$$\sum_k f_2'(N_j,N_k) - \sum_i f_2'(N_i,N_j) = \begin{cases} v^*(a;t), & \text{if } j = a \\ 0, & \text{if } j \neq a,t \\ -v^*(a;t), & \text{if } j = t \end{cases}$$

$$\sum_k f_3'(N_j,N_k) - \sum_i f_3'(N_i,N_j) = \begin{cases} v(s;t), & \text{if } j = s \\ 0, & \text{if } j \neq s,t \\ -v(s;t), & \text{if } j = t \end{cases}$$

$$|f_1'(N_i,N_j)|+|f_2'(N_i,N_j)|+|f_3'(N_i,N_j)| \leq c(N_i,N_j), \text{ all } i,\ j$$

$$v^*(s;a) = v^*(a;t) = v^*(s;a;t)$$

where $f_3'(N_i,N_j)$ is the flow through edge $(N_i,N_j)\varepsilon G$ from s to t other than that through a.

But we have shown that $v^*(s;a;t)$ can be represented by $v^*(s'';a)$ and therefore the problem may be rewritten as:

Maximize: $[v(s;t)+v^*(s'';a)]$

Subject to:

$$\sum_k f_{12}'(N_j,N_k) - \sum_i f_{12}'(N_i,N_j) = \begin{cases} v^*(s'';a), & \text{if } j = s'' \\ 0, & \text{if } j \neq s'',a \\ -v^*(s';a), & \text{if } j = a \end{cases}$$

$$\sum_{k\neq s''} f_3'(N_j,N_k) - \sum_{i\neq s''} f_3'(N_i,N_j) = \begin{cases} v(s;t), & \text{if } j = s \\ 0, & \text{if } j \neq s,t \\ -v(s;t), & \text{if } j = t \end{cases} \tag{11}$$

$$|f_{12}'(N_i,N_j)|+|f_3'(N_i,N_j)|\leq c(N_i,N_j),\ \text{all } i,\ j$$

where $f_{12}'(N_i,N_j)$ is the flow through edge $(N_i,N_j)\varepsilon G''$ from s" to a.

Except for the requirement in (11) that the summations not include $f_3'(s,s'')$, $f_3'(s'',s)$, $f_3'(t,s'')$ and $f_3'(s'',t)$, this is a "two-commodity, max-flow" problem on G" in which one of the two commodities has been maximized by the "single-commodity, max-flow" algorithm. With the exception noted, this is exactly the initial condition required by Hu's "two-commodity, max-flow" algorithm. (The reader is referred to the paper by Hu for the details of this algorithm.)

We now show that the use of Hu's algorithm will, in effect, eliminate the exception and, therefore, it may be used to solve the problem. Since the algorithm guarantees that the flow of the commodity that is initially maximal will remain maximal, the edges appended to form G" will always be capacitated with flow, v(s",a) and the application of the algorithm will always result in $f_3'(s,s'') = 0$, $f_3'(s'',s) = 0$, $f_3'(t,s'') = 0$ and $f_3'(s'',t) = 0$. Thus the use of Hu's algorithm will insure that the conditions required by (11) are met and, therefore, the algorithm will yield a valid result to MAX[v(s;t)+v*(s;a;t)] after the usual decomposition of the flow pattern corresponding to v*(s";a) is performed. Figure 7 shows the result of the application of this procedure to the original road network. Note that v*(s;a;t) is still equal to 8; however, we can now send three trucks from s to t without passing through a. Thus,

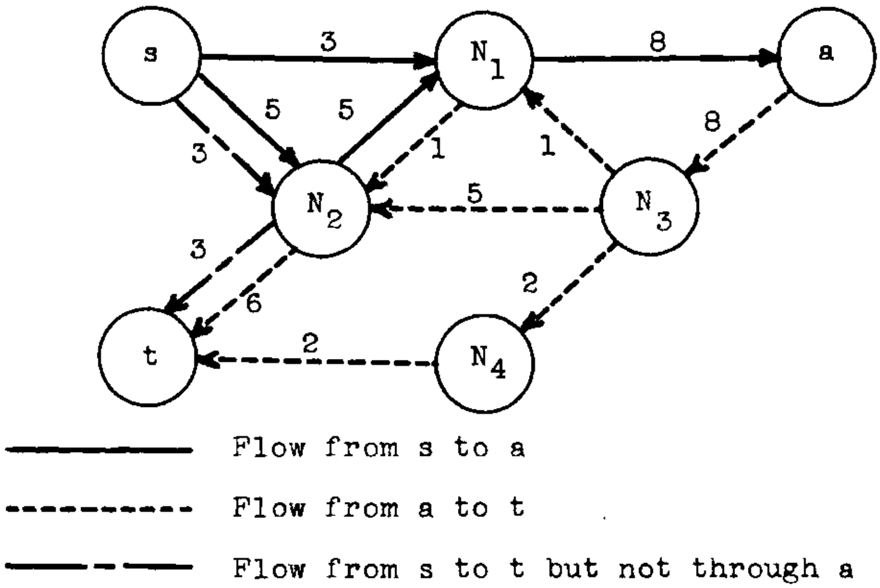

Figure 7. Flow Pattern for MAX[v(s;t)+v*(s;a;t)]

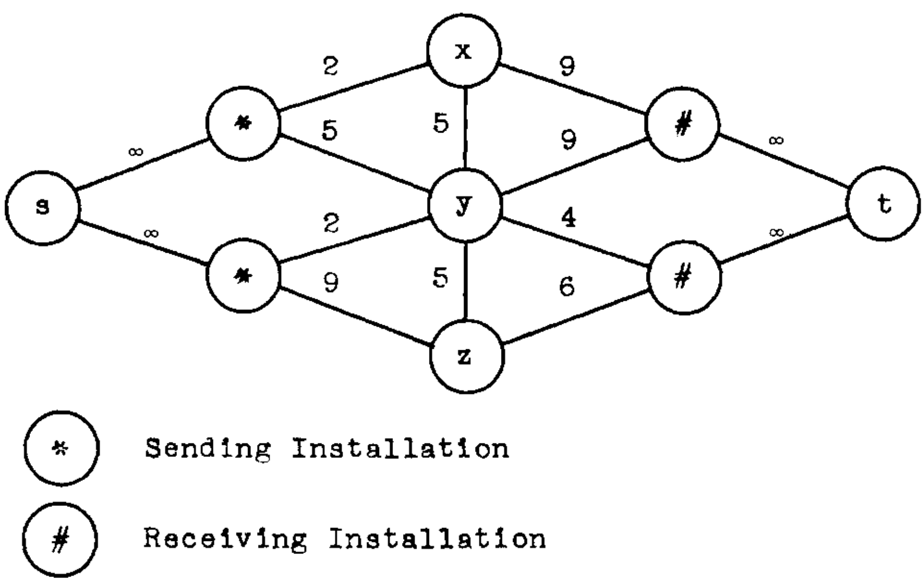

Figure 8. Augmented communication Network

$$MAX[v(s;t)+v^*(s;a;t)] = 11.$$

We now examine the second example problem. If we append nodes s and t and edges with infinite capacity between s and all sending installations and between t and all receiving installations, we obtain the graphical representation of the communication network shown in Figure 8. The reader will recall that we desire to determine that node, N_j, such that $v^*(s;N_j;t)$ is maximal for $N_j = x, y, z$. The results of the appropriate computations are contained in Table 2. It can be seen that the communication system designer would locate the message center at the installation represented by node y.

Table 2. Computational Results for Second Example Problem

Node x	Node y	Node z
$v^*(s;x) = 16$	$v^*(s;y) = 18$	$v^*(s;z) = 18$
$v^*(x;t) = 16$	$v^*(y;t) = 26$	$v^*(z;t) = 13$
$\frac{1}{2}v^*(s';x) = 8$	$\frac{1}{2}v^*(s';y) = 15$	$\frac{1}{2}v^*(s';z) = 10$

CHAPTER III

CONCLUSIONS AND RECOMMENDATIONS

The principal results of this thesis are:

(i) The concept of a funnel-node flow in an undirected network has been defined.

(ii) An algorithm has been presented which computes the maximal value of a funnel-node flow by three applications of a "single-commodity, max-flow" algorithm and determines the maximal flow pattern by an additional application of a "single-commodity, max-flow" algorithm and the application of a chain flow decomposition algorithm.

Further research is recommended in three general areas of study. These are:

(i) An investigation of the applicability of well-known, single-commodity flow results to funnel-node flows. Of particular interest would be those results pertaining to minimal cost flows. Since we have shown that funnel-node flows can be represented by a single-commodity flow, it seems likely that this area of research might prove fruitful.

(ii) An attempt to produce results similar to those of this thesis for directed networks and for more than one funnel-node. It is the conjecture of the author that the "Funnel-Node, Max-Flow" Theorem is also valid for directed networks. It seems likely that an argument similar to the one developed in this thesis could be made for directed networks, if a result similar to the first result of Hu could be established for these networks. Of course, even the truth of this conjecture would not lead directly to an algorithm as there would still be the difficulty of determining $\frac{1}{2}\text{MAX}[v(s;a)+v(a;t)]$. There is presently no known method of determining this value. An attempt to develop results for more than one funnel-node seems to be directly tied to the ability to solve multicommodity flow problems for more than two commodities. Except for very special cases, these problems have not been solved. It might be profitable, however, to try to identify the cut sets of these problems and to relate, not only, the capacity of these sets to a maximal flow value but to investigate the relationship between the number of times that a flow traverses the edge of a cut set and maximal flow value.

(iii) Some interesting results in the analysis of networks might be generated by defining a flow center as a node, N_i, such that

$$\underset{N_i}{\text{MAX}}\{\text{MAX}[v(N_i,s)+v(N_i,t)]\}.$$

These results might lead to a procedure for characterizing networks relative to the value of this flow center. The second example problem found such a flow center with the additional constraint that $v(N_i;s) = v(N_i;t)$.

APPENDIX A

"SINGLE-COMMODITY, MAX-FLOW" ALGORITHM

The following is the Ford and Fulkerson algorithm for the construction of maximal single-commodity flows.

Step 1. (Labeling Process). Every node is always in one of three states: Unlabeled, Labeled and unscanned, Labeled and scanned. Initially all nodes are unlabeled. Assign the label $[-,\varepsilon(s)=\infty]$ to s. Node s is now labeled and unscanned. To all unlabeled nodes N_k connected to N_j by an edge, assign the label:

(i) $[N_j^+, \varepsilon(N_k)]$, if $0 \le f(N_j,N_k) < c(N_j,N_k)$

(ii) $[N_j^-, \varepsilon(N_k)]$, if $f(N_k,N_j) > 0$

where $\varepsilon(N_k) = \mathrm{MIN}[\varepsilon(N_j),\ f(N_k,N_j)+c(N_j,N_k)]$. (Note that a negative flow in an edge means that the flow is in the opposite direction to the original flow in that edge.) When all N_k have been labeled, N_j is considered to be scanned and may be disregarded during the remainder of this step. Continue this process until:

(i) Node t is labeled. Go to Step 2.

(ii) No additional labels can be assigned and node t is not labeled. Stop. The current flow is optimal.

Step 2. (Flow Change). Starting at t replace $f(N_j,N_k)$ by $f(N_j,N_k)+\varepsilon(t)$, if N_k is labeled $[N_j^+, \varepsilon(N_k)]$ or replace $f(N_k,N_j)$ by $f(N_k,N_j)-\varepsilon(t)$, if N_k is labeled $[N_j^-, \varepsilon(N_k)]$. Continue this process until node s is reached. Remove all labels and return to Step 1.

(Note that this algorithm computes $v^*(s;t)$. If maximal flow between a different source and sink is desired, appropriate substitutions must be made.)

APPENDIX B

FLOW DECOMPOSITION ALGORITHM

The following is a modification of the Ford and Fulkerson algorithm for the chain decomposition of a feasible flow.

Let $f(N_i,N_j)$ denote the flow through edge (N_i,N_j) in the maximal single-commodity flow pattern obtained using the algorithm given in Appendix A. Let $f^1(N_i,N_j)$ denote a flow through edge (N_i,N_j) in a flow pattern from s" through s to a. Let $f^2(N_i,N_j)$ denote a flow through edge (N_i,N_j) in a flow pattern from a through t to s".

Step 0. Set $f^1(N_i,N_j) = 0$ and $f^2(N_i,N_j) = 0$.

Step 1. (Identify Chain Flow 1). Assign label, $[s'',\varepsilon(s)]$, to node s where $\varepsilon(s) = [f(s'',s)-f^1(s'',s)]$. To a single unlabeled node N_k for which $f(N_j,N_k)-f^1(N_j,N_k)>0$ assign the label, $[N_j,\varepsilon(N_k)]$, where $\varepsilon(N_k) = \mathrm{MIN}[\varepsilon(N_j),\ f(N_j,N_k)-f^1(N_j,N_k)]$. Continue to assign labels until a has been labeled. Go to Step 2.

Step 2. (Flow Change). Replace $f^1(N_j,N_k)$ by $[f^1(N_j,N_k)+\varepsilon(a)]$ for all edges in the chain flow from s" to a formed in Step 1. If $f^1(s'',s)\neq f(s'',s)$, remove all labels and return to Step 1. Otherwise, go to Step 3.

Step 3. (Identify Chain Flow 2). Assign label, $[s'',\varepsilon(t)]$, to node t where $\varepsilon(t) = [f(s'',t)-f^2(s'',t)]$. Node t is now labeled. In general select the last labeled node N_j. To a single unlabeled node N_k for which $[f(N_j,N_k)-f^1(N_j,N_k)-f^2(N_j,N_k)]>0$, assign the label

$[N_j, \varepsilon(N_k)]$, where $\varepsilon(N_k) = \mathrm{MIN}[\varepsilon(N_j),\ f(N_j,N_k)-f^1(N_j,N_k)-f^2(N_j,N_k)]$. Continue to assign labels until a has been labeled. Go to Step 4.

Step 4. (Flow Change). Replace $f^2(N_j,N_k)$ by $f^2(N_k,N_j) = f^2(N_j,N_k)+$ (a) for all edges in the chain flow from s" to a formed in Step 3. If $f^2(s'',t) \neq f(t,s'')$, remove all labels and return to Step 3. Otherwise stop. The flow has been decomposed.

BIBLIOGRAPHY

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

2. Hu, T. C., "Multicommodity Network Flows," Operations Research, 11, 1963.

3. Rothfarb, W., N. P. Shein and I. T. Frisch, "Common Terminal Multicommodity Flow," Operations Research, 16, 1968.

4. Rothschild, B. and A. Whinston, "On Two Commodity Network Flows," Operations Research, 14, 1966.

5. Arinal, J., "Maximal Biflow in an Undirected Network," IBM Journal of Research and Development, 13, 1969.

6. Rothschild, B. and A. Whinston, "Maximal Two-Way Flows," SIAM Journal of Applied Mathematics, 15, 1967.