A PRIMAL-DUAL MULTI-COMMODITY FLOW ALGORITHM [†]

by

William S. Jewell
Operations Research Center
University of California, Berkeley

California University, Berkley.

September 1966

ORC-66-24

[†] Research was originally supported under a Grant-in-Aid from the Union Carbide Corporation and by the Army Office of Ordnance Research under Contract DA-19-020-ORD-2684 with the Center for Operations Research at the Massachusetts Institute of Technology from 1955 to 1958.

Preparation and reproduction of this report was supported by the Army Research Office Grant Number DA-31-124-ARO-D-331 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

ABSTRACT

The author's 1958 multi-commodity flow algorithm is reproduced, together with historical and technical comments.

"A Primal-Dual Multi-Commodity Flow Algorithm," ORC 66-24

Introduction

Recently, renewed interest has developed in constructing special solution algorithms for multi-commodity network flow problems. The classic reference is that of Ford and Fulkerson (84)(F)[†] ; at about the same time, the author presented an independent proposal in a thesis report (P) which received only limited distribution. Because the same idea -- constructing a loop-arc incidence matrix -- was basic to both proposals, this portion of the thesis was never submitted for publication.

However, the increasing number of recent papers in this area (DD)(Y)(A) suggests that there might be interest in giving wider circulation to the original algorithm, particularly as certain "folklore" of that period keeps on being rediscovered.

Thus, the main part of this report is an exact reproduction of Chapter IV and Appendix C of (P). Although some of the terminology and explanations are now outdated, it seemed preferable to reproduce the original version, adding historical comments and interpretation, rather than attempting a revision.

Historical Remarks [††]

From 1955-1958, a group of faculty (D. N. Arden, D. A. Huffman, S. J. Mason, W. K. Linvill) and students (J. B. Dennis, the author, and others) at the Massachusetts Institute of Technology became interested in the pioneering research on network flow

[†] Numbers refer to the original bibliography in (P), reproduced at the end of this report; letters refer to supplementary references which follow.

[††] J. B. Dennis and D. R. Fulkerson have graciously reviewed this section. However, since a history must of necessity be a personal viewpoint, only the author is responsible for omissions or errors of fact.

problems being done at the RAND Corporation by Dantzig, Ford, Fulkerson, Robacker, and others (See (6)(15)(22)(27)(28)(30)(31)(32)(34)(51)(52)(74)(75)(84)). At first, our interest was in computational results as they might apply to multi-stage transportation models and computer codes which were being constructed under a grant-in-aid from the Union Carbide Corporation (47). However, as various theoretical results and algorithms (such as the max-flow, min-cut theorem (15)(22), the labelling methods for maximal flow (28)(29)(34), and the Hungarian method (59)(60) applied to transportation problems (29)(30)) became available, a Signal Nets Seminar was organized in 1956 to discuss these topics and stimulate research. The author can recall Professor D. N. Arden demonstrating how the labelling techniques for optimal flow could be carried out directly "on the network," instead of in a matrix formulation (31) using Orden's transhipment device (69). Professor S. J. Mason constructed a diode, current - and voltage-source analogue to the optimal flow problem, thus relating networks with capacity-restricted flow to electrical networks; I believe he should be credited with the first use of the complementary-slackness (voltage-current) diagram in the United States, although Sunaga and Iri(Z) also recognized the parallelism and later used these diagrams effectively (O), as did Dennis (C)(D) and the author (P). The fact that the Ford-Fulkerson algorithm was a <u>cheapest</u> (incremental) <u>flow-augmenting procedure</u>, whereas the simplex procedures, (such as the stepping-stone method), were cost-saving <u>flow-rerouting</u> methods, was also known to this group.

From this seminar came much interest in the area of network models and, ultimately, two theses (P) and (C). The author became interested in making two different extensions to the simpler network flow models: adding multipliers to the arcs, and allowing simultaneous (multi-commodity) flows. The first topic was stimulated by a paper of Markowitz (64), and the second by the communication models of Kalaba and Juncosa (51)(52); helpful discussions took place at the RAND Conference on Computational Aspects of Linear Programming, Santa Monica, August, 1956.

Research progress on these two extensions was reported in a series of unpublished papers (46), and at two meetings of the Operations Research Society of America (48)(Q). Due to the limited understanding of the nature of linear programs at that time, every new solution algorithm or model variant appeared unrelated to previous work, and progress was uneven. For example, the author originally made a false conjecture on the integrality of multi-commodity network flow solutions, which was cleared up by a counterexample of D. R. Fulkerson and R. E. Kalaba. However, by Fall, 1957, the author was successfully solving multi-commodity and gain problems by hand; what was lacking was a solid theoretical foundation for these procedures.

This foundation was provided by the results of further research at the RAND Corporation -- the primal-dual algorithm of Dantzig, Ford, and Fulkerson (20), which generalized the transportation algorithm of (31) to arbitrary linear programs. By specializing this general method back to the gain and multi-commodity flow models, it became obvious that the essential subroutine was one of maximal-flow in a restricted subnetwork, and that further effort should be directed to finding efficient maximal-flow procedures. For networks with gains, the idea of carrying along products of the gains in the labelling procedure to find a flow-absorbing loop was developed by the author in March 1958. The realization that a general linear programming subroutine would have to be used for multi-commodity flows, and the idea of using a loop-arc incidence matrix to simplify this computation was conceived in April 1958. The final form of both algorithms was presented at the Boston Meeting of ORSA in May 1958 (Q), and the thesis (P) appeared in June 1958.

In their 1954 paper (28), Ford and Fulkerson used a loop-arc incidence matrix to describe single commodity maximum flows problem; they also formulated the multi-commodity problem, and showed that the min-cut, max-flow theorem did not generalize in a simple way. As a result of research during 1957-1958, they proposed a multi-commodity algorithm based on (what would now be called) a "decomposition" approach, rather than a "primal-dual" one. The method was described to the author at the Boston Meeting and the research paper (84)(F) appeared shortly thereafter with a dateline of March 1958. As mentioned above, since the loop-arc incidence formulation was central to both papers, the author's algorithm was never submitted for publication, although the work on networks with gains later appeared in revised form in (R).

The other product of the research effort at M.I.T. was a thesis by J. B. Dennis, which appeared in book form (C). Here the possibilities of using electrical analogues to construct algorithms for ordinary flow problems, as well as for general linear and quadratic programs, were exploited in great detail. Considering its timeliness, it is surprising how little known are Dennis's methods; for example, he made the first extended use of the complementary-slackness diagram to explain algorithms, and developed the essentials of an isolated-component ("black-box") approach, which is related to the out-of-kilter method (E). A further description of this work may be found in (U).

Since that time, many different people have contributed to the literature on multi-commodity problems, and it is difficult to put their work into a proper historical context because of their diversity. For example, we have the continued work of Haley (I , J, K, L) on the multi-index transportation problem; the primal-dual formulation of "bundle" constraints by Matthys (V); a "distributed-algorithm" proposal of Dennis (D); and a "nonlinear exchange" method of Sakarovitch (Y).

Tomlin (DD)and Bradley (A) related the multi-commodity proposal of Ford and Fulkerson (84) to the decomposition method of Dantzig and Wolfe (B), but apparently were not aware of the strong historical connection between the two methods. Finally, in a slightly different direction, we have the undirected arc "communication network" problems, with the interesting "bi-flow" algorithm of Hu (M), and various results of Hakimi (H), Hu (N), Tang (AA,BB,CC), Rothschild and Whinston (W, X) , etc.. The problem of synthesis has been tackled by Gomory and Hu (G).

Given that 10 years have elapsed since the first multi-commodity proposals were made, it is surprising that no definitive paper has yet appeared. To the author's knowledge, no explicit multi-commodity computer codes are currently available, problems of this type usually being solved by a general simplex routine, or by a decomposition code; thus no experimental statistics on the efficiency of various proposals is available. Even the nature of the rational solution to multi-commodity problems is not yet well understood (S).

Technical Remarks

The original thesis report (P) consisted of four chapters, with appendices containing proofs and various computational remarks. Chapter 2 was on "on-the-network" description of the Ford-Fulkerson algorithm (31), with emphasis on the physical interpretation described earlier; Chapter 3 was presented in expanded and revised form in (R). For this report Chapter 4, Appendix C, and the Bibliography have been reproduced.

Most of the algorithm could, of course, be rewritten in a simpler and more condensed form in light of current knowledge about linear programs. For example, the assumption of nonnegative costs and zero lower bounds is not at all essential to the algorithm. In circulation form, the maximal flow subroutine can be explained solely in terms of loop flows; the "Direct-Flow" phase then becomes an incremental change in some uncoupled loop flow variables. It is also easy to imagine various combinations of the Ford-Fulkerson proposal with this one, say, by using their min-cost pricing-out procedure to add entries to the loop-arc incidence matrix.

In fact, using the COMPLEX approach (U), it is now possible to produce a multi-commodity algorithm which would start with an arbitrary solution, and work on the network in an "out-of-kilter" manner; as always, the key subroutine is an incremental maximal flow problem in a restricted subnetwork.

The remaining problem, then, is not a conceptual or procedural one, but is one of computational efficiency. Simple network flow problems are efficient because of the serial manner in which labelling and flow augmentation is carried out. On the other hand, the multi-commodity problems have grave difficulties in that many such labellings must be made at each iteration, or a full-size inverse must be carried along. Even proper representation of multi-commodity network topology presents special problems in data storage and linkage.

It is the author's hope that reproduction of this algorithm will stimulate work on the computational aspects of the multi-commodity problem, and perhaps discourage further development of algorithmic variations.

(The Primal-Dual Algorithm)

Chapter IV. Multi-Commodity Flow

1. Introduction to Multi-Commodity Flow

The last two chapters have considered flow of identical items through a network, or by using gains, the flow of different items in different parts of a network. In determining the optimal routing pattern, each output branch didn't care which input furnished the flow, since all flows satisfied the output requirements.

However, the assumption of a single kind of flow becomes impossible when several different flows are available, and the input-output requirements are different for each type. For example, in any communication network, the flow of letters, telephone calls, etc. is a multi-commodity flow, since each message has a unique origin and ultimate destination. Another example is that of a soup canner who has plants producing and storing certain varieties, but whose customers want product mixes of all the various soups.

The reason that the different kinds of flows interfere with one another in a multi-commodity network is that they may share the same branch -- i.e., a common warehouse, the same relaying center, etc. Thus the limitations of a branch's total carrying capacity may impose a mutual flow capacity restriction on all the flows through the branch; clearly there may still be individual flow capacities for the different kinds of flow through a network.

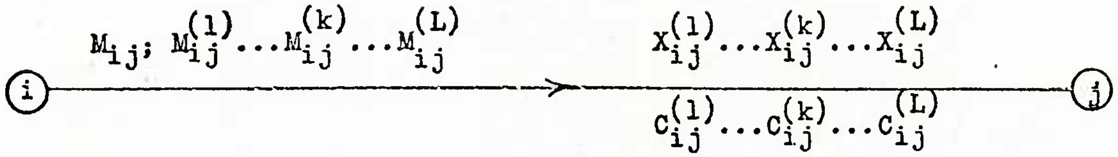
As before, it is assumed that the disutility of establishing flow is directly proportional to the amount of flow in each branch; but, it is now possible for the per-unit costs of the different kinds of flows to be different, even in a "shared" branch. An example is the routing of different priority messages through the same communications net, where it is possible to assign different numerical utility values to the different priorities. Figure IV.1 illustrates the idealized branch that will be considered in formulating an algorithm for multi-commodity flow. If certain types of flow are forbidden (by policy restrictions, physical limitations, etc.) to traverse certain branches in the multi-commodity network, then one may make the corresponding individual capacities zero.

Conservation of flow (for each type of flow individually) is still assumed to hold at the nodes; in other words, only the topology of the network and the mutual flow capacity are shared in common. If all of the mutual constraints were removed, then one could separate $\underline{L}$ distinct images of the network (one for each flow type), and solve each one individually.

Conceptually, it is easier to visualize the algorithm if one assumes that this "separation into layers" is actually made, as in Figure IV.3. By introducing additional feeder branches, one can introduce all of the flow at the common source $\underline{O}$, and remove it at the common sink $\underline{R}$; the placement and individual capacities of the feeder branches within each "layer" assure the proper production and consumption of each type of flow. Aside from these feeder branches, each layer is a replica of the original network.

What about the mutual capacity restriction, M_{ij}? This constraint now introduces a coupling between flows in the different layers such that the sum of flows on topologically similar branches is limited. This feature will introduce complications into any optimal flow algorithm for a multi-commodity network.

A problem of transporting several products simultaneously was first proposed by Schell (79); Charnes and Cooper have formulated a warehousing problem with different commodities (8). Kalaba and Juncosa (51)(52) first considered the problem of routing communication messages by linear programming;


M_{ij}; $M_{ij}^{(1)} \ldots M_{ij}^{(k)} \ldots M_{ij}^{(L)}$
$X_{ij}^{(1)} \ldots X_{ij}^{(k)} \ldots X_{ij}^{(L)}$
i
$C_{ij}^{(1)} \ldots C_{ij}^{(k)} \ldots C_{ij}^{(L)}$
j

$i,j = 1,2,\ldots N$

$k = 1,2,\ldots L$

$X_{ij}^{(k)}$ - non-negative flow of type k in (i,j)

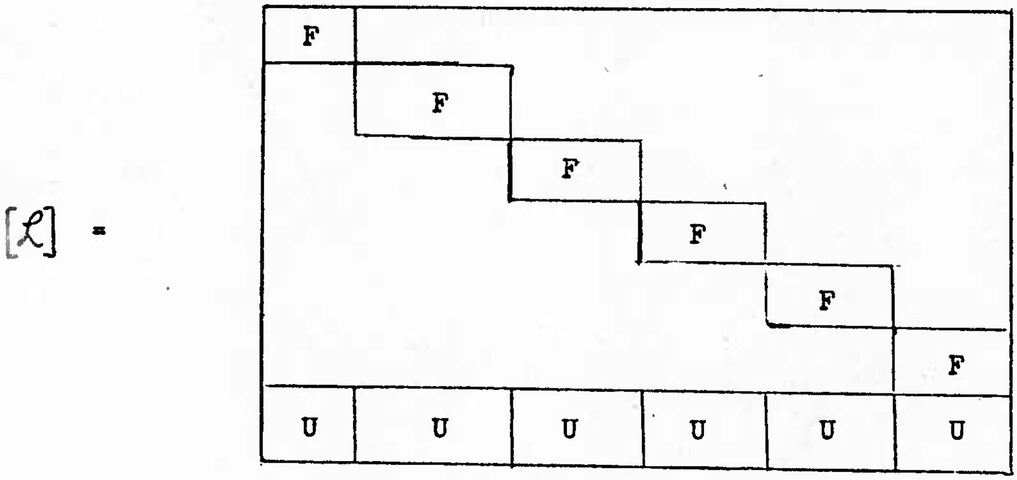
$M_{ij}^{(k)}$ - individual flow capacity of type k

M_{ij} - mutual flow capacity of branch (i,j)

$C_{ij}^{(k)}$ - per-unit cost of type k flow

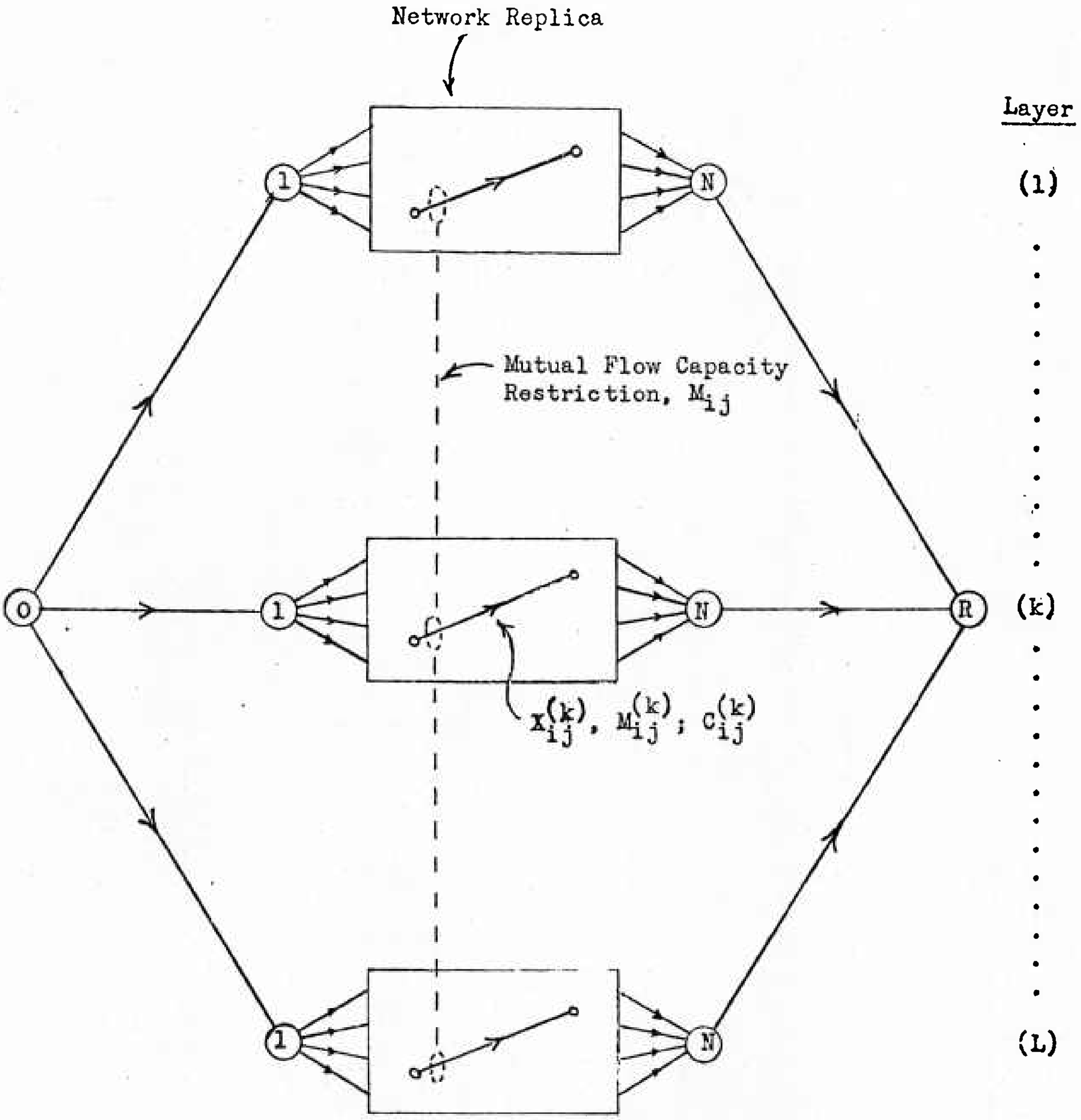
A Multi-Commodity Branch

Figure IV.1



Constraint Matrix for Multi-Commodity Flow

Figure IV.2


Network Replica
Layer
(1)
(k)
(L)
Mutual Flow Capacity Restriction, M_{ij}
$x_{ij}^{(k)}, M_{ij}^{(k)}; c_{ij}^{(k)}$
0
1
N
R

Separation of Multi-Commodity Network
into Single-Commodity Layers

Figure IV.3

Robacker (75) produced a non-constructive min-cut, max-flow theorem for multi-commodity flows. In these references, solution was indicated by using the revised simplex method.

Ford and Fulkerson have recently (84) proposed a computation for maximizing multi-commodity flows which also uses the idea of an incidence matrix (see Section 3). Their treatment uses this formulation within the simplex method, with several applications of the "least-cost route" algorithm to determine the new chain from source to sink to enter the basis. Labelling is not used to find the chains, and thus none of the "easy part" (Section 4) of the maximal-flow subroutine is done first.

2. Multi-Commodity Flow and Linear Programming

The optimization problem to be considered is:

> What routing of each type of flow will minimize the total cost of all flows, and satisfy the individual and mutual restrictions, for given inputs and outputs of each flow type?

Assuming that there are L different types of flow, whose flow through branch (i,j) is denoted by $X_{ij}^{(k)}$ (k = 1,2,...L), and that appropriate feeder branches distribute the flow correctly (as in Figure IV.3), one obtains the following primal and dual formulations to the multi-commodity flow problem:

PRIMAL

$$\text{Min } \mathcal{C} = \sum_{i,j,k} C_{ij}^{(k)} X_{ij}^{(k)} \qquad (2.1)$$

$$\sum_{j} (X_{ij}^{(k)} - X_{ji}^{(k)}) = 0 \qquad (2.2)$$

$$\sum_{j,k} (X_{0j}^{(k)} - X_{j0}^{(k)}) = +Q$$

$$\sum_{j,k} (X_{Rj}^{(k)} - X_{jR}^{(k)}) = -Q$$

$$X_{ij}^{(k)} \leq M_{ij}^{(k)}$$

$$\sum_{k} X_{ij}^{(k)} \leq M_{ij}$$

$$X_{ij} \geq 0$$

DUAL

$$\text{Max } \mathcal{C} = Q(V_0 - V_R) - \sum_{i,j,k} M_{ij}^{(k)} U_{ij}^{(k)} - \sum_{i,j} M_{ij} U_{ij} \qquad (2.3)$$

$$V_i^{(k)} - V_j^{(k)} - U_{ij}^{(k)} - U_{ij} \leq C_{ij}^{(k)} \qquad (2.4)$$

$V_i^{(k)}$ unrestricted

$$U_{ij}^{(k)} \geq 0$$

$$U_{ij}^{(k)} \geq 0$$

where $C_{ij}^{(k)}$, $M_{ij}^{(k)}$, M_{ij}, and Q are given, non-negative constants.

The constraint matrix for the multi-commodity network is shown in Figure IV.2. It consists of flow sub-matrices (F) (see Figure III.2), for each type of flow, coupled together by unit matrices (U) which represent the mutual flow restrictions in topologically similar branches. As will be seen, the algorithm actually works for coupling between any subset of the branches in any simple flow network; in this case the coupling matrices U would be arbitrary matrices with zeroes and ones.

The simple structure of the dual is retained for each branch and each type of flow, except for the coupling term, U_{ij}, which must be the same for all branches sharing the same mutual restriction. Since this mutual slack potential U_{ij} may not be non-zero unless the mutual restriction is saturated one would expect that the early solution stages to a "loosely coupled" problem would be the same as L simple flow problems solved simultaneously.

One may more easily see what happens by considering the complementary slackness conditions;

$$\text{If } V_i^{(k)} - V_j^{(k)} < C_{ij}^{(k)} + U_{ij} \qquad (U_{ij} \geq 0)(U_{ij}^{(k)} = 0) \qquad (2.5a)$$
$$\text{then } X_{ij}^{(k)} = 0$$

$$\text{If } V_i^{(k)} - V_j^{(k)} = C_{ij}^{(k)} \qquad (U_{ij} = 0)(U_{ij}^{(k)} = 0) \qquad (2.5b)$$
$$\text{then } 0 \leq X_{ij}^{(k)} \leq \min(M_{ij}^{(k)}; M_{ij} - \sum_k X_{ij}^{(k)})$$

$$\text{If } V_i^{(k)} - V_j^{(k)} = C_{ij}^{(k)} + U_{ij} \qquad (U_{ij} > 0)(U_{ij}^{(k)} = 0) \qquad (2.5c)$$
$$\text{then } X_{ij}^{(k)} = M_{ij} - \sum_{\ell \neq k} X_{ij}^{(\ell)}$$

$$\text{If } U_{ij}^{(k)} > 0 \quad (V_i^{(k)} - V_j^{(k)} = C_{ij}^{(k)} + U_{ij}^{(k)} + U_{ij}) \quad (U_{ij} \geq 0) \qquad (2.5d)$$
$$\text{then } X_{ij}^{(k)} = M_{ij}^{(k)}$$

$$\text{If } U_{ij} > 0 \quad (\text{and for at least one } k \varepsilon L \; (V_i^{(k)} - V_j^{(k)} = C_{ij}^{(k)} + U_{ij}; \; U_{ij}^{(k)} = 0) \qquad (2.5e)$$
$$\text{then } \sum_k X_{ij}^{(k)} = M_{ij}$$

The last conditions (2.5e) are of particular interest, since they represent a new type of complementary slackness. If a mutual slack potential is greater than zero, and several branches sharing that potential are active, it is possible to change the individual flows, provided that one increases flow in one layer, while decreasing it in another; in other words, individual flows may change when $U_{ij} > 0$, provided that one "trades off" the mutual capacity restriction. This novel feature will introduce a new procedure into the maximal-flow subroutine where more complicated trade-off possibilities must be handled.

It may help to visualize the role of the mutual slack potential if one considers the electrical analogue. To incorporate mutual restrictions into the model considered previously, one must add 1:1 "D.C. transformers" into each branch in each layer of the network sharing the same common restriction; the secondaries of each transformer are connected in parallel across a diode-current-source loop which limits the total flow of all the secondaries (and hence the primaries) to M_{ij} units. In the case that all the flow from the current source is backed up, a common potential would appear across the diodes, and in every branch sharing the common restriction.

3. Maximal Flow in a Multi-Commodity Network

The most important part of the optimization algorithm to be described is a maximal flow subroutine, which will find the maximum increase in flow in a restricted multi-commodity network.

In the case of simple networks, and flow with gains, it has been seen that this subroutine involves a labelling procedure which traces out paths which can absorb more flow from the source. To the extent that mutual capacity restrictions do not affect the optimal choice of a route for a particular type of flow, one may expect the maximal flow subroutine for many flows also to be a labelling method.

On the other hand, if this problem were solved using the primal-dual method, this subroutine could be as general as solving a linear programming problem by the revised simplex method; this might be expected in the case that the mutual capacity restrictions were extremely tight. Nevertheless, one would hope that the flow part of the problem -- the conservation equations -- would still give a simpler approach to the problem than attempting to solve the subroutine by the simplex method.

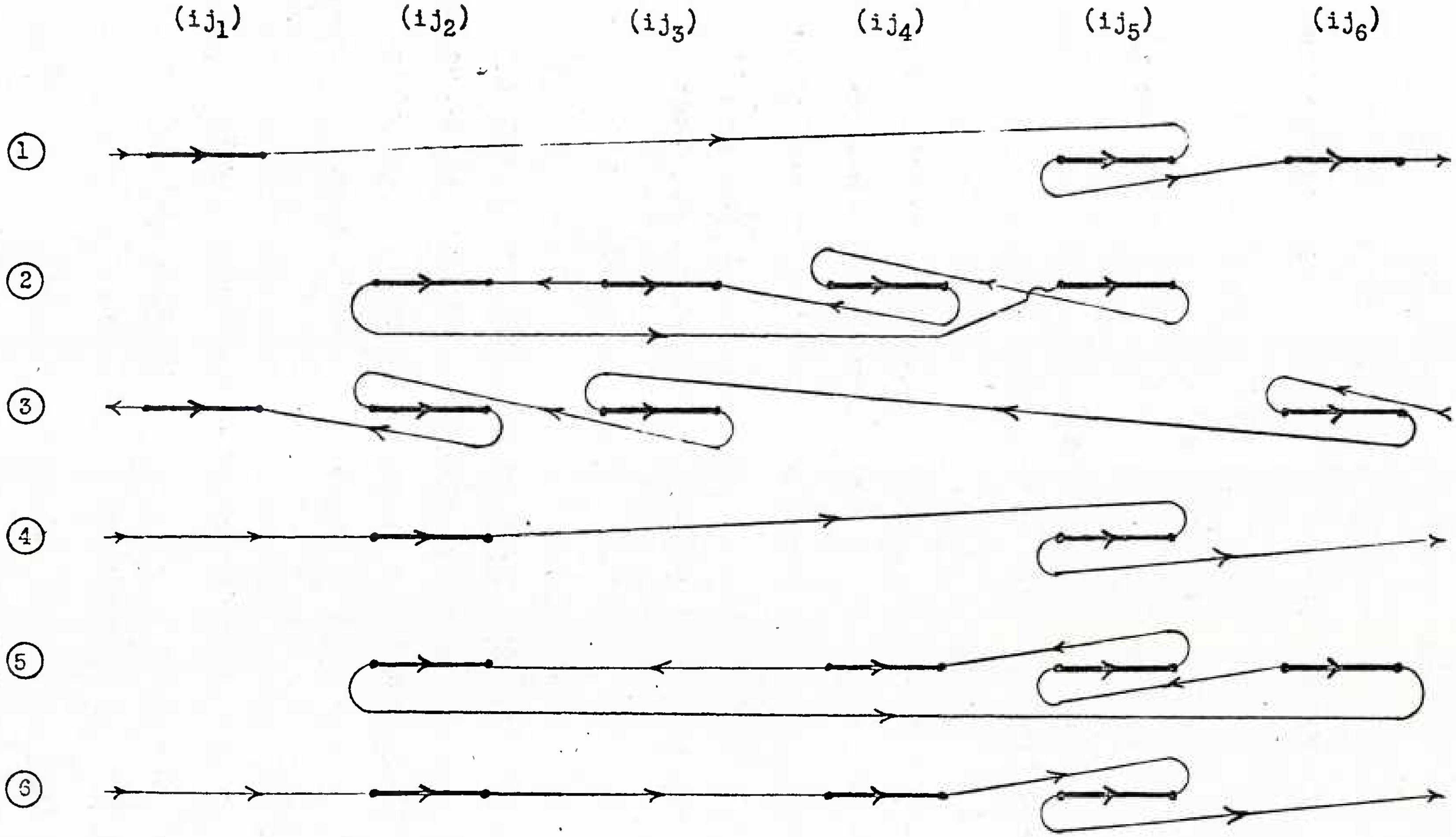
In the maximal flow subroutine to be described in Section 5, emphasis is first placed on doing as much of the "easy" increasing of flow as is possible; in many cases, and especially during the early cycles of the optimization algorithm, this partial procedure is all that is needed. Then, if necessary, the second part of the subroutine is used for the "l.p.-like" part of the increase in flow, where mutual capacity must be exchanged between different types of flow.

Therefore, in the first part (the Direct Flow Phase), one considers only the changes in the variables that would be made in a simple flow problem. Starting from any feasible flow pattern, one investigates all active branches which can individually have their flow increased. If a chain of these branches is found from source to sink, then one can increase the direct flow. This Phase is repeated until all such increases have been made.

In the second part of the subroutine (the Exchange Flow Phase), one attempts to increase flow into the network by exchanging some of the mutual capacity restrictions between the different layers. Since one can envisage an exchange involving many layers and different mutually saturated branches, this exchange could be quite complex; in general, a linear programming problem does result. However, the procedure to be described reduces the problem as much as possible before resorting to the simplex method; in any case, one needs to use this part of the procedure only infrequently during the solution of a problem, and the resulting linear programming problem condenses the restricted primal problem so as to consider only the meaningful changes in the flow.

The main feature of the exchange flow phase is that a table is constructed, which is essentially an incidence matrix, showing how forward chains (which increase total flow in the network), backward chains (which decrease total flow), and loops (which leave total flow unchanged) are "incident" upon the various mutual restrictions of branches which have positive mutual slack potentials. This incidence matrix defines a set of equalities which restrict the changes to be made in the flows in chains and loops in the various layers of the network so that the mutual restrictions are still met. The procedure to be described uses this table to eliminate all possibilities of exchange which are clearly not advantageous (if no forward chains exist, for instance). If elimination does not simplify the problem (it usually does), then one must solve a small linear programming problem, which determines the increases in flow in the chains and loops which maximize the total flow increase into the network, subject to the "exchange of mutual restriction" condition. The limiting factor on this increase is usually the saturation or emptying of some branch, so that the table to be described also carries along these additional restrictions.

An example should clarify the use of the incidence table (Step 6). Consider Figure IV.4, which shows the skeleton of a network which might be encountered in a maximal-flow subroutine for a multi-commodity network. Each row represents some chain or loop of branches whose flow may be changed; the heavy branches are those which are mutually saturated; those branches which share the same mutual flow capacity are in the same column -- for example, branch $(ij)_3$ is in loop ② and chain ③. The arrows on the light portions of each chain or loop indicate the direction in which flow may be increased. Assuming that the source is to the left of the diagram, rows ①, ④, and ⑥ represent forward chains, row ③ represents a backward chain, and ② and ⑤ are loops. With respect to "incidence", it can be seen that an increase in flow through chain ⑥ would (a) increase total flow, (b) tend to increase the mutual sum of flow in branches $(ij)_2$ and



Example of Exchange Flow Phase of Maximal Flow
Subroutine for Multi-Commodity Flow

Figure IV.4

and $(ij)_4$, and (c) would tend to decrease the mutual sum of flows through branch $(ij)_5$. Similar statements hold for the other rows.

If one denotes the increase in the flow in configuration r ($r = 1,2,\ldots 6$) by δ_r then the appropriate capacity exchange equations read:

$$\begin{array}{rrrrrrl} \delta_1 & & -\delta_3 & & & & = 0 \\ & -\delta_2 & +\delta_3 & +\delta_4 & -\delta_5 & +\delta_6 & = 0 \\ & -\delta_2 & +\delta_3 & & & & = 0 \\ & \pm\delta_2 & & & -\delta_5 & +\delta_6 & = 0 \\ -\delta_1 & +\delta_2 & & -\delta_4 & +\delta_5 & -\delta_6 & = 0 \\ \delta_1 & & +\delta_3 & & -\delta_5 & & = 0 \end{array} \tag{3.1}$$

Because of effect of the various increases in flow in the different configurations one would like to maximize the total increase in flow into the network, which is $\delta_1 - \delta_3 + \delta_4 + \delta_6$. By the elimination procedure of the subroutine, however, one finds that $\delta_1 = \delta_2 = \delta_3 = \delta_4 = \delta_6 = \frac{1}{2}\delta_5$ is the only possible solution. Other constraints not shown actually limit the flow.

4. Discussion of the Optimal Flow Algorithm for Multi-Commodity Networks

Having discussed the salient ideas of the maximal-flow subroutine, we consider the basic features of the optimal flow algorithm. Many of its steps are the same as those discussed in previous chapters, so only the differences will be emphasized here.

With different commodities flowing, we must add to our terminology in order to completely describe the state of each branch. Using the layer formulation to separate the different commodities, a branch carrying no flow (of a single type) is said to be empty; if the flow is at its upper bound, the branch is individually saturated; and if several branches, sharing the same mutual capacity restriction, have the sum of their flows equal to this upper bound, then one speaks of the branches as mutually saturated. For convenience in the algorithm it is assumed that a branch is not individually and mutually saturated at the same time. This is not a loss of generality, since one can separate any such branch into two branches in series, and thus resolve any ambiguity.

Using the dual inequalities, one can again describe an individual branch as inactive, active, and overactive, depending on the slackness or tightness of the constraints (2.4). One new state is needed, however, to describe the set of branches for which the mutual slack potential, U_{ij}, is positive; these branches may be described as coupled. The complementary slackness conditions (2.5) become:

> If an individual branch is inactive, it is empty.
>
> If an individual branch is active and uncoupled, it may have any feasible flow.
>
> If an individual branch is overactive, it is individually saturated.
>
> If a group of branches is coupled, at least one of the branches is active, and the group is mutually saturated.

The last condition provides the interesting possibility that the mutual capacity restriction may be "exchanged" between coupled branches in order to increase total flow into the network, provided that the branches are kept mutually saturated; a systematic procedure for doing this was described in the last section. Occasionally, the optimal flow algorithm detects a change of state called "decoupling", when the mutual slack potential is decreased to zero from some positive value.

The algorithm to be presented in Section 6 proceeds in much the same way as the other network solutions. The first step starts the procedure by finding a set of feasible potentials. Since all of the unit costs are assumed non-negative, setting all of the potentials equal to zero gives a feasible dual; or, one may find the least-cost path from source to sink through the network of Figure IV.3, and use the resulting potentials to get started.

The second step identifies Set A as the set of indices of overactive branches, Set B as the set of indices of coupled branches, set C as the set of active (coupled or uncoupled) branches, and Set D as the union of all these sets. The complement of Set D is, of course, the set of indices of inactive branches. The algorithm then identifies a restricted network, in which all inactive branches are to be kept empty, all overactive branches are to be kept individually saturated, and all coupled branches are to be kept mutually saturated. The last condition provides a great deal of latitude.

The restricted primal problem then requires one to maximize flow into the restricted network, keeping as many of the usual restrictions satisfied as possible. As before, one can arrange matters so that only the input (and output) flow requirement is not met; one can again think of the primal problem as using feasible potentials and complementary slackness to guide the restricted primal's attempt to reduce infeasibility in the flows.

In Step (3), the maximal flow procedure of Section 5 is used to solve the restricted primal and its dual. This subroutine first explores all active, uncoupled branches using the simple labelling procedure of Chapter II to see if these is a "direct" way to send more flow from source to sink through any layer; it proceeds to establish as much of this flow as is possible. Then the active, coupled branches are added, and one considers how to change the flow in coupled branches so as to keep them mutually saturated, yet provide new paths of active branches to increase the total flow through the network. One constructs an incidence table by using some very simple labelling techniques to find all chains and loops of active branches; the table then describes the effect of an increase in flow along a chain or loop upon a mutual restriction. Because the sum of such flow changes for a group of coupled branches must be zero, certain logical impossibilities may be eliminated immediately. In general, however, one may be required to solve a small linear programming problem.

In Step (4), one notes that if the input flow is the desired value Q, the algorithm is finished, since one has feasible flows and potentials, and the conditions of complementary slackness are satisfied.

If the total input to the network is still too small, Step (5) looks for a way to change the potentials so that they remain feasible, yet increase the dual functional, and add at least one new active, uncoupled branch to the restricted network.

As in Chapter III, it is not easy to describe the relative changes in potential which occur at this step. The difficulty is not due to gains in the dual equations but because the mutual slack potentials are common to the dual restrictions of each group of coupled branches. The way to correctly evaluate these potential changes is to use the solutions to the dual of the restricted primal problem. This dual provides a set of potential increments $\sigma_i^{(k)}$, $\sigma_{ij}^{(k)}$, and σ_{ij} such that

$$\sigma_i^{(k)} - \sigma_j^{(k)} - \sigma_{ij}^{(k)} - \sigma_{ij} \leq 0$$

for all non-inactive branches, with equality holding for non-empty branches. The source potential increment, σ_0, is fixed at +1, so that all potential changes are expressed relative to an increase in the potential, V_0; the other node potential increments, $\sigma_i^{(k)}$, are unrestricted. The individual potential increment, $\sigma_{ij}^{(k)}$, is non-negative for branches which are not overactive, and is zero if a branch is not individually saturated. The mutual slack potential increment, σ_{ij}, is non-negative for groups of branches which are not coupled, and is zero if the group of branches is not mutually saturated.

With these relative changes in the potentials established, one then argues that it must be possible to increase the source potential by a finite amount ϑ, without violating the feasibility of the potentials (see Appendix D). The conditions on the potential increments assure one that active, non-empty, unsaturated (either way) branches remain in the same state; empty active branches may become inactive; individually saturated active branches may be overactive; and mutually saturated active branches may become coupled -- all without violating potential feasibility, or disturbing the complementary slackness with the flows, for <u>any</u> value of ϑ.

The limit on the increase of the source potential comes from three possible changes of state:

(1) An inactive branch becomes active. ("break-down")

(2) An overloaded branch becomes active. ("easing-off")

or,

(3) A group of active coupled branches decouples.

If none of these possibilities is present, the problem is infeasible, and is detected in Step (6) of the algorithm. Otherwise, one repeats the algorithm with the new feasible potentials, and a different restricted network with at least one new active, uncoupled branch. One also finds that the dual functional had a strict increase on the last cycle, and that the previous set of optimal flows is feasible in the new restricted primal problem.

The arguments for optimality are the same as those presented in Chapters II and III. One proves the maximal-flow subroutine by defining rules to generate the potential increments (Step 15, Section 5), noting that they solve the dual to the restricted primal (Appendix C) and obey the correct complementary slackness conditions. One thus proves that the restricted primal actually finds the maximal possible increase in flow into the network.

The reader is once more urged to reinterpret the mathematical statements of the following sections in terms of what is happening to the flows and potentials in the actual network. The flow-maximizing subroutine is complicated only because of the mutual coupling; the optimization algorithm is merely finding incrementally more costly ways of routing various types of flow through the network, until the desired input is reached in a finite number of steps or infeasibility is discovered.

5. The Maximal Flow Subroutine for Multi-Commodity Networks

The maximal flow subroutine is:

(1) Start with any set of flows, $X_{ij}^{(k)}$, satisfying the restricted primal problem (6.2) and (6.3). An "efficient" set is the optimal solution of the preceding cycle of the algorithm.

Direct Flow Phase

(2) Label source with $\lambda_0 = (/F_0 = +\infty/)$.

(3) Consider any node ik previously labelled with

$$\lambda_i^{(k)} = (hk\ /\ F_i^{(k)}\ /\ + \text{ or } -)$$

and any other node jk not previously labelled.

(a) If $(ijk \ \varepsilon\ C \cap \bar{B})$ and $(X_{ij}^{(k)} < M_{ij}^{(k)})$ and $(\sum_{k \varepsilon L} X_{ij}^{(k)} < M_{ij})$

label jk with:

$$\lambda_j^{(k)} = (ik\ /\ F_j^{(k)} = \min(F_i^{(k)};\ M_{ij}^{(k)} - X_{ij}^{(k)};\ M_{ij} - \sum_{k \varepsilon L} X_{ij}^{(k)})\ /\ +\ )$$

(b) If $(jik \varepsilon C \cap \bar{B})$ and $(X_{ji}^{(k)} > 0)$, label jk with:

$$\lambda_j^{(k)} = (ik \; / \; F_j^{(k)} = \text{Min } (F_i^{(k)};\; X_{ji}^{(k)}) \; / \; - \;)$$

(c) Otherwise, do not label node jk.

(4) If node R is labelled, go to Step (5), otherwise repeat Step (3), until R is labelled, or, no new labels are defined. In the latter case, go to Step (6).

(5) Node R has been labelled with a positive F_R in the label, and the flow into the network may be increased by a "direct flow" of this amount.
Starting at R, trace the labels back to the origin, adding F_R to $X_{ij}^{(k)}$ if the following label is encountered en route:

$$\lambda_j^{(k)} = (ik \; / \; F_j^{(k)} \; / \; + \;)$$

or subtracting F_R from $X_{ji}^{(k)}$ if the following label is met:

$$\lambda_j^{(k)} = (ik \; / \; F_j^{(k)} \; / \; - \;)$$

When the source is reached the flow into the network has been increased by F_R, and at least one branch has saturated or emptied. Erase all labels and repeat Step (2).

Exchange Flow Phase

(6) No new nodes can be labelled, and node R is not labelled. No direct flow can be sent from source to sink, unless an "exchange" in mutual flow capacity can be made between coupled branches.

Add to the branches ijk and jik ε $C \cap \bar{B}$ previously considered, all ijk and jik ε $C \cap B$, as additional candidates for labelling. If there are no ijk ε $C \cap B$, this phase is completed; go to Step (14).

The next few steps will consider a labelling procedure to establish chains and loops in all layers; an incidence table will be constructed, which describes the incidence of permissible chains and loops on the ij ε B, as well as on the other branches which make up the chains and loops. Figure IV.5 illustrates the form of the table; the chains and loops (indexed by r = 1,2,...) form the rows of the table. One column, headed F, will describe the possible increase in flow due to exchange. The set of headings over that portion of the table marked E are the set of all ij ε B. The set of columns marked L^+, are headed by ijk ε C. The set of columns marked L^- are headed by jik ε C.

				All ijk such that	
		ij ε B	F	ijk ε C	jik ε C
All chains and loops defined in (7)(8)(9) and (10)	1				
	2				
	.				
	.				
	r	$E_{r,ij}$	f_r	$L^{+}_{r,ijk}$	$L^{-}_{r,ijk}$
	.				
	.				

Form of Incidence Table

Figure IV.5

(7) Establish "forward chains" from <u>0</u> to <u>R</u>.

(a) Label source with $\lambda_0 = (//+)$.

(b) Consider any node ik previously labelled with

$$\lambda_i^{(k)} = (hk\ //\ + \text{ or } - \text{ or } Ex+ \text{ or } Ex-)$$

and any other node jk. Nodes may be labelled any number of times (except that looping situations are to be avoided in (7) and (8), and each new label may be continued according to the following rules:

(i) If $(ijk \ \varepsilon \ C \cap \overline{B})$ and $(X_{ij}^{(k)} < M_{ij}^{(k)})$

and $(\sum_{k \varepsilon L} X_{ij}^{(k)} < M_{ij})$

$$\lambda_j^{(k)} = (ik\ //\ +)$$

(ii) If $(jik \ \varepsilon \ C \cap \overline{B})$ and $(X_{ij}^{(k)} > 0)$

$$\lambda_j^{(k)} = (ik\ //\ -\ )$$

(iii) <u>$ijk \ \varepsilon \ C \cap B$</u>

$$\lambda_j^{(k)} = (ik\ //\ Ex+)$$

(iv) <u>$jik \ \varepsilon \ C \cap B$</u>

$$\lambda_j^{(k)} = (ik\ //\ Ex-)$$

(v) <u>Otherwise, do not label.</u>

(c) Repeat (b) until no new nodes are labelled. If node <u>R</u> is labelled (at least once), go to (d). If <u>R</u> is not labelled, go to (e).

(d) For each label on R, establish a new row (say, r) in the incidence table, and determine the set of branches which form a particular path from O to R by following *any* set of labels backwards. Note that if there are several ways to describe a connected network of chains, any description will suffice - provided that one avoids loops. For each labelling of:

Type (7bi) set $L^{+}_{r,ijk}$ equal to +1

Type (7bii) set $L^{-}_{r,jik}$ equal to -1

Type (7biii) set $L^{+}_{r,ijk}$ and $E_{r,ij}$ equal to +1

Type (7biv) set $L^{-}_{r,ijk}$ and $E_{r,ji}$ equal to -1

Set f_r equal to +1

Erase the labels thus followed, and repeat for each label on R, until all such forward chains are entered in the table. Go to Step (8).

(e) No increase in flow is possible; go to Step (14).

(8) Establish "backward chains" from R to O. Repeat Step (7), except start at node R and attempt to label node O. The entries in the incidence table are the same as in Step (7), except that f_r = -1 for each backward chain in the table.

(9) Establish "forward loops" around coupled branches.

(a) For some (k) ε L, start at node jk of a branch with ijk ε C∩B, and label as in Step (7) attempting to relabel this node via node ik and branch ijk, except do not label the source or sink.

(b) If node jk is not relabelled via branch ijk, repeat (a) for all different kεL. If no relabelling occurs, go to (e).

(c) If node jk is relabelled via branch ijk, establish a new row in the incidence table, with entries the same as in Step (7), except that f_r = 0.

(d) Repeat the labelling procedure (a) to find all such loops from jk to ik.

(e) Repeat (a) for another ijk ε C∩B, until all such mutually saturated branches have been examined.

(10) Establish all "backward loops" around mutually saturated

branches. Repeat Step (9), except begin at node ik and attempt to relabel the same node via node jk, labelling backwards across branch ijk. The entries are the same as in Step (9).

(11) The complete incidence table of (chains and loops) on (mutually) saturated branches) and (flow limiting branches) has been constructed, with at least one positive f_r. In order to find an increase in flow by exchange, it is in general necessary to solve a linear programming problem, with non-negative δ_r variables defined for each row.

First, the problem is reduced as much as possible by examining the entries $E_{r,ij}$. Reference should be made to Figure IV.5.

(a) If all entries in a column of E are zero, eliminate that column.

(b) If all entries in a column of E are non-negative, (or all non-positive), eliminate all rows with the positive (or negative) entries, setting the corresponding $\delta_r = 0$, and finally eliminating the column.

(c) If one entry (in row r_1, say) is +e (e = 1,2,...), and the only other non-zero in the same column of E (in row r_2, say) is -ne (n = 1,2,...), then $\delta_{r_1} = n\,\delta_{r_2}$. Add <u>n</u> times the entries in row r_1 to the entries of r_2 in the same column (for all columns), store in row r_2 and eliminate row r_1, and the column which had the two opposite sign entries.

(d) Similarly for the two non-zero entries (-e) and (+ne).

(e) Repeat (a) through (e) until:

either (i) There are no rows left. Set all $\delta_r = 0$. Go to Step (14).

or (ii) There is no positive entry under F. Set all $\delta_r = 0$. Go to Step (14).

or (iii) There is one row with all zero entries in E and a positive entry in F. Go to Step (12a).

(iv) There are two or more rows with arbitrary entries in E, and at least one positive entry under F. Go to Step (12b).

(12) The reduction of the incidence table in (11) did not indicate that a non-zero solution was not possible. In what follows, it is assumed that the reduced incidence table is being used; the symbols are the same as Figure IV.5, however.

(a) If there is only one row, r_1, with all zero entries in E, and f_{r_1} positive, set:

$$\delta_{r_1} = \underset{L^{+}\neq 0,\, L^{-}\neq 0}{\text{Min}} \left\{ \begin{aligned} &\left(\underset{L>0}{\text{Min}} \left[\frac{(M_{ij} - \sum_{k\varepsilon L} X_{ij}^{(k)};\; M_{ij}^{(k)} - X_{ij}^{(k)})}{L^{+}_{r_1,ijk}} \right] \right) && (5.1a) \\ &\underset{L<0}{\text{Min}} \left(\frac{X_{ij}^{(k)}}{-L_{r_1,ijk}} \right) && (5.1b) \end{aligned} \right\}$$

and the other δ_r follow from the reduction procedure. Go to Step (13).

(b) If there are two or more rows with arbitrary entries in E and one or more entry in F is positive, it is necessary to solve a small linear programming problem.

(i) First reduce the constraints associated with L^+ and L^- as much as possible. For every set of columns in L^+ which has only one entry different from zero, and for which all members of the set have this entry in the same row (say, r_1), eliminate all columns in the set except the one which would give the minimum if calculated by (5.1a). Similarly, for L^-, eliminate redundant constraints which operate only on one and the same δ_{r1}, by dropping only those for which (5.1b) is not a minimum in the set.

(ii) Solve the following linear programming problem

$$\text{Maximize} \sum_{\text{all } r} f_r\, \delta_r \qquad (5.2)$$

subject to

$$\sum_{\text{all } r} E_{r,ij}\, \delta_r = 0 \qquad (5.3a)$$

$$\sum_{\text{all } r} L^{+\,(k)}_{r,ij}\, \delta_r \leq \text{Min}\, (M_{ij} - \sum_{k\varepsilon L} X_{ij}^{(k)};\; M_{ij}^{(k)} - X_{ij}^{(k)}) \qquad (5.3b)$$

$$\sum_{\text{all } r} L^{-\,(k)}_{r,ij}\, \delta_r \geq (-X_{ij}^{(k)}) \qquad (5.3c)$$

$$\delta_r \geq 0 \qquad (5.3d)$$

the solution to this problem gives the solution to all δ_r, from the reduction procedure. If all δ_r are zero, go to Step (14). If not go to Step (13).

(13) Not all δ_r are zero in the optimal solution to (13) and (14); for each non-zero result, the flow in that chain or loop can be increased by δ_r (in the direction labelled), such that the net result of all such increases (as labelled) will be to keep the restricted primal satisfied, and increase the net flow into the network.

The change on the flow variables is made as follows: For ijk in the loop or chain r, increase $X_{ij}^{(k)}$ by δ_r if jk is labelled

$$\lambda_j^{(k)} = (ik \,//\, + \text{ or } Ex+)$$

or decrease $X_{ji}^{(k)}$ by δ_r if jk is labelled

$$\lambda_j^{(k)} = (ik \,//\, - \text{ or } Ex-)$$

Another way to think of the resulting change to be made in the flow variables is: ttake the original incidence table and the (column vector) of optimal δ_r, and form the dot product with each column of the original incidence table in L^+ and L^-, and add the result to the variable which has the same index ijk as the column.

The resulting increase in total flow into the network is the expression (4.2). There is no need to repeat the direct flow phase of this subroutine, since no new ijk ε $C \cap B$ have been defined. Proceed to Step (15).

(14) Either there are no ij ε $C \cap \overline{B}$, or no additional exchange labelling can be made, or no increase in flow via "exchange" is possible; set all $\delta_r = 0$.

The restricted primal has been solved, and the maximum possible flow in the restricted network is the same as for the last cycle.

It is now necessary to find the dual variables to the restricted primal.

Dual Variable Definition Phase

(15) One now defines the dual restricted variables $\sigma_i^{(k)}$, $\sigma_{ij}^{(k)}$, and σ_{ij}. In what follows, it is assumed that the flow does not achieve both its individual flow capacity restriction, and its mutual flow capacity restriction, simultaneously. This is equivalent to separating each branch for which this occurs into two series branches, and is done merely for simplicity in some of the rules presented below. Furthermore, it is assumed that parallel branches are similarly transformed.

First, examine the optimal solution of the linear program (5.2) and (5.3). If one found (via Step (14)) that all $\delta_r = 0$, one can still imagine that this solution was found the difficult way, by solving a linear programming problem.

In the program, equalities (5.3a) reduce (drastically) the allowed set of solutions; at optimality, it is one (or more) of the inequalities (5.3b) and (5.3c) which determines the actual bound on (13).

The solution to the dual of (5.2) and (5.3) is a set of β_{ij} such that

$$\sum_{ij\in B} E_{r,ij}\beta_{ij} \geq f_r - \beta_r \qquad (5.4)$$

and equality holds if $\delta_r > 0$; β_r equals zero unless this chain or loop had a branch which is saturated or emptied during the exchange flow, in which case it is non-negative. For example, for a chain from 0 to R with $\delta_r > 0$ which did not saturate or empty, (5.4) reads

$$\sum \pm \beta_{ij} = +1$$

where the sum is positive if ij is traversed in the forward direction, negative if traversed from j to i.

For a loop, the sum is equal to zero, etc.

Define a special subset of the nodes as follows, and denote them by the product notation IK:

Proceed with the labelling of steps (7)(8)(9) and (10), after optimality is reached, even though such labels do not "complete" chains or loops. The set of all labelled nodes is defined as IK (with some possible additions, described later). In other words:

(i) Nodes 0 and R and ik and jk $\ni$ ij $\in$ B are in IK.

(ii) If ik $\in$ IK, then jk $\in$ IK if

either (a) $(ijk \in C\cap\overline{B})\cap(X_{ij}^{(k)} < M_{ij}^{(k)})$

$$\cap\Big(\sum_{k\in L} X_{ij}^{(k)} < M_{ij}\Big)$$

or (b) $(jik \in C\cap\overline{B})\cap(X_{ij}^{(k)} > 0)$

or (c) $ijk \in C\cap B$

Define the dual variables by means of the following rules.

<u>Rule A</u>. Set $\sigma_0 = +1$, and $\sigma_R = 0$.

<u>Rule B.</u> If ijk ε IIK

and ijk ε $C \cap \overline{B}$, $\sigma_{ij} = \sigma_{ij}^{(k)} = 0$; $\sigma_i^{(k)} = \sigma_j^{(k)}$

and ijk $C \cap B$ $\sigma_{ij} = \beta_{ij}$; $\sigma_{ij}^{(k)} = 0$; $\sigma_i^{(k)} - \sigma_j^{(k)} = \sigma_{ij}$

That this set of rules can be applied consistently may be seen by examining all possible conflicting situations.

For ijk ε $C \cap \overline{B}$, the rules are consistent since no chain from 0 to R, or conflicting loop composed of only this type of branch is generated by the algorithm.

For "forward chains" joining 0 to R with non-zero changes in flow, (5.4) guarantees that

$$\sum \pm \sigma_{ij} = +1$$

and the rule is compatible since all $\sigma_i^{(k)}$ can be uniquely defined by following the chain from 0 to R. This procedure is also unique if several such chains have branches in common, since the incidence table took all such chains into account.

For "backward chains" joining R to 0, with non-zero δ_r, (5.4) guarantees that

$$\sum \pm \sigma_{ij} = -1$$

and the rule is compatible for the same reasons.

For loops in either direction around some mutually saturated branch, with $\delta_r > 0$, (5.4) guarantees that

$$\sum \pm \sigma_{ij} = 0$$

and the rule is compatible, with the $\sigma_i^{(k)}$ determined to within an additive constant; however, if the loop is part of a complete or incomplete chain of ijk ε IIK connecting any previously defined node variables, this redundancy will be removed, and the node variables are uniquely defined. This procedure works even for loops and chains which have branches in common, since the incidence table assures a compatible set for all chains and loops.

For chains and loops which were "broken" during the optimization, (5.4) states that

$$\sum \pm \sigma_{ij} \geq 0;\ +1,\ -1 \quad \text{(loop; forward, backward chain)}$$

and thus the rule can be satisfied for the remaining "pieces" of the chain or loop, with

$$\rho_{ij}^{(k)} = \sigma_i^{(k)} - \sigma_j^{(k)} - \sigma_{ij}^{(k)} - \sigma_{ij} \leq 0$$

for the branch which emptied, or $\rho_{ij}^{(k)} = 0$ and (say) $\sigma_{ij}^{(k)} \geq 0$ for the branch which saturated, etc. Any incomplete chains or loops may be considered as "broken", and the above argument applied.

For solutions with all $\delta_r = 0$, one may not have actually used the linear program, and dual variables σ_{ij} would not be defined. In this case, one must "do it yourself", and solve the following set of simultaneous equations:

$$\sum_{ij\varepsilon B}' E_{r,ij}\sigma_{ij} \geq f_r \qquad \text{all } r \qquad (5.5)$$

with equality holding for those "completed" chains and loops with non-zero flows.

This solution always exists, since one could always solve the linear programming problem (5.2) and (5.3). The rule continues to hold, and thus the dual variables can be uniquely computed for all nodes in IK, and all branches in ^{I}IK.

Rule C.

For

$$(ijk \ \varepsilon \ C \cap I\bar{I}K) \cap (\sum_{k\varepsilon L}' X_{ij}^{(k)} = M_{ij})$$

set

$$\sigma_{ij} = \max_{k\varepsilon L} (\sigma_i^{(k)}, 0) \quad \text{and} \quad \sigma_{ij}^{(k)} = 0$$

Furthermore, if $X_{ij}^{(k)} > 0$, set $\sigma_j^{(k)} = \sigma_i^{(k)} - \sigma_{ij}$, add the node jk to IK and continue with the labelling of new nodes in IK, defining new variables with Rule B. Otherwise, set $\sigma_j^{(k)} = 0$.

This rule defines a consistent set of σ_{ij} and $\sigma_j^{(k)}$ by "stopping" at the branch with greatest positive $\sigma_i^{(k)}$. and continuing the labelling on other branches ij if the flow is non-zero. If all the $\sigma_i^{(k)}$ are negative, the labelling continues on all branches with non-zero flow. This insures that $\rho_{ij}^{(k)} = 0$ for $X_{ij}^{(k)} > 0$, and $\rho_{ij}^{(k)} \leq 0$ otherwise.

Rule D. For

$$(ijk \ \varepsilon \ C \cap I\bar{I}K) \cap (X_{ij}^{(k)} = M_{ij}^{(k)}$$

set

$$\sigma_{ij}^{(k)} = \max (\sigma_i^{(k)}, 0) \qquad \sigma_{ij} = 0$$

and if $\sigma_i^{(k)}$ is negative, set $\sigma_j^{(k)} = \sigma_i^{(k)}$, add node jk to the set IK and continue with the labelling of new nodes in IK, defining new variables with Rule B. Otherwise, set $\sigma_j^{(k)} = 0$.

This rule insures that $\sigma_{ij}^{(k)} \geq 0$ and $\rho_{ij}^{(k)} = 0$ for this set. Note that there are no such branches in IIK.

Rule E. For ijk ε A, set

$$\sigma_{ij}^{(k)} = \sigma_i^{(k)} - \sigma_j^{(k)} \qquad \sigma_{ij} = 0$$

for all branches for which both node potentials were previously defined. If not, set undefined potentials equal to zero and then use this rule.

This rule makes $\rho_{ij}^{(k)} = 0$ for this set.

Rule F. For all other nodes and branches,

$$\sigma_i^{(k)} = \sigma_j^{(k)} = \sigma_{ij}^{(k)} = \sigma_{ij} = 0$$

Step (15) is completed.

- - - - - - - - - - - - - End of Maximal-Flow Subroutine - - - - - - - - - - - -

6. The Optimal Flow Algorithm for Multi-Commodity Networks

The algorithm to be described has the following program:

Begin → 1. Select an initial feasible solution to the dual problem.

2. In terms of the feasible dual, define a restricted primal problem, and a set of the primal variables which can be changed.

3. Solve the restricted primal problem by maximizing flow into the restricted primal network, in two steps:
   (a) Maximize flow which can be sent directly through each layer.
   (b) Maximize flow which can be sent by "exchanging" mutual capacity between layers.

Optimal Solution ← 4. If the flow input to the network is equal to Q, the optimal solution is obtained.

5. Otherwise, use the solution to (3) to define changes in the dual variables which form a new feasible solution to the dual problem; repeat Step (2), until

Problem Infeasible ← 6. No such changes can be made, and the flow input is not Q, and the problem is infeasible.

- -

Step (1)

To begin with, select a dual feasible solution, such as the least-cost route from source to sink, or, set all dual variables equal to zero.

Step (2)

From Step (1), or from the output of Step (5), define the following sets:

$$A = (ijk \ \varepsilon \ NNL \ | \ U_{ij}^{(k)} > 0) \tag{6.1a}$$

$$B = (ij \ \varepsilon \ NN \ | \ U_{ij} > 0 \ ) \tag{6.1b}$$

$$C = (ijk \ \varepsilon \ NNL \ | \ (V_i^{(k)} - V_j^{(k)} - U_{ij}^{(k)} - U_{ij} = c_{ij}^{(k)}) \bigcap (U_{ij}^{(k)} = 0) \bigcap (U_{ij} \geq 0) \ ) \tag{6.1c}$$

and

$$D = A \bigcup B \bigcup C \tag{6.1d}$$

Define the following *restricted primal problem*:

$$\text{Maximize } F_0 \tag{6.2}$$

subject to constraints:

$$\sum_{j \varepsilon N} (X_{ij}^{(k)} - X_{ji}^{(k)}) = 0 \qquad ik \ \varepsilon \ NL \tag{6.3a}$$

$$\sum_{k \varepsilon L} (X_{0j}^{(k)} - X_{j0}^{(k)}) = F_0 \tag{6.3b}$$

$$\sum_{k \varepsilon L} (X_{Rj}^{(k)} - X_{jR}^{(k)}) \geq -Q \tag{6.3c}$$

$$X_{ij}^{(k)} \leq M_{ij}^{(k)} \qquad ijk \ \varepsilon \ NNL \tag{6.3d}$$

$$\sum_{k \varepsilon L} X_{ij}^{(k)} \leq M_{ij} \qquad ij \ \varepsilon \ NN \tag{6.3e}$$

$$X_{ij}^{(k)} \geq 0 \qquad ijk \ \varepsilon \ NNL \tag{6.3f}$$

$$X_{ij}^{(k)} = 0 \qquad ijk \ \varepsilon \ \overline{D} \tag{6.3g}$$

$$X_{ij}^{(k)} = M_{ij}^{(k)} \qquad ijk \ \varepsilon \ A \tag{6.3h}$$

$$\sum_{k \varepsilon L} X_{ij}^{(k)} = M_{ij} \qquad ij \ \varepsilon \ B \tag{6.3i}$$

The dual to (6.2) and (6.3), called the *dual restricted problem*, is:

$$\text{Minimize} \sum_{ijk \varepsilon NNL} M_{ij}^{(k)} \sigma_{ij}^{(k)} + \sum_{ij \varepsilon NN} M_{ij} \sigma_{ij} \tag{6.4}$$

subject to constraints:

$$\sigma_i^{(k)} - \sigma_j^{(k)} - \sigma_{ij}^{(k)} - \sigma_{ij} \leq 0 \qquad ijk \; \varepsilon \; D \qquad (6.5a)$$

$$\sigma_0 = +1 \qquad (6.5b)$$

$$\sigma_{i \neq 0}^{(k)} \text{ unrestricted} \qquad (6.5c)$$

$$\sigma_{ij}^{(k)} \geq 0 \qquad ijk \; \varepsilon \; D \cap \overline{A} \qquad (6.5d)$$

$$\sigma_{ij} \geq 0 \qquad ij \; \varepsilon \; D \cap \overline{B} \qquad (6.5e)$$

Step (3)

Solve (6.2) and (6.3) by using the maximal flow subroutine described in Section 5. The result is a set of flows, satisfying the restricted primal and a set of dual restricted variables, satisfying (6.4) and (6.5).

Step (4)

If $F_0 = Q$, the algorithm is terminated, with the set of $X_{ij}^{(k)}$ just defined as the optimal solution to (2.1) and (2.2), and the dual feasible defined at the beginning of this cycle being the optimal solution to (2.3) and (2.4).

If $F_0 < Q$, go to Step (5).

Step (5)

In terms of the solution to the dual restricted problem in Step (3), define a new set of dual feasible variables by:

$$V_i^{(k)\prime} = V_i^{(k)} + \vartheta \sigma_i^{(k)} \qquad ik \; \varepsilon \; NL \qquad (6.6a)$$

$$U_{ij}^{(k)\prime} = U_{ij}^{(k)} + \vartheta \sigma_{ij}^{(k)} \qquad ijk \; \varepsilon \; NNL \qquad (6.6b)$$

$$U_{ij}^{\prime} = U_{ij} + \vartheta \sigma_{ij} \qquad ij \; \varepsilon \; NN \qquad (6.6c)$$

with

$$\vartheta = \min \left\{ \begin{array}{l} \min\limits_{ijk\varepsilon\overline{D}} \left(\dfrac{c_{ij}^{(k)} - (V_i^{(k)} - V_j^{(k)} - U_{ij}^{(k)} - U_{ij})}{\sigma_i^{(k)} - \sigma_j^{(k)} - \sigma_{ij}^{(k)} - \sigma_{ij}} \right) ; \min\limits_{ijk\varepsilon A} \left(\dfrac{U_{ij}^{(k)}}{-\sigma_{ij}^{(k)}} \right), \\ \min\limits_{ij\varepsilon B} \left(\dfrac{U_{ij}}{-\sigma_{ij}} \right) \end{array} \right\} \qquad (6.6d)$$

for all ijk such that the denominators are positive. If none of the denominators are positive, $\vartheta = +\infty$.

Repeat Step (2) with the new dual variables defined by this step if ϑ is finite.

Step (6)

If $\Theta = +\infty$, terminate the algorithm, since no feasible solution to (2.1) and (2.2) exists, and the dual set (2.3) and (2.4) is unbounded.

By comparison with Chapters II and III, it is seen that the optimal flow algorithm is quite similar to those developed for simple networks, and networks with gain. The principle difference is in the maximal flow subroutine. The proof may be found in Appendix D.

7. Alternate Formulations

It should be apparent that nothing in the formulation of this algorithm (except notation) depends upon the "layer" concept of separating components which are mutually constrained. For this reason, it is possible to solve problems in which the coupling involves any arbitrary subset of branches in a network. For example, in a "bottleneck operation" in a large distribution system, the sum of flows in a large group of branches may represent the total flow through the bottleneck -- especially if it is not convenient to merge the flows to restrict this total.

8. Communication Message Routing

As an interesting application of multi-commodity networks, consider the following interoffice trunking problem, due to R.E. Kalaba and M.L. Juncosa (51)(52):

Let $\underline{a_{ij}}$ denote the known number of trunks needed between station $\underline{i}$ and station $\underline{j}$, in order to handle all communications between the two stations. Furthermore, c_{ij} is the known capacity (in number of channels) available between stations $\underline{i}$ and $\underline{j}$. How should one make up the trunks out of the existing channels in order to:

(a) maximize the number of completed interoffice trunks?

(b) find that solution, among those generated by (a), which additionally uses the minimal number of channels?

Because the channels are assumed bi-directional, one must introduce branches (i,j) and (j,i) between stations, and constrain the sum of their flows to be less than c_{ij}. Then, one "separates" the network into a layer for each receiving station; the trunking demands, a_{ij}, are introduced as a capacity on the input to the _i_th station in the _j_th layer, and the output from the _j_th layer comes entirely from the _j_th station, and is limited to $\sum_i a_{ij}$. The mutual capacity constraints involve two branches from each layer.

With this multi-commodity synthesis of the communication network, problem (a) is just a maximal-flow problem, which may be solved by the subroutine of Section 5. Problem (b) is a minimal-cost problem in which the per-unit cost of each branch in the original network is unity.

Appendix C. Proof of the Optimal Flow Algorithm for Multi-Commodity Networks

Proof of the Maximal Flow Subroutine

The essential part of the algorithm is the maximal-flow subroutine which solves (IV.6.2)(IV.6.3) and (IV.6.4)(IV.6.5). We shall prove that this procedure provides an optimal solution by showing that the constraints to both the restricted primal and its dual are satisfied, and that the principle of complementary slackness holds. This, incidentally, will also show the equality of the functionals, and provide a new "max-flow, min-cut" theorem for multi-commodity networks.

The direct flow phase of the subroutine presents nothing new or unusual. Some remarks on the exchange flow phase are in order, however.

The incidence table introduced in Step (6) of the subroutine merely describes the effect of an increase in flow (in a certain direction along a chain or loop of branches in a certain subset of the network) on:

(i) the mutual capacity restrictions (E),
(ii) increase of flow into the network (f),
(iii) saturating (L^+) or emptying (L^-) a branch in the chain or loop.

Thus, before reduction, all elements in the incidence table are zero, or plus or minus one.

The reason for the choice of the subset described in Step (7b), is that equations (IV.6.3g) and (IV.6.3h) prohibit any changes in branches in A and $\overline{D}$. By eliminating all direct chains of branches in $C \cap \overline{B}$ from 0 to R in the direct flow phase, all chains or loops described in the incidence table will contain at least one mutually saturated branch. Direct chains of branches in $C \cap \overline{B}$ from R to 0 without any mutually saturated branches are without value, and never need be examined.

The reduction procedure, Step (11) of the subroutine, is superfluous, since one could let the linear program do all the work. However, from a computational standpoint, it is worthwhile to eliminate these possibilities which could never give an increase in flow.

For example, Step (11a) states that a certain set of mutually saturated branches are not affected by any proposed changes in flow, and may therefore be eliminated.

Step (11b) states that is, for some ij ε B, all proposed increases in flow will tend to over- or under-shoot the mutual restriction, then no compensating arrangement can be made to keep the restriction exactly satisfied, and one can set the proposed changes in flow in these chains and loops equal to zero. (Recall that all proposed changes in flow are non-negative ($\delta_r \geq 0$); a decrease in flow would be expressed by a labelling in the reverse direction, and a separate entry in the incidence table.)

Steps (11c) and (11d) state that if one has an equality:

$$e\delta_{r1} - ne\delta_{r2} = 0$$

or

$$-e\delta_{r_1} + ne\delta_{r_2} = 0$$

then $\delta_{r_1} = n\delta_{r_2}$ and one variable may be eliminated in terms of the other.

The condition that there be at least one non-negative entry under F requires that there be at least one way of increasing flow into the network with the proposed changes. The reduction procedure on the constraints merely eliminates redundant ones.

More complicated reduction and elimination procedures could probably be made, but are not worth the trouble, in the author's opinion. The procedures presented can be easily programmed, and will detect most simple exchange situations; if a general linear program is still left after reduction, chances are that the optimal exchange is a subtle one, indeed.

Lemma I. If $ijk \in D \cap \bar{A} = B \cup C$, then $\sigma_{ij}^{(k)} \geq 0$.

For $ij \in B$, $\sigma_{ij}^{(k)} = 0$, by Rule B.

For $ijk \in C$ and $X_{ij}^{(k)} < M_{ij}^{(k)}$, $\sigma_{ij}^{(k)} = 0$, Rules B and F.

and $X_{ij}^{(k)} = M_{ij}^{(k)}$, $\sigma_{ij}^{(k)} \geq 0$, Rule D.

Lemma II. If $X_{ij}^{(k)} < M_{ij}^{(k)}$, then $\sigma_{ij}^{(k)} = 0$.

No $\sigma_{ij}^{(k)}$ is non-zero, unless the flow is individually saturated.

Lemma III. If $ij \in \bar{B}$, then $\sigma_{ij} \geq 0$

The only non-zero σ_{ij} in this set is defined in Rule C as non-negative.

Lemma IV. If $\sum_{k \varepsilon L} X_{ij}^{(k)} < M_{ij}$, then $\sigma_{ij} = 0$.

No σ_{ij} is non-zero, unless the flow is mutually saturated.

Lemma V. If $ijk \in D$, then $\rho_{ij}^{(k)} = \sigma_i^{(k)} - \sigma_j^{(k)} - \sigma_{ij}^{(k)} - \sigma_{ij} \leq 0$

For $ijk \in A$	$\rho_{ij}^{(k)} = 0$	Rule E
For $ij \in B$	$\rho_{ij}^{(k)} = 0$	Rule B

For ijk ε C $\quad X_{ij}^{(k)}$ non-zero, non-saturated (either kind)

$\rho_{ij}^{(k)} = 0$ Rule B

$X_{ij}^{(k)}$ saturated (either kind)

$\rho_{ij}^{(k)} = 0$ Rule C or D

$X_{ij}^{(k)}$ zero, $\sigma_i^{(k)} - \sigma_j^{(k)} \leq 0$; otherwise, a different labelling would have been made.

Lemma VI. If $X_{ij}^{(k)} > 0$, then $\rho_{ij}^{(k)} = 0$.

In Lemma V, the only cases for which this is not true are for zero flow, and hence if ijk ε D, the Lemma is proved. But by (IV.6.3g), flow must be zero if ijk ε $\overline{D}$.

Lemma VII. The subroutine solves the restricted primal problem and its dual.

The primal constraints are always satisfied, since it is assumed that they are satisfied at this cycle in the algorithm, and a Lemma in the next section will show that it is feasible in the next cycle.

Dual constraints (10a) are satisfied, by Lemmas I, III, and V.

Also, the principle of complementary slackness holds, by Lemmas II, IV, and VI. Therefore, by the fundamental theorem of linear programming, an optimal solution to both the restricted primal problem and its dual have been found, and furthermore........

Lemma VIII. $F_0 = \sum_{ijk} M_{ij}^{(k)} \sigma_{ij}^{(k)} + \sum_{ij} M_{ij} \sigma_{ij}$

By the fundamental theorem, the two functionals are equal when the optimal solution is reached.

This statement is the equivalent of the "max-flow, min-cut" theorem for simple networks. Note that the cut must intercept all of the individually saturated branches, and one of the mutually saturated branches for each ij.

Proof of the Algorithm

Given that the subroutine works correctly, we must now prove that the algorithm solves (IV.2.1)(IV.2.2) and (IV.2.3)(IV.2.4).

In the following Lemmas, unprimed symbols refer to a given cycle of the algorithm, primed symbols refer to the succeeding one. It is assumed that the restricted primal constraints are feasible on this cycle, and we will show that it is feasible on the next cycle, showing (by induction) that it is feasible always.

Since we show that the dual constraints are always satisfied, and that the dual functional increases for every cycle, the optimal solution is reached in a finite number of cycles of the algorithm.

Lemma IX. If $V_i^{(k)}$, $U_{ij}^{(k)}$ and U_{ij} satisfy the dual (4), so do $V_i^{(k)'}$, $U_{ij}^{(k)'}$, U_{ij}'.

The new dual variables are given by (IV.6.6).

(i) Dual inequality (IV.2.4a) states that $\underbrace{V_i^{(k)} - V_i^{(k)} - U_{ij}^{(k)} - U_{ij}}_{= \text{L.H.S.}} \leq C_{ij}^{(k)}$

By (IV.5.6),

$$(\text{L.H.S.})' = (\text{L.H.S.}) + \vartheta \rho_{ij}^{(k)}$$

If equality holds, ijk ε D, and $\rho_{ij}^{(k)} \leq 0$ (Lemma V), so that the inequality holds in the new cycle for any non-negative θ. If (IV.2.4a) is a strict inequality, the inequality will hold in the new cycle for

$$0 < \theta \leq \vartheta_1$$

where

$$\vartheta_1 = \min_{\rho_{ij}^{(k)} > 0} \left\{ \frac{C_{ij}^{(k)} - (\text{L.H.S.})}{\rho_{ij}^{(k)}} \right\}$$

or $= +\infty$, if all $\rho_{ij}^{(k)} \leq 0$.

(ii) Dual inequality (IV.2.4b) states that $U_{ij}^{(k)} \geq 0$.

If $U_{ij}^{(k)} = 0$, then ijk ε $C \cup B \cup \bar{D}$. For ijk ε $C \cup B$, $\sigma_{ij}^{(k)} \geq 0$ (Lemma I).

For ijk ε $\bar{D}$, $\sigma_{ij}^{(k)} = 0$, and no change occurs.

Hence the inequality holds in the new cycle for any non-negative θ.

If $U_{ij}^{(k)} > 0$, then (IV.2.4b) will hold in the new cycle for

$$0 < \theta \leq \theta_2$$

where

$$\vartheta_2 = \min_{\sigma_{ij}^{(k)} < 0} \left\{ \frac{U_{ij}^{(k)}}{-\sigma_{ij}^{(k)}} \right\}$$

or $+\infty$ if all $\sigma_{ij}^{(k)} \geq 0$.

(iii) Dual inequality (IV.2.4c) states $U_{ij} \geq 0$

If $U_{ij} = 0$, then ijk ε $C \cup A \cup \bar{D}$. For ijk ε $C \cup A$, $\sigma_{ij} \geq 0$ (Lemma III).

For ijk ε $\bar{D}$, $\sigma_{ij} = 0$, and no change occurs.

Hence the inequality holds in the new cycle for any nonnegative ϑ.

If $U_{ij} > 0$, then (IV.2.4c) will hold in the new cycle for

$$0 < \vartheta \leq \vartheta_2$$

where

$$\vartheta_2 = \min_{\sigma_{ij} \leq 0} \left\{ \frac{U_{ij}}{-\sigma_{ij}} \right\}$$

or $+\infty$ if all $\sigma_{ij} \geq 0$.

Since the algorithm picks $\vartheta = \min(\vartheta_1, \vartheta_2, \vartheta_3)$ in Step (11d), the dual remains satisfied in every cycle.

Lemma X. If ijk ε A', $X_{ij}^{(k)} = M_{ij}^{(k)}$

If ijk was in A, the subroutine leaves $X_{ij}^{(k)}$ unchanged. Otherwise, it must have been that $U_{ij}^{(k)}$ was zero, and $\sigma_{ij}^{(k)} > 0$, which occurs for individually saturated branches.

Lemma XI. If ij ε B', $\sum_k X_{ij}^{(k)} = M_{ij}$

If ij was in B, the subroutine leaves the sum unchanged. Otherwise, it must have been that U_{ij} was zero, and $\sigma_{ij} > 0$, which occurs only for mutually saturated branches.

Lemma XII. If ijk ε $\bar{D}'$, $X_{ij}^{(k)} = 0$

If ijk was in $\bar{D}$ the subroutine leaves $X_{ij}^{(k)}$ unchanged. Otherwise, it must have been that (IV.2.4a) was an equality, and $\rho_{ij}^{(k)} < 0$, which occurs only for zero flow.

Lemma XIII. The constraint set (IV.6.3) is feasible for the new cycle, and the previous $X_{ij}^{(k)}$ may be used as an initial solution in the new restricted primal.

Constraints (IV.6.3a)(IV.6.3b)(IV.6.3c)(IV.6.3d)(IV.6.3e) and (IV.6.3f) are always satisfied in the algorithm. Lemmas X, XI, and XII show that (IV.6.3g) (IV.6.3h) and (IV.6.3i) are feasible for the new cycle, and the subroutine does not violate this feasibility.

Lemma XIV. An optimal solution to the restricted primal with a maximal flow F_0 provides a new feasible solution to the dual problem, with a strict increase in the dual functional equal to $\vartheta(Q - F_0)$.

The difference between the new and the old functional is:

$$\Theta(Q\sigma_0 - \sum_{ijk} M_{ij}^{(k)} \sigma_{ij}^{(k)} - \sum_{ij} M_{ij} \sigma_{ij})$$

which by Lemma VIII is equal to

$$\Theta(Q - F_0)$$

which is strictly positive.

Lemma XV. When $\Theta = +\infty$, the problem is infeasible; when $Q - F_0$ is zero, the algorithm terminates with the optimal solution.

The first result follows from the duality theorem, since we have an unbounded increase in the dual functional.

The second result follows from the fact that the primal and dual restrictions have been satisfied simultaneously, and the two functionals are equal for,

$$\text{If (L.H.S.)} < c_{ij}^{(k)} \quad , \quad x_{ij}^{(k)} = 0$$

and by some algebraic manipulation, one can show that the functionals are indeed, equal.

Optimality follows from the fundamental theorem of linear programming.

Bibliography*

1. Antosiewicz, H.A., ed., Proceedings of the Second Symposium in Linear Programming, Vols. 1 and 2, Directorate of Management Analysis, DCS/Comptroller, Headquarters U.S. Air Force, Washington, D.C.

2. Bartlett, T.E., "An Algorithm for the Minimum Number of Transport Units to Maintain a Fixed Schedule," Naval Res. Logist. Quart., 4, No. 2 (1957), 139-150.

3. Beale, E.M.L., "An Alternative Method for Linear Programming," Proc. Cambridge Philos. Soc. 50 (1954), 513-523.

4. Bellman, E., "Notes on the Theory of Dynamic Programming - VII: Transportation Models," The RAND Corporation, Paper 771, 28 Nov. 1955.

5. Bishop, G.T., "On a Problem of Production Scheduling," Jour. Oper. Res. Soc. Amer. 5, no. 1 (1957), 97-103.

6. Boldyreff, A., "Determination of the Maximal Steady State Flow of Traffic Through a Railroad Network," Jour. Oper. Res. Soc. Amer. 3, (1955), 443-465.

7. Bowman, E,H., "Production Scheduling by the Transportation Method of Linear Programming," Jour. Oper. Res. Soc. Amer. 4, no. 1 (1956), 100-103.

8. Charnes, A., and Cooper, W.W., "Generalization of the Warehousing Model," Operations Res. Quart., 6 (1955), 131-172.

9. Charnes, A., Cooper, W.W., and Farr, D. and Staff, "Linear Programming and Profit Preference Scheduling for a Manufacturing Firm," Jour. Oper. Res. Soc. Amer., 1, no. 3 (1953), 114-129.

10. Charnes, A., Cooper, W.W., and Henderson, A., An Introduction to Linear Programming, Wiley, New York, 1953.

11. Charnes, A., and Miller, M.H., "A Model for the Optimal Programming of Railway Freight Train Movements," Management Science, 3, no. 1 (1956), 74-92.

12. Dantzig, G.B., "Maximizing of a Linear Function of Variables Subject to Linear Inequalities," Chapter XXI of Activity Analysis of Production and Allocation, edited by Koopmans, T.C., Wiley, N.Y., 1951.

13. Dantzig, G.B., "Application of the Simplex Method to a Transportation Problem," Chapter XXII of Activity Analysis of Production and Allocation, edited by Koopmans, T.C., Wiley, N.Y., 1951.

* For a more complete listing of general references in linear programming and its applications, the reader is referred to the excellent listings in (61), (73), (76), and (83).

14. Dantzig, G.B., "Computational Algorithm of the Simplex Method", The RAND Corporation, Paper 394, 10 April 1953. (RM-1266, Part III, 26 October, 1953).

15. Dantzig, G.B., "Constructive Proof of the Min-Max Theorem," RM-1267, Part IV, 8 Sept. 1954. (Pacif. J. Math., 6, no. 1, (1956), 25-33).

16. Dantzig, G.B., "Upper Bounds, Secondary Constraints, and Block Triangularity in Linear Programming," The RAND Corporation, RM-1367, 4 Oct. 1954. (Econometrica 23, no. 2 (1955), 174-183).

17. Dantzig, G.B., "Linear Programming Under Uncertainty," Management Science, 1, no. 3-4 (1955), 197-206.

18. Dantzig, G.B., "Recent Advances in Linear Programming," The RAND Corporation, RM-1475, Part XXII, 12 April 1955. (Management Sci., 2, no. 2 (1956), 131-144.)

19. Dantzig, G.B., and Ferguson, A.R., "The Allocation of Aircraft to Routes--An Example of Linear Programming Under Uncertain Demand," The RAND Corporation, RM-1833 Part XXXVI, 7 Dec. 1956. (Management Sci., 3, no. 1 (1956), 45-73).

20. Dantzig, G.B., Ford, L.R., and Fulkerson, D.R., "A Primal-Dual Algorithm for Linear Programs," The RAND Corporation, RM-1709, Part XXXl, 9 May 1956, (Paper 7 in Linear Inequalities and Related Systems, Princeton Univ. Press, Princeton, 1956), 171-181.

21. Dantzig, G.B., and Fulkerson, D.R., "Minimizing the Number of Tankers to Meet a Fixed Schedule," Naval Res. Log. Quart., 1, no. 3 (1954), 217-222.

22. Dantzig, G.B., and Fulkerson, D.R., "On the Min-Cut Max-Flow Theorem of Networks," The RAND Corporation, RM-1418, Part XXl, 15 April 1955, (Paper 12 in Linear Inequalities and Related Systems, Princeton Univ. Press, Princeton 1956), 215-222.

23. Dantzig, G.B., Manne, A., Orchard-Hays, W., and Robacker, J., "An Algorithm for the Solution of Certain Block Triangular Linear Programming Systems," the RAND Corporation, D-3514, 28 Feb. 1956.

24. Dantzig, G.B., Orden, A., and Wolfe, P., "The Generalized Simplex Method for Minimizing a Linear Form Under Linear Inequality Restraints", The RAND Corporation, RM-1264, 5 April 1954. (Pacif. J. Math., 5, no. 2 (1955), 183-195.)

25. Dreyfus, S.E., "An Analytic Solution of the Warehouse Problem," Management Sci., 4, no. 1 (1957), 99-104.

26. Flood, M.M., "Application of Transportation Theory to Scheduling a Military Tanker Fleet," Jour. Oper. Res. Soc. Amer., 2, no. 2 (1954), 150-162.

27. Ford, L.R., "Network Flow Theory", The RAND Corporation, Paper 923, 14 Aug. 1956.

28. Ford, L.R., and Fulkerson, D.R., "Maximal Flow Through a Network," The RAND Corporation, RM-1400, Part XX, 19 Nov. 1954. (Can. J. Math., 8, no. 3 (1956), 399-404.)

29. Ford, L.R., and Fulkerson, D.R., "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem," The RAND Corporation, Paper 743, 26 Sept. 1955; RM-1604, 29 Dec. 1956.

30. Ford, L.R., and Fulkerson, D.R., "Solving the Transportation Problem," The RAND Corporation, Paper 895, 20 June 1956. (Management Sci., 3, no. 1 (1956), 24-32.)

31. Ford, L.R., and Fulkerson, D.R., "A Primal Dual Algorithm for the Capacitated Hitchcock Problem," The RAND Corporation, RM-1798 Part XXXIV, 25 Sept. 1956. (Naval Res. Logist. Quart., 4, no. 1 (1957), 47-54, RAND Conference on Computational Aspects of Linear Programming, 30-31 Aug. 1956.)

32. Ford, L.R., and Fulkerson, D.R., "Constructing Maximal Dynamic Flow from Static Flows," The RAND Corporation, RM-1981, Part XLI, 17 Sept. 1957.

33. Friedman, L. and Yaspan, A.J., "The Assignment Problem Technique," Annex A: "An Analysis of Stewardess Requirements and Scheduling," by J.A. McCloskey and F. Hanssmann, Naval Res. Logist. Quart., 4, no. 3 (1957), 193-198. Purdue Univ. First Transportation Research Symposium, 27-28 Feb. 1957.

34. Fulkerson, D.R., and Dantzig, G.B., "Computation of Maximum Flows in Networks," The RAND Corporation, RM-1489, Paper XXVI, 1 April 1956. (Naval Res. Logist. Quart., 2, no. 4 (1955), 277-283.)

35. Gaddum, J.W., Hoffman, A.J., and Sokolowsky, D., "On the Solution of the Caterer Problem," Naval Res. Logist. Quart., 1, no. 3 (1954), 223-229.

36. Gale, D., "The Basic Theorems of Real Linear Equations, Inequalities, Linear Programming, and Game Theory," Naval Res. Logist. Quart., 3, no. 3 (1956), 193-200.

37. Gale, D., Kuhn, H., and Tucker, A., "Linear Programming and the Theory of Games," Chapter XIX of Activity Analysis of Production and Allocation Wiley, New York, 1951.

38. Garvin, W.W., Crandall, H.W., John, J.B., and Spellman, R.A., "Applications of Linear Programming in the Oil Industry," Management Sci., 3, no. 4 (1957), 407-430.

39. Gass, S.I., "On the Distribution of Manhours to Meet Scheduled Requirements," Naval Res. Logis. Quart., 4, no. 1 (1957), 17-26. RAND Conference on Computational Aspects of Linear Programming, 30-31 Aug. 1956.

40. Goldman, A.J., and Tucker, A.W., "Theory of Linear Programming," Paper 4 in Linear Inequalities and Related Systems, Princeton Univ. Press, Princeton, 1956, 53-98.

41. Heller, I. and Tompkins, C.B., "An Extension of a Theorem of Dantzig," Paper 14 in Linear Inequalities and Related Systems, Princeton Univ. Press, Princeton, 1956, 247-254.

42. Hitchcock, F.L., "The Distribution of a Product from Several Sources to Numerous Localities," J. Math. Physics (M.I.T.), 20 (1941), 224-230.

43. Houthakker, H.S., "On the Numerical Solution of the Transportation Problem," J. Oper. Res. Soc. Amer., 3, no. 2 (1955), 210-214.

44. Jacobs, W., "The Caterer Problem," Naval Res. Logist. Quart., 1 (1954), 154-165.

45. Jewell, W.S., "Multi-Stage Allocation of Limited Resources," Proposal for a Thesis, M.I.T. Department of Electrical Engineering, June 1956.

46. Jewell, W.S., Progress Reports nos. 1-12 (unpublished), 1956-1958, M.I.T., Department of Electrical Engineering and Operations Research Project.

47. Jewell, W.S., "Warehousing and Distribution of a Seasonal Product," Naval Res. Logist. Quart., 4, no. 1 (1957), 29-34. RAND Conference on Computational Aspects of Linear Programming, 30-31 Aug. 1956.

48. Jewell, W.S., "Multi-Stage Allocation and Limited Warehousing Problems," Talk presented at the Fifth Annual Meeting of O.R.S.A., 9-10 May 1957, Philadelphia, Pa.

49. Johnson, S.M., "Sequential Production Planning Over Time At Minimum Cost," Management Sci., 3, no. 4 (1957), 435-437.

50. Johnson, S.M., and Dantzig, G.B., "A Production Smoothing Problem," in Proceedings of the Second Symposium in Linear Programming, Vol. 1 and 2, Directorate of Management Analysis, DCS/Comptroller, Headquarters, U.S. Air Force, Washington, D.C., 151-176.

51. Kalaba, R.E., and Juncosa, M.L., "Optimal Design and Utilization of Communication Networks," The RAND Corporation, RM-1687, 23 April 1956. (Management Sci., 3, no. 1 (1956), 33-44.)

52. Kalaba, R.E., and Juncosa, M.L., "Interoffice Trunking Problems," The RAND Corporation, RM-1688, 15 Nov. 1956.

53. Kantorovitch, L., "On the Translocation of Masses," Dokl. Akad. Nauk. S.S.R., 37 (1942), 199-201.

54. Karush, W., and Vazsonyi, A., "Mathematical Programming and Service Scheduling," Management Sci., 3, no. 2 (1957), 140-148.

55. Kelley, J.E. Jr., "A Dynamic Transportation Model," Naval Res. Logist. Quart., 2, no. 3 (1955), 175-180.

56. König, D., Theorie Der Endlichen Und Unendlichen Graphen, Akademische Verlagsgesellschaft M.B.H., Leipzig, 1936. (Chelsea Publishing Co., New York, 1950.)

57. Koopmans, T.C., ed., Activity Analysis of Production and Allocation, Wiley, New York, 1951.

58. Koopmans, T.C., and Reiter, S., "A Model of Transportation," Chapter XIV of Activity Analysis of Production and Allocation, Wiley, N.Y., 1951.

59. Kuhn, H.W., "The Hungarian Method for the Assignment Problem," Naval Res. Logist. Quart., 2, nos. 1 and 2 (1955), 83-97.

60. Kuhn, H.W., "Variants of the Hungarian Method for Assignment Problems," Naval Res. Logist. Quart., 3, no. 4 (1956), 253-258.

61. Kuhn, H.W. and Tucker, A.W., ed., Linear Inequalities and Related Systems, Princeton University Press, Princeton, 1956.

62. Lemke, C.E., "The Dual Method of Solving the Linear Programming Problem," Naval Res. Logist. Quart., 1 (1954), 36-47.

63. Manne, A.S., "A Note on the Modigliani-Hohn Production Smoothing Model," Management Sci., 3, no. 4 (1957), 371-379.

64. Markowitz, H., "Concepts and Computing Procedures for Certain X_{ij} Programming Problems," in the Proceedings of the Second Symposium in Linear Programming, Vols. 1 and 2, Directorate of Management Analysis, DCS/Comptroller, Headquarters, U.S. Air Force, Washington, D.C., 509-565. (The RAND Corporation, Paper 602, 19 Nov. 1954.)

65. Mitten, L.G., "An Algorithm for a Restricted Class of Linear Programming Problems," Talk presented at the Fifth Annual Meeting of O.R.S.A., 9-10 May 1957, Philadelphia, Pa.

66. Motzkin, T.S., "The Multi-Index Transportation Problem," Bull. of the Amer. Math. Soc. 58, no. 4 (1952), 494 (abstract).

67. Motzkin, T.S., "The Assignment Problem," Proc. Sixth Symp. in Applied Math. (Amer. Math. Soc.), New York, 1956.

68. Orden, A. and Goldstein, L., eds., Symposium on Linear Inequalities and Programming (Project SCOOP), Directorate of Management Analysis, DCS/Comptroller, Headquarters U.S. Air Force, Washington, D.C., 1952.

69. Orden, A., "The Transshipment Problem," Management Sci., 2, no. 3 (1956), 276-285.

70. Prager, W., "On the Caterer Problem," Management Sci., 3, no. 1 (1956), 15-23.

71. Prager, W., "Numerical Solution of the Generalized Transportation Problem," Naval Res. Logist. Quart., 4, no. 3 (1957), 253-261.

72. Prager, W., "On Warehousing Problems," J. Oper. Res. Soc. Amer., 5, no. 4 (1957), 504-512.

73. Riley, V., "Programming for Policy Decision", Mathematical Techniques, 1, Bibliographific Reference Series no. 2, Operations Research Office, Chevy Chase, Md., 1 March 1954.

74. Robacker, J.T., "On Network Theory," The RAND Corporation, RM-1498, 26 May 1955.

75. Robacker, J.T., "Concerning Multicommodity Networks," The RAND Corporation, RM-1799, Part XXXVII, 26 Sept. 1956.

76. Rohde, F.V., "Bibliography on Linear Programming," J. Oper. Res. Soc. Amer., 5, no. 1 (1957), 45-62.

77. Saaty, T.L. and Gass, S.I., "The Parametric Objective Function," J. Oper. Res. Soc. Amer., 2, no. 3 (1954), 316-319, Part I; 3, no. 4 (1955), 395-405, Part II.

78. Salverson, M.E., "A Problem in Optimal Machine Loading," Management Sci., 2, no. 3 (1956), 232-260.

79. Schell, E.D., "Distribution of a Product by Several Properties," in Proceedings of the Second Symposium in Linear Programming, Vols. 1 and 2, Directorate of Management Analysis, DCS/Comptroller, Headquarters, U.S. Air Force, Washington, D.C., 615-642.

80. Tucker, A.W. "Dual Systems of Homogenous Linear Relations," Paper I in Linear Inequalities and Related Systems, Princeton Univ. Press, Princeton, 1956.

81. Vajda, S., The Theory of Games and Linear Programming, John Wiley and Sons, New York, 1956.

82. Votaw, D.F. and Orden, A., "The Personal Assignment Problem," in the Symposium on Linear Inequalities and Programming, (Project SCOOP), Directorate of Management Analysis, DCS/Comptroller, Headquarters, U.S. Air Force, Washington, D.C., (1952), 155-163.

83. Wagner, H.M., "A Supplementary Bibliography on Linear Programming," J. Oper. Res. Soc. Amer., 5, no. 4 (1957), 555-563.

Added in proof:

84. Ford, L.R. Jr., and Fulkerson, D.R., "A Suggested Computation for Maximal Multi-Commodity Network Flows," The RAND Corporation, D-1114, 27 March 1958.

Additional Bibliography

A. Bradley, S., "Solution Techniques for the Traffic Assignment Problem," ORC 65-35, Operations Research Center, University of California, Berkeley, (June, 1965).

B. Dantzig, G. B., and P. Wolfe, "The Decomposition Algorithm for Linear Programs," Econometrica, Vol. 29, pp. 767-778.

C. Dennis, J. B., MATHEMATICAL PROGRAMMING AND ELECTRICAL NETWORKS, John Wiley & Sons, New York, (1959).

D. Dennis, J. B., "Distributed Solution of Network Programming Problems," IEEE Transactions on Communication Systems, CS-12, No. 2, pp. 176-184, (June, 1964).

E. Ford, L. K.,and D. R. Fulkerson, FLOWS IN NETWORKS, Princeton University Press, Princeton, New Jersey, (1962).

F. Ford, L. K., and D. R. Fulkerson, Revised version of (84), Management Science, pp. 97-101, (October, 1958).

G. Gomory, R. E., and T. C. Hu, "Synthesis of a Communication Network," Journal of the Society of Industrial and Applied Mathematics, Vol. 12, No. 2, pp. 348-369, (June, 1964).

H. Hakimi, S. L., "Simultaneous Flows Through a Communication Network," IRE Transactions on Circuit Theory, Vol. CT-9, No. 2, pp. 169-175, (June, 1962).

I. Haley, K. B., "The Solid Transportation Problem," Operations Research, Vol. 10, No. 4, pp. 448-463, (July-August, 1962).

J. Haley, K. B., "The Multi-Index Problem," Operations Research, Vol. 11, No. 3, pp. 368-379, (May-June, 1963).

K. Haley, K. B., "The Existence of a Solution to the Multi-Index Problem," Operational Research Quarterly (U.K.), Vol. 16, No. 4, pp. 471-474, (1965).

L. Haley, K. B., "A General Method of Solution for Special Structure Linear Programs," Operational Research Quarterly, (U.K.), Vol. 17, No. 1, pp. 83-90, (1966).

M. Hu, T. C., "Multi-Commodity Network Flows," Operations Research, Vol. 11, No. 3, pp. 344-360, (May-June, 1963).

N. Hu, T. C., "On the Feasibility of Simultaneous Flows in a Network," Letters to the Editor, Operations Research, Vol. 12, No. 2, pp. 359-360, (March-April, 1964).

O. Iri, M., "A New Method of Solving Transportation-Network Problems," Journal of the Operations Research Society of Japan, Vol. 3, No. 1 & 2, pp. 27-87, (October, 1960).

P. Jewell, W. S., "Optimal Flow Through Networks," (Sc. D. Thesis), Interim Technical Report No. 8, Army of Ordnance Research: Fundamental Investigations in Methods of Operations Research, Center for Operations Research, Massachusetts Institute of Technology, Cambridge, Mass.,146 pp., (June, 1958).

Q. Jewell, W. S., "Optimal Flow Through Networks," Paper presented at the 13th National Meeting of the Operations Research Society of America, Boston, Massachusetts, (May 15-17, 1958).

R. Jewell, W. S., "Optimal Flow With Gains," Operations Research, Vol. 10, No. 4, pp. 476-422, (July-August, 1962), (presented at the Second International Conference on Operational Research, Aix-ex-Provence, September, 1960).

S. Jewell, W. S., "Multi-Commodity Network Solutions," ORC 66-23, Operations Research Center, University of California, Berkeley, (September, 1966), [To appear in Theórie des Graphes,Actes des Journeés Internationales d' Etudes de l' I.C.C. - Rome 1966, Dunod, Paris, (1967)]

T. Jewell, W. S., "Models of Traffic Assignment," Transportation Research, Vol. 1, No. 1, pp. 1-16, (June, 1967).

U. Jewell, W. S., "Complex: A Complementary Slackness, Out-of-Kilter Algorithm for Linear Programming," ORC 67-6, Operations Research Center, University of California, Berkeley, (March, 1967).

V. Matthys, G., "Flot Optimum dans un Reseau à Capcités de Faisceaux," Proceedings of the Second International Conference on Operational Research, pp. 164-170, Aix-ex-Provence, English Universities Press, ltd., London, p. 809, (September, 1960).

W. Rothschild, B., and A. Whinston, "On Two-Commodity Network Flows," Operations Research, Vol. 14, No. 3, pp. 377-387, (May-June, 1966).

X. Rothschild, B., and A. Whinston, "Feasibility of Two-Commodity Network Flows," Operations Research, Vol. 14, No. 6, pp. 1121-1129, (November-December, 1966).

Y. Sakarovitch, M., "The Multi-Commodity Maximum Flow Problem," ORC 66-25, Operations Research Center, University of California, Berkeley, (December, 1966).

Z. Sunoga, T., and M. Iri, "Theory of Communication and Transportation Networks," RAAG Memoirs (Japan), Vol. 2, pp. 444-468, (1958).

AA. Tang, D. T., "Communication Networks with Simultaneous Flow Requirements," IRE Transactions on Circuit Theory, Vol. CT-9, No. 2, pp. 176-182, (1962).

BB. Tang, D. T., "Bi-Path Networks and Multi-Commodity Flows," IEEE Transactions on Circuit Theory, Vol. CT-11, (December, 1964).

CC. Tang, D. T., "Comments on Feasibility Conditions of Simultaneous Flows in a Network," Letters to the Editor, Operations Research, Vol. 13, No. 1, pp. 143-146, (January-February, 1965).

DD. Tomlin, J. A., "Minimum Cost Multi-Commodity Network Flows," Operations Research, Vol. 14, No. 1, pp. 45-57, (January-February, 1966).