# ON THE MAX FLOW MIN CUT THEOREM OF NETWORKS

G. B. Dantzig
D. R. Fulkerson

P-826

April 15, 1955

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

## SUMMARY

It is shown that Menger's theorem and the Max Flow Min Cut Theorem on networks are applications of the duality theorem of linear inequality theory.

# ON THE MAX FLOW MIN CUT THEOREM OF NETWORKS

by

G. B. Dantzig
D. R. Fulkerson

The RAND Corporation

## INTRODUCTION

The problem discussed in this paper arises naturally in the study of transportation networks. Roughly stated, it is as follows. Consider a network connecting two nodes by way of a number of intermediate nodes, and suppose the arcs and nodes can handle certain designated amounts of traffic per unit time. Assuming a steady state condition, find a maximal flow of traffic from one given node (the source) to the other (the sink).

For example, let the network be that of Fig. 1

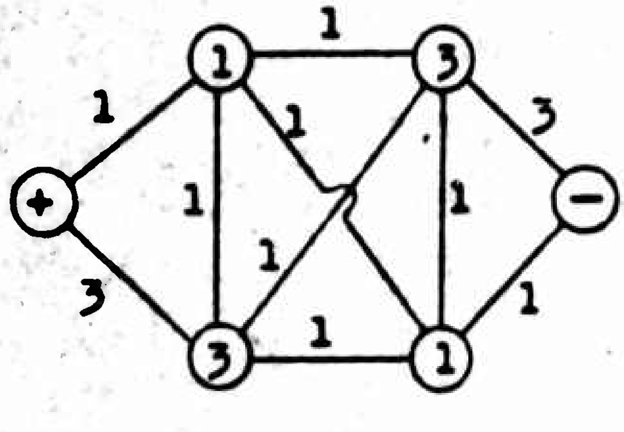

Fig. 1

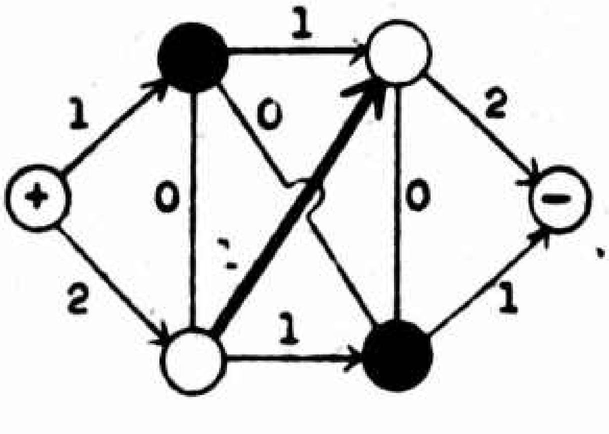

Fig. 2

where source and sink are denoted by + and − respectively, with flow capacities of the arcs and nodes as indicated. A maximal flow from source to sink is shown in Fig. 2. Notice that the quantity of flow leaving the source (or entering the sink) is equal to the sum of the capacities of the two nodes and one arc which are emphasized in the figure, and that this collection of nodes and arcs forms a "cut" in the network; i.e., meets every chain from source to sink.

A nonconstructive proof based on convexity arguments has been given in an unpublished manuscript by L. R. Ford, Jr. and one of the present writers (D. R. Fulkerson) that the maximal flow value, relative to a given source and sink, attainable in any network is equal to the minimal sum of capacities on arcs and nodes constituting a cut. Our aim is to formulate the problem of finding a maximal flow as a linear programming problem (§1) and to deduce the max flow min cut theorem from the dual problem (§3). In §4 we observe that a combinatorial form of this theorem yields Menger's theorem [4, p.244] concerning linear graphs.

## 1. THE PROGRAMMING PROBLEM

There are various ways of formulating the flow problem as a linear programming problem. One way[1], convenient from both

[1] A. Hoffman has given a different formulation of the problem which also yields the max flow min cut theorem. While the techniques employed in his approach are similar to those of this paper, he uses an entirely different set of variables which are of interest in themselves.

a computational and theoretical viewpoint, is as follows. Set up the pseudo transportation array

$$\begin{array}{c|ccccc|c}
 & & & & & & \text{Totals}\\
 & -x_{oo} & x_{o1} & x_{o2} & \cdots & x_{on} & 0\\
 & x_{1o} & -x_{11} & x_{12} & \cdots & x_{1n} & 0\\
 & x_{2o} & x_{21} & -x_{22} & \cdots & x_{2n} & 0\\
 & \vdots & \vdots & \vdots & \ddots & & \vdots\\
 & x_{no} & x_{n1} & x_{n2} & \cdot & -x_{nn} & 0\\
\text{Totals} & 0 & 0 & 0 & \cdots & 0 &
\end{array}$$

schematizing the equations

(1a) $$-x_{ii} + \sum_{j,\, j\neq i} x_{ij} = 0 \qquad (i=0,1,\ldots,n)$$

(1b) $$-x_{jj} + \sum_{i,\, i\neq j} x_{ij} = 0 \qquad (j=0,1,\ldots,n),$$

where $x_{ij} \geq 0$ (i;j=1,...,n; i≠j) denotes the flow from node i to node j, $x_{ii} \geq 0$ (i=1,...,n) represents the total flow through node i, $x_{oj} \geq 0$ (j=1,...,n) is the flow from the source to node j, and $x_{io} \geq 0$ (i=1,...,n) the flow from node i to the sink. Thus $x_{oo}$ is the total flow through the network and the problem is to maximize $x_{oo}$ subject to (1a), (1b), and

(2) $$x_{ij} + x_{ji} \leq c_{ij} \qquad (c_{ij} = c_{ji})$$

(3) $$x_{ii} \leq c_{ii} \qquad (i,j = 1,\ldots,n)$$

$$x_{io} \leq c_{io}$$

$$x_{oj} \leq c_{jo},$$

(4) $$x_{ij} \geq 0 \qquad (i,j = 0,\ldots,n),$$

where the c's are given nonnegative constants. We have formally included all variables $x_{ij}$ in the problem; arcs not present in the network have $c_{ij} = 0$.

Because of (2), we refer to this as the undirected problem; that is, except for source and sink arcs, the direction of flow is not specified in the arcs.

Given an undirected problem,it is easy to describe an equivalent directed problem. Simply replace each undirected arc by a pair of oppositely directed arcs, each with capacity equal to that of the original arc. That the two problems are equivalent follows from the fact that given any $x = (x_{ij})$ satisfying (1), (3), (4), and

(2') $$x_{ij} \leq c_{ij}$$

$$x_{ji} \leq c_{ji} \qquad (c_{ij}=c_{ji}),$$

a flow x' of equal value is obtained by setting

$$x'_{io} = x_{io} \qquad (i = 0,\ldots,n)$$

$$x'_{oj} = x_{oj} \qquad (j = 0,\ldots,n)$$

$$x'_{ij} = \max\,(x_{ij} - x_{ji}, 0) \qquad (i,j = 1,\ldots,n;\ i \neq j)$$

$$x'_{ii} = x_{ii} - \sum_{\substack{j=1 \\ j\neq i}}^{n} \min\,(x_{ij}, x_{ji}) \qquad (i = 1,\ldots,n).$$

Thus (2) may be replaced by (2') without changing the value of a maximal flow.

A cut in an undirected network has been defined as a collection of arcs and nodes meeting every chain joining source and sink; a cut in a directed network is similarly defined as a collection of directed arcs and nodes meeting every directed chain from source to sink. The value of a cut in either case is the sum of the capacities of all its member nodes and arcs. One proves easily that the minimal cut value is the same for an undirected network and its equivalent directed network. Thus, to prove the max flow min cut theorem, it suffices to consider directed networks only. Accordingly, we shall make no further use of the condition $c_{ij} = c_{ji}$ in (2').

Let us now rewrite the inequalities (2'), (3), as

$$(5) \qquad x_{ij} + y_{ij} = c_{ij},\ y_{ij} \geq 0 \quad (i,j = 0,1,\ldots,n)$$

where, to avoid special cases, we have included a sufficiently large upper bound of $c_{oo}$ on the variable $x_{oo}$; for example, choose $c_{oo} > \sum_{i=1}^{n} c_{io}$. Then the problem is to maximize $x_{oo}$ subject to (1a), (1b), (4), and (5).

## 2. BASES

We turn momentarily to the question of what constitutes a basis,[12] in order to note that the Hitchcock-Koopmans transportation theory carries over to the flow problem.

There are $(n+1)^2 + 2(n+1) - 1$ linearly independent equations in the set (1a), (1b), (5), since one of the set (1a), (1b), is redundant. Drop the first equation of (1a) as the redundant one, and denote by $X_{ij}$, $Y_{ij}$ the column vectors of the coefficient matrix of the remaining equations corresponding to the variables $x_{ij}$, $y_{ij}$, respectively.

It is clear that at least one of $X_{ij}$, $Y_{ij}$ must belong to any basis B. Thus the pairs ij fall into one of three classes:

$\alpha$: those ij for which $X_{ij} \in B$, $Y_{ij} \in B$;

$\beta$: those ij for which $X_{ij} \in B$, $Y_{ij} \notin B$;

$\gamma$: those ij for which $X_{ij} \notin B$, $Y_{ij} \in B$.

The number of pairs ij of type $\alpha$ is always 2n+1. For if there are k of type $\alpha$, hence $(n+1)^2 - k$ of types $\beta$ and $\gamma$, then

[12] Let $\sum_{j=1}^{n} a_{ij}x_j = b_i$, $x_j \geq 0$ $(i = 1,\ldots,m)$ be the constraints of a linear programming problem, and suppose $A = (a_{ij})$ has rank m. A set of m linearly independent columns of A is a "basis", the corresponding $x_j$ are "basic variables". The vector $\hat{x} = (\hat{x}_1,\ldots,\hat{x}_n)$ obtained by assigning non-basic variables zero values and solving the resulting equations for the basic variables is called a "basic solution". If $\hat{x}$ has non-negative components, it is termed a "basic feasible solution". Geometrically, basic feasible solutions correspond to extreme points of the convex set defined by the constraints.

$2k + (n + 1)^2 - k = (n + 1)^2 + 2(n + 1) - 1$, $k = 2n + 1$. Moreover, it is impossible to find among the pairs of type $\alpha$ a subset of the form

$$i_1j_1,\ i_1j_2,\ i_2j_2,\ i_2j_3, \ldots,\ i_kj_k,\ i_kj_1$$

where the i's and j's are distinct among themselves, as otherwise the column vectors $X_{ij}$, $Y_{ij}$ corresponding to these pairs can easily be shown to be dependent.

These two statements together imply that B can be arranged in triangular form, just as in the Hitchcock-Koopmans case. To see this, it is convenient to associate a linear graph G with B, and to look at the problem of finding the basic solution corresponding to B in terms of this graph.[12]

Let $a_o, \ldots, a_n$, $b_c, \ldots, b_n$ be the nodes of G; the arcs of G are those $a_ib_j$ for which ij is of type $\alpha$. As we have seen, G has 2n + 1 arcs and contains no cycles. It is therefore a tree. Call a node of G which has only one arc on it an "end-node". There are at least two such.

We associate with node $a_i(b_j)$ of G the equation $-x_{ii} + \sum_{j, j \neq i} x_{ij} = 0$ $(-x_{jj} + \sum_{i, i \neq j} x_{ij} = 0)$. Now locate an end-node,

[12] There are several alternative ways one can view the equation solving process in terms of a linear graph. Since the equations came from such a graph, one way would be to use the original network. This appears to be most efficient for hand-computation. Another way, in terms of the array (1), can be developed as in [2]. A third way, the one we adopt, is suggested in [1b]. In all of these, the notion of "basis" in the programming sense is closely related to the notion of "tree" in the graph sense.

say $a_k$, and let its arc be $a_k b_\ell$. Since for pairs ij of type $\beta$, $y_{ij} = 0$, $x_{ij} = c_{ij}$, and similarly $x_{ij} = 0$, $y_{ij} = c_{ij}$ for pairs of type $\gamma$, all the variables of the equation $-x_{kk} + \sum_{j, j \neq k} x_{kj} = 0$ are determined but one, $x_{k\ell}$, and thus its value may be found immediately. Then use (5) to get $y_{k\ell} = c_{k\ell} - x_{k\ell}$. Delete $a_k$ and $a_k b_\ell$ from G, leaving a tree, and repeat the procedure. After 2n + 1 steps, the values of all variables are determined.

Notice that only addition and subtraction are required. Thus, if the $c_{ij}$ are integral, so are the values of all variables in a basic solution, hence in a basic feasible solution. We will make use of this fact in the concluding section.

## 3. SIMPLEX CRITERION AND THE DUAL PROBLEM

Let $u_i$, $v_j$, $w_{ij}$ be the multipliers (dual variables) corresponding to the equations (1a), (1b), (5), respectively, in applying the simplex algorithm. Then the conditions for an optimal basis B* are:

(6a) $\quad -u_i - v_i + w_{ii} \geq \delta_{i0} \qquad (\delta_{00} = 1,\ \delta_{i0} = 0 \text{ for } i > 0)$

(6b) $\quad u_i + v_j + w_{ij} \geq 0 \qquad (i \neq j)$

(6c) $\quad w_{ij} \geq 0$

with equality holding in (6a), (6b), if the corresponding $X_{ij} \in B^*$, in (6c) if $Y_{ij} \in B^*$. Ignoring the redundant equation amounts to taking $u_o = 0$. Then, since $X_{oo}$, $Y_{oo} \in B^*$;[14] $w_{oo} = 0$, $v_o = -1$. For all other pairs ij of type $\alpha$, $w_{ij} = 0$, and the equations $u_i + v_j = 0$ hold. It follows that all $u_i = 0$ or 1, all $v_j = 0$ or $-1$. (A convenient way to see this is to associate the variable $u_i(v_j)$ with node $a_i(b_j)$ of the graph $G^*$ corresponding to $B^*$ and the equations $u_i + v_j = 0$ with the appropriate arcs of $G^*$). Substituting these values into (6a), (6b), to determine the $w_{ij}$ corresponding to $Y_{ij} \notin B^*$ and noting (6c) shows that all $w_{ij} = 0$ or 1.

The dual programming problem is to minimize $\Sigma c_{ij} w_{ij}$ subject to (6a) – (6c), and the multipliers corresponding to an optimal primal solution solve the dual problem. Thus

$$(7) \qquad \max x_{oo} = \min \Sigma c_{ij} w_{ij} = \Sigma_{\sigma} c_{ij}$$

where $\sigma$ is that set of pairs ij corresponding to $w_{ij} = 1$; in terms of the network, $\sigma$ is some subset of those (directed) arcs and nodes which are at capacity in the flow x. We claim that $\sigma$ is a cut. For suppose all $c_{ij}$, $ij \notin \sigma$, are increased by $\epsilon > 0$. This does not change the solution to the dual, hence

[14] Our choice of $c_{oo}$ implies that $Y_{oo} \in B$ for any B yielding a basic feasible solution; also clearly $X_{oo} \in B^*$ except possibly in the trivial case where the maximal flow over the network is zero. The assertion is valid in general, however, as otherwise all multipliers have zero values, violating (6a) with $i = 0$.

cannot increase the flow in the network. But if there were some directed chain from source to sink not meeting $\sigma$, the maximal flow value would be increased by at least $\epsilon$. Thus $\sigma$ is a cut, and since it is clear that no flow can exceed the value of any cut, the proof of the max flow min cut theorem is complete.

## 4. MENGER'S THEOREM

Given an arbitrary linear graph G, let $I_1$, $I_2$ be two disjoint sets of nodes of G. Menger's theorem states that the maximal number of pairwise node-disjoint chains joining $I_1$ to $I_2$ is equal to the minimal number of nodes necessary to separate $I_1$ from $I_2$. To deduce this theorem from the max flow min cut theorem, join all the nodes of $I_1$ to a new node, the source, and all the nodes of $I_2$ to another new node, the sink; then assign unit capacity to each of the old nodes, infinite capacity to each arc. Menger's theorem now follows by selecting a maximal flow x with integral components.

## REFERENCES

[1] Activity Analysis of Production and Allocation, Cowles Commission monograph No. 13, T. C. Koopmans, Ed., John Wiley and Sons, Inc., New York, 1951.
(a) Dantzig, G. B., "Application of the Simplex Method to a Transportation Problem," pp. 359-373.
(b) Koopmans, T. C., and S. Reiter, "A Model of Transportation," pp. 222-259.

[2] Flood, M. M., "On the Hitchcock Distribution Problem," Pac. Jour. Math., Vol. 3, No. 2, June, 1953, pp. 369-386.

[3] Hitchcock, F. L., "The Distribution of a Product From Several Sources to Numerous Localities," Jour. Math. Physics, 20, 1941, pp. 224-230.

[4] König, D., Theorie der Endlichen und Unendlichen Graphen, Chelsea Publishing Co., New York, N.Y., 1950.