Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Technical Note—Maximal Funnel-Node Flow in an Undirected Network

John J. Jarvis, Duane D. Miller,

Maximal Funnel-Node Flow in an Undirected Network

John J. Jarvis

Georgia Institute of Technology, Atlanta, Georgia

and

Duane D. Miller

Office of the Assistant Vice Chief of Staff, United States Army, Washington, D.C.

(Received May 25, 1971)

This note formulates, examines, and solves a funnel-node maximal-flow problem for an undirected network. The solution procedure requires only applications of the single-commodity flow algorithm, and is therefore extremely efficient. Several applications are presented.

CONSIDER THE following transportation problem: Supplies are located at a supply terminal for shipment to a destination terminal. The only vehicles available to transport the supplies are located at still another terminal. Given a road network of limited capacity, it is desired to determine the routing of vehicles that will allow the maximum number of vehicles to proceed from the vehicle terminal through the supply terminal to the destination terminal. The road network can be represented by an abstract undirected network $G(N; E)$ as shown in Fig. 1.

Now consider a second problem. A communication-systems designer is required to establish a message center for an existing communication network. All messages are to pass through the message center and the message center must be located at an existing installation. It is desirable to maintain the maximum possible message flow under the given conditions. The communication network can be represented as shown in Fig. 2.

These two problems are examples of the type of problem with which this paper will be concerned. In each case a special node is singled out, through which all flow from source to sink must pass. We define this special node as a *funnel-node*.

A mathematical statement of the funnel-node max-flow problem is:

$$\text{maximize } v(s; a; t), \qquad \text{subject to}$$

$$\sum_j [f_1(N_i, N_j) - f_1(N_j, N_i)] = \begin{cases} v(s; a), & \text{if } i = s, \\ 0, & \text{if } i \neq s, a, \\ -v(s; a), & \text{if } i = a, \end{cases}$$

$$\sum_j [f_2(N_i, N_j) - f_2(N_j, N_i)] = \begin{cases} v(a; t), & \text{if } i = a, \\ 0, & \text{if } i \neq a, t, \\ -v(a; t), & \text{if } i = t, \end{cases}$$

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

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

where $v(x; y)$ is the value of the flow from node x to node y in the network and $v(s; a; t)$ is the value of the flow from s to t passing through the funnel-node a.

The similarity of the funnel-node max-flow problem and the two-commodity max-flow problem is evident (*see* Hu[8]). The constraint sets are identical with

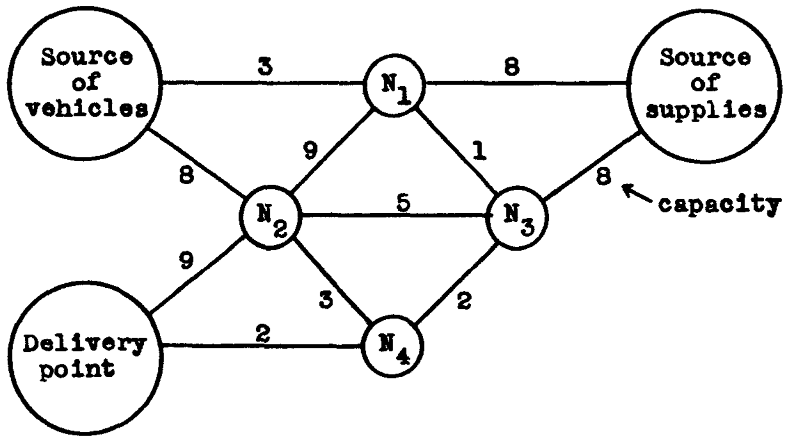

Fig. 1. A road network.

the exception that, in the funnel-node problem, (i) the first source is s; the second sink is t, (ii) the first sink and second source are both a, and (iii) the flows of commodity 1 and of commodity 2 are both $v(s; a; t)$. Now with the addition of the

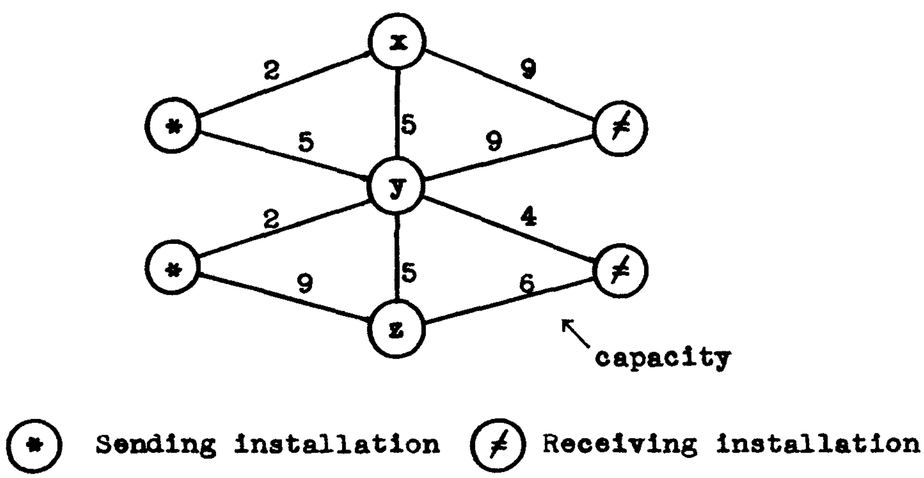

Fig. 2. A communication network.

last constraint, the function to be maximized is exactly one-half of the one that is to be maximized in the two-commodity max-flow problem.

Hu[8] has solved the two-commodity max-flow problem by using an algorithm involving flow exchanges that determines both $\max[v(s_1; t_1)+v(s_2; t_2)]$ and the appropriate routing of flows through the network. If we define $v^*(x; y)$ to be the maximal value of $v(x; y)$, then the first of T. C. Hu's results is that a solution to the two-commodity max-flow problem always exists, and there are at least two solutions (possibly identical) of which one has the property that $v(s_1; t_1) = v^*(s_1; t_1)$ and the other has the property that $v(s_2; t_2) = v^*(s_2; t_2)$.

Let $v^+(s_1; t_1) = \max[v(s_1; t_1) + v(s_2; t_2)] - v^*(s_2; t_2)$ and $v^+(s_2; t_2) = \max[v(s_1; t_1) + v(s_2; t_2)] - v^*(s_1; t_1)$.

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

$$\max[v(s_1; t_1) + v(s_2; t_2)] = v^*(s_1; t_1) + v^+(s_2; t_2) = v^+(s_1; t_1) + v^*(s_2; t_2).$$

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

THE FUNNEL-NODE MAX-FLOW THEOREM

THEOREM. $v^*(s; a; t) = \min\{v^*(s; a), v^*(a; t), \frac{1}{2}\max[v(s; a) + v(a; t)]\}$.

Proof. It is clear that $v^*(s; a; t)$ cannot exceed $v^*(s; a)$ or $v^*(a; t)$. Also $\frac{1}{2}\max[v(s; a) + v(a; t)] \geqq \frac{1}{2}\max_{v(s;a)=v(a,t)}[v(s; a) + v(a; t)] \geqq \frac{1}{2}\max[2v(s; a; t)] \geqq v^*(s; a; t)$. Thus $v^*(s; a; t) \leqq \min\{v^*(s; a), v^*(a; t), \frac{1}{2}\max[v(s; a) + v(a; t)]\}$ and it is only necessary to show that equality holds. There are two cases.

Case 1. $v^*(s; a) \leqq v^+(a; t)$ or $v^*(a; t) \leqq v^+(s; a)$. Suppose $v^*(s; a) \leqq v^+(a; t)$. Now, from the flow solution yielding $v^*(s; a)$ and $v^+(a; t)$, reduce flow along paths from a to t successively up to an amount $\delta = [v^+(a; t) - v^*(s; a)]$. This new flow is a feasible funnel-node flow and has value $v^*(s, a)$. Therefore, it must be optimal.

The result is similar for $v^*(a; t) \leqq v^+(s; a)$.

Case 2.

$$v^*(s; a) > v^+(a; t), \tag{1}$$

and

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

Let

$$\begin{aligned} v_0 &= \max[v(s; a) + v(a; t)] \\ &= v^*(s; a) + v^+(a; t) \qquad (3) \\ &= v^+(s; a) + v^*(a; t). \qquad (4) \end{aligned}$$

Equations (1) and (3) imply

$$v_0/2 \leqq v^*(s; a). \tag{5}$$

Equations (2) and (4) imply

$$v_0/2 \leqq v^*(a; t), \tag{6}$$

and obviously

$$v_0/2 + v_0/2 \leqq v_0. \tag{7}$$

But (5), (6), and (7) are exactly conditions (1), (2), and (3) of Theorem 1 of Hu.[8] Hence, there exists a two-commodity flow with $v(s; a) = \frac{1}{2}v_0$ and $v(a; t) = \frac{1}{2}v_0$.

THE ALGORITHM

THE FUNNEL-NODE, max-flow theorem leads directly to the following algorithm for the construction of maximal funnel-node flows in an undirected network.

Step 1. Solve for $v^*(s;a)$ and $v^*(a;t)$ using the single-commodity flow algorithm.

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

Step 3. Determine

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

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

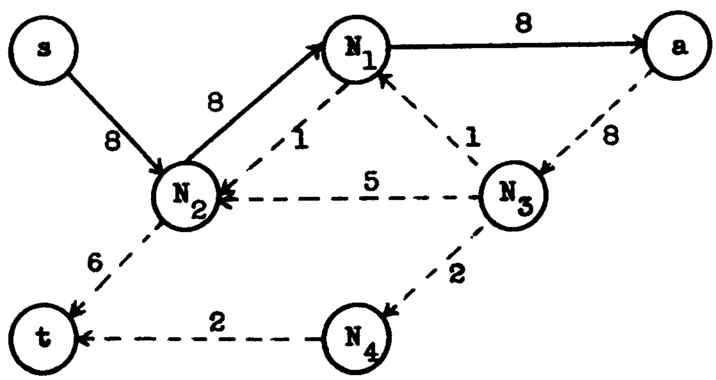

Fig. 3. The flow pattern for $v^*(s;a;t)$; the optimal flow for Fig. 1 has a value of 8.

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

The use of $v^*(s';a)$ to determine the value of $\max[v(s;a)+v(a;t)]$ in Step 2 of the algorithm is a consequence of a second result of Hu:

$$\max[v(s_1;t_1)+v(s_2;t_2)]=\min[c(s_1-s_2;t_1-t_2),\, c(s_1-t_2;t_1-s_2)].$$

[He defines $c(x-y;z-w)$ as the capacity of the minimal cut that has nodes x and y in one set with z and w in the other set.] But, for our problem, $s_1=s$, $s_2=t_1=a$, and $t_2=t$. Making these substitutions, Hu's result is

$$\max[v(s;a)+v(a;t)]=\min[c(s-a;a-t),\, c(s-t;a)].$$

But $(s-a;a-t)$ implies that a is in both components of G, which violates the definition of a cut. Therefore this case cannot exist and $\max[v(s;a)+v(a;t)]=c(s-t;a)$. Thus we can determine $\max[v(s;a)+v(a;t)]$ by computing $v^*(s-t;a)$ or, equivalently, by computing $v^*(s';a)$.

Figure 3 shows the optimal flow (of value 8) for Fig. 1.

ACKNOWLEDGMENTS

THE AUTHORS are indebted to the referee for his simplifying suggestions. This research was partially supported by the Army Research Office-Durham.

REFERENCES

1. J. ABINAL, "Maximal Biflow in an Undirected Network," *IBM J. of Res. and Develop.* **13,** 373–379 (1969).
2. L. R. FORD, JR., AND D. R. FULKERSON, *Flows in Networks*, Princeton University Press, Princeton, New Jersey, 1962.
3. T. C. HU, "Multicommodity Network Flows," *Opns. Res.* **11,** 344–360 (1963).
4. W. ROTHFARB, N. P. SHEIN, AND I. T. FRISCH, "Common Terminal Multicommodity Flow," *Opns. Res.* **16,** 202–205 (1968).
5. B. ROTHSCHILD, AND A. WHINSTON, "On Two Commodity Network Flows," *Opns. Res.* **14,** 202–205 (1966).
6. ——— AND ———, "Maximal Two-Way Flows," *SIAM J. Appl. Math.* **15,** 1228–1238(1967).

On a Duality Theorem for a Nonlinear Programming Problem

Bertram Mond

La Trobe University, Melbourne, Australia

(Received October 21, 1971)

Duality theorems were given recently for a mathematical program with a nonlinear nondifferentiable objective function. Here we point out that the converse dual theorem holds without the usually stated restrictions and assumptions.

CONSIDER THE following pair of problems:

$$\text{I: Maximize } f(x)=p^t x-\sum_{i=1}^{m}(x^t D^i x)^{1/2}, \quad \text{subject to } Ax\leqq b,\ x\geqq 0. \tag{1}$$

$$\text{II: Minimize } g(y)=b^t y, \quad \text{subject to:}$$

$$A^t y+\sum_{i=1}^{m} D^i w^i \geqq p; \tag{2}$$

$$w^{it} D^i w^i \leqq 1, \qquad i=1,\cdots,m; \tag{3}$$

$$y\geqq 0. \tag{4}$$

Here D^i, $i=1,\cdots,m$, are symmetric positive semidefinite matrices. Problem I

Published in Operations Research on February 01, 1973 as DOI: 10.1287/opre.21.1.365.
This article has not been copyedited or formatted. The final version may differ from this version.