BULLETIN OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 83, Number 3, May 1977

A GOOD ALGORITHM FOR LEXICOGRAPHICALLY OPTIMAL FLOWS IN MULTI-TERMINAL NETWORKS

BY NIMROD MEGIDDO

Communicated by Alston S. Householder, December 27, 1976

ABSTRACT. Let a network have several sources and sinks. For any flow f let σ^f and τ^f denote the vectors of net flows out of the sources and into the sinks, respectively, arranged in order of increasing magnitude. Our algorithm computes an f for which both σ^f and τ^f are lexicographic maxima. For a network with n nodes this algorithm terminates within $O(n^5)$ operations.

1. The problem. A *network* (N, c) consists of a set of *nodes* $N = \{1, \ldots, n\}$ $(n \geqslant 1)$ and an $n \times n$ nonnegative matrix c of *capacities*. $S \subset N$ is a nonempty set of *sources* and $T \subset N$ $(T \cap S = \emptyset)$ is the set of *sinks*. A flow f is an $n \times n$ matrix such that $0 \leqslant f_{ij} \leqslant c_{ij}$ $(i, j \in N)$ and $\Sigma_{j=1}^n (f_{ij} - f_{ji}) = 0$ for $i \notin S \cup T$. Denote $s = |S|$, $t = |T|$.

Let σ^f $[\tau^f]$ denote the s-tuple [t-tuple] of the numbers $\Sigma_{j=1}^n(f_{ij} - f_{ji})$, $i \in S$ $[\Sigma_{j=1}^n (f_{ji} - f_{ij}), i \in T]$ arranged in order of increasing magnitude. f is called *optimal* if it maximizes both σ^f and τ^f in the lexicographic orders on R^s and R^t, respectively.

Optimal flows reduce to maximum flows (see [5]) when $s = t = 1$. Existence of optimal flows is proved in [7]. The goal of this note is to present a good algorithm (in the sense of [2]) for finding an optimal flow.

2. The algorithm. The algorithm has two phases. In Phase I the network is decomposed to two networks, one with a single source and t sinks, and the other with s sources and a single sink. In Phase II optimal flows are found in these two networks.

PHASE I. Find a flow f which maximizes $\Sigma_{i\in S} \Sigma_{j=1}^n(f_{ij} - f_{ji})$. Any of the following algorithms may be utilized: Karzanov [6] terminates within $O(n^3)$ operations, Dinic [1] and Even and Tarjan [4] $O(n^4)$, and Edmonds and Karp [3] $O(n^5)$. During the computation of f a set X, $S \subset X \subset N\backslash T$, is generated such that for $i \in X$ and $j \notin X$, $f_{ij} = c_{ij}$ and $f_{ji} = 0$. Next, construct the X-condensed and the $(N\backslash X)$-condensed networks (see [7]).

PHASE II. Find optimal flows in the X-condensed and the $(N\backslash X)$-condensed networks independently. These two are treated symmetrically and, hence, without loss of generality assume that $S = \{1\}$.

AMS (MOS) subject classifications (1970). Primary 90B10, 90C35; Secondary 05C35, 94A20, 68A10.

Key words and phrases. Optimal flow, multi-terminal network, tree, polynomial-time algorithm.

The following terminology is used throughout Phase II. The *flow* is the current flow through the network. An *active sink* is a sink i such that it is possible to increase the net flow into i without decreasing the net flows into other sinks. The *graph* is a directed graph whose set of nodes is N and (i, j) is an arc if and only if $f_{ij} - f_{ji} < c_{ij}$. The *manual* is the subgraph consisting of all nodes and arcs of the graph that lie on a shortest chain from the source to some active sink. The *tree* is a directed subtree of the manual satisfying: a. The source is the root of the tree. b. Every active sink is a node of the tree. c. Every tip of the tree is an active sink.

A rough description of Phase II follows. We start with the zero flow. The manual is constructed and a tree is chosen. Next, using tree-arcs only, we increase the net flows into active sinks equally until one of the tree-arcs leaves the graph. Then, if all the active sinks remain active, another tree is chosen and another maximum equal increase is employed as before; otherwise, the new manual is determined and again a tree is chosen. The algorithm terminates when the new manual is empty, or equivalently, when no sink is active. Specifically, Phase II is processed as follows.

Step 1. *Construction of the manual.* The set M of the nodes of the manual is partitioned into *layers* $M_0, M_1, \ldots, M_r$. These are constructed as follows. First, set $M_0 = S$. Then, set M_ν $(\nu > 0)$ to consist of all graph-nodes adjacent to nodes in $M_{\nu-1}$ but do not belong to $\bigcup_{\lambda=0}^{\nu-1} M_\lambda$. Let M_r be the first layer such that either $\bigcup_{\lambda=0}^{r} M_\lambda$ contains all the active sinks, or $M_{r+1} = \varnothing$. In the latter case every sink $i \notin \bigcup_{\lambda=0}^{r} M_\lambda$ becomes inactive. If every sink is inactive, terminate. Next, every $i \in M_r$ which is not an active sink is deleted from M_r and, recursively, every $i \in M_{\nu-1}$ which is neither an active sink nor adjacent to a node in M_ν, is deleted from $M_{\nu-1}$. During the computation we maintain, for each $i \in M_\nu$, lists of the nodes in $M_{\nu-1}$ and $M_{\nu+1}$ which are adjacent to i.

Step 2. *Construction of a tree.* Every $i \in M_r$ is a node of the tree. Suppose that the part of the tree that connects the layers $M_\nu, \ldots, M_r$ $(\nu \leq r)$ has been specified. For each $i \in M$ which either belongs to the tree or is an active sink, let j be the first node in $M_{\nu-1}$ adjacent to i. Then (j, i) also belongs to the tree. During the construction keep track of the numbers v^i of active sinks j such that i lies on the unique chain of the tree that leads from the source to j.

Step 3. *Flow change.* Let ϵ be the minimum, taken over tree-arcs, of the numbers $(c_{ij} - f_{ij} + f_{ji})/v^j$. For each tree-arc (i, j), if $\epsilon v^j \leq f_{ji}$ set $f_{ji} = f_{ji} - \epsilon v^j$; otherwise, set $f_{ij} = f_{ij} - f_{ji} + \epsilon v^j$ and $f_{ji} = 0$.

Step 4. *Manual change.* During this step we maintain a list L of manual-arcs which are successively deleted. Initially, L consists of those arcs of the last tree such that $f_{ij} - f_{ji} = c_{ij}$. Let (i, j) be the first in L which has not been deleted yet. If there is no $k \neq i$ such that (k, j) is a manual-arc then add all manual-arcs of the form (j, k) to the end of L. Similarly, all manual-arcs of the form (k, i) are added to the end of L if there is no manual-arc (i, k) with $k \neq j$.

Then (i, j) is deleted and the next in L is treated. Once $L = \varnothing$, if none of the active sinks has been deleted go to Step 2; otherwise, go to Step 1.

3. Discussion. Let ν^i be the index of that layer of the manual to which the active sink i belongs currently $(0 < \nu^i < n)$. Throughout the computation, ν^i does not decrease and at least one of the $\nu^i - s$ increases when a new manual is constructed. Thus, no more than n^2 manuals are constructed. A tree is constructed within $O(n)$ operations. A flow change results in a deletion of at least one manual-arc. A deletion of a manual-arc requires no more than $O(n)$ operations for updating L. Thus, since a manual has n^2 arcs at most, it will be deleted completely within $O(n^3)$ operations. This implies that the algorithm terminates within $O(n^5)$ operations. Optimality of the final flow follows from the standard theorem on max-flow and augmenting paths (see [5]).

REFERENCES

1. E. A. Dinic, *Algorithm for solution of a problem of maximum flow in a network with power estimation*, Dokl. Akad. Nauk SSSR **194** (1970), 754–757 = Soviet Math. Dokl. **11** (1970), 1277–1280. MR **44** #5178.

2. J. Edmonds, *Paths, trees, and flowers*, Canad. J. Math. **17** (1965), 449–467. MR **31** #2165.

3. J. Edmonds and R. M. Karp, *Theoretical improvements in algorithm efficiency for network flow problems*, J. Assoc. Comput. Mach. **19** (1972), 248–264.

4. S. Even and R. E. Tarjan, *Network flow and testing graph connectivity*, SIAM J. Comput. **4** (1975), 507–518.

5. L. R. Ford, Jr. and D. R. Fulkerson, *Flows in networks*, Princeton Univ. Press, Princeton, N. J., 1962. MR **28** #2917.

6. A. V. Karzanov, *Determining the maximal flow in a network by the method of preflows*, Dokl. Akad. Nauk SSSR **215** (1974), 49–52 = Soviet Math. Dokl. **15** (1974), 434–437.

7. N. Megiddo, *Optimal flows in networks with multiple sources and sinks*, Math. Programming **7** (1974), 97–107. MR **50** #15878.

DEPARTMENT OF STATISTICS, TEL AVIV UNIVERSITY, TEL AVIV, ISRAEL