Dokl. Akad. Nauk SSSR
Tom 194 (1970), No. 4

Soviet Math. Dokl.
Vol. 11 (1970), No. 5

ALGORITHM FOR SOLUTION OF A PROBLEM OF MAXIMUM FLOW IN A NETWORK WITH POWER ESTIMATION

UDC 518.5

E. A. DINIC

Different variants of the formulation of the problem of maximal stationary flow in a network and its many applications are given in [1]. There also is given an algorithm solving the problem in the case where the initial data are integers (or, what is equivalent, commensurable). In the general case this algorithm requires preliminary rounding off of the initial data, i.e. only an approximate solution of the problem is possible. In this connection the rapidity of convergence of the algorithm is inversely proportional to the relative precision.

In the present work an algorithm is presented which solves the problem exactly in the general case after not more than Cn^2p (machine) operations, where n is the number of nodes of the net, p is the number of arcs in it, and C is a constant not depending on the network. For integer data this algorithm, like the algorithm of Ford and Fulkerson, gives an integer solution with a supplementary estimate of the number of operations C_1np plus an estimate of the last.

Formulation of the problem. Designating by a network Γ we shall mean an oriented graph Γ without multiple arcs with isolated nodes s (source) and t (sink), for each arc q of which is prescribed a nonnegative number, its throughput capacity $\rho(q)$. Let $V(\Gamma)$ be the set of nodes, and $Q(\Gamma)$ the set of arcs, of the graph Γ.

By a flow f in the network Γ we shall denote a collection of nonnegative numbers, the magnitudes of the flow $f(q)$, prescribed for all arcs of the network, such that:

$$\text{I.}\quad \operatorname{Div}(v) = \sum_{[v,x]\in Q(\Gamma)} f([v,x]) - \sum_{[y,v]\in Q(\Gamma)} f([y,v]) = 0 \quad \forall v \in V(\Gamma),\ v \neq s, t.$$

$$\text{II.}\quad 0 \leqslant f(q) \leqslant \rho(q) \qquad \forall q \in Q(\Gamma).$$

The quantity $M = \operatorname{Div}(s)$ we shall call the capacity of the flow f.

It is required to find in the given network Γ a flow of maximal capacity.

We shall assume that in the graph Γ for each arc $q = [a, b]$ there exists an opposite arc $\bar{q} = [b, a]$. In fact, the addition of missing arcs with null throughput capacities does not change the problem. The number of arcs is thereby increased by not more than two.

Construction of the algorithm. The proposed algorithm, like the algorithm of Ford and Fulkerson ([1], Chapter I, §8), is an iterative process, each step of which consists in the following: having in the network Γ a flow f, we find a certain path L from s to t from the class of increasing paths (see below for the definition); then we construct a flow f': $M(f') > M(f)$ altering the flow f in the arcs of the path L and in their opposite arcs. The process can be initiated from any flow (integer in the case of integer data), for example, from a null flow, and concludes with the construction of a flow not

Copyright © 1971, American Mathematical Society

admitting increasing paths. In [1], Chapter I, §5, it is proved that such a flow is maximal.

Let there be given in the network a flow f. Let us introduce the quantity $\delta(q) = \rho(q) - f(q) + f(\overline{q})$, defined for each arc of Γ. It can be considered as the throughput capacity of the pair of arcs q, $\overline{q}$ in the direction of q in the presence of the flow f. We shall call an oriented path L from s to t enlarging a flow, if $\forall q \in L$ the condition $\delta(q) > 0$ is fulfilled. This definition is equivalent to the definition used in [1].

Having such a path, one can construct a flow f': $M(f') = M(f) + \Delta M$, where $\Delta M = \min_{q \in L} \delta(q) > 0$. For this it is sufficient to alter the flow in the arcs of the path L and in the arcs opposite to them so that $\forall q \in L$, $[f'(q) - f(q)] + [f(\overline{q}, - f'(\overline{q})] = \Delta M$. We note that under such a change $\delta'(q) = \delta(q) - \Delta M$ and $\delta'(\overline{q}) = \delta(\overline{q}) + \Delta M$ $\forall q \in L$ (here and subsequently values with a prime mark correspond to the flow f'). For the remainder of the arcs, obviously, $\delta'(q) = \delta(q)$.

Let us consider now the problem of finding a path enlarging the flow. Let us agree to consider as the length of a path the number of its arcs, and as the distance $l(a, b)$ between the nodes a and b the length of the shortest path between them.

The basic idea of the algorithm guaranteeing convergence consists in finding at each step of the algorithm a *shortest* enlarging path.

Let us consider the portions $\widetilde{\Gamma}$ of the graph Γ whose arcs are defined by the inequality $\delta(q) > 0$. Clearly, the condition $L \subset \widetilde{\Gamma}$ is necessary and sufficient for the oriented path L from s to t to be a flow-enlarging path. For finding shortest paths from s to t in the graph $\widetilde{\Gamma}$ we employ the following auxiliary entity.

Referent. We shall call the union of all shortest paths from s to t in the graph $\widetilde{\Gamma}$ the referent S of the graph $\widetilde{\Gamma}$. We shall prove several assertions about its structure.

Let $l(s, t) = k$ in the graph $\widetilde{\Gamma}$.

1. $v \in V(S) \iff v \in V(\widetilde{\Gamma}),\ l(s, v) + l(v, t) = k.$

Let us consider a node $v \in V(\widetilde{\Gamma})$ possessing the indicated property. It lies on a shortest path from s to t which is the union of the shortest paths from s to v and from v to t. The converse assertion follows from the fact that any segment of a shortest path is a shortest path between its endpoints.

We reformulate the assertion thus:

1'. $V(S) = \bigcup_{i=0}^{k} V_i$, where $v \in V_i \iff v \in V(\widetilde{\Gamma}),\ l(s, v) = i,\ l(v, t) = k - i.$

2'. $Q(S) = \bigcup_{i=1}^{k} Q_i$, where $[a, b] \in Q_i \iff [a, b] \in Q(\widetilde{\Gamma}),\ a \in V_{i-1},\ b \in V_i.$

The assertion follows almost obviously from the preceding. The direct proof is analogous to that adduced above.

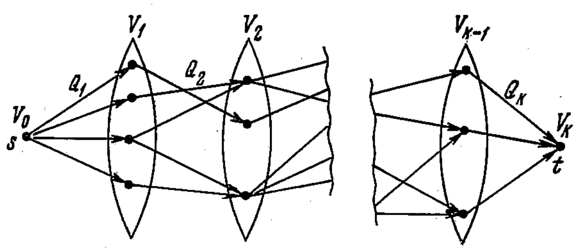

Figure 1

The referent can be constructed as follows. Let the graph $\tilde{\Gamma}$ be given. Let us construct the sets of nodes A_i $(0 \leq i \leq l(s, t))$, where $a \in A_i \Leftrightarrow a \in V(\tilde{\Gamma})$, $l(s, a) = i$. Obviously, $A_0 = \{s\}$. Let the sets A_i $(i < j)$ have been constructed, and the nodes belonging to them marked by the index i. Then, looking over all arcs emanating from the nodes of A_{j-1}, we mark all those not marked up to this vertex by the index j and form from them the set A_j. We conclude the process when t enters into some set A_k. Hence $l(s, t) = k$. Let us now recursively construct the sought sets V_i and Q_i. It is known that $V_k = \{t\}$. Let the set V_j $(j \leq k)$ have been constructed. Then Q_j consists of those arcs which lead from A_{j-1} to V_j, and V_{j-1} consists of the initial nodes of these arcs. Proofs of the method are trivial. We note that it is expedient for the construction if for each node all arcs entering and leaving it are looked over in advance.

Thus, let the referent S of the graph $\tilde{\Gamma}$ exist. Then it is easy to construct any shortest path from s to t, sequentially finding in the referent arbitrary arcs of the form $[s, a_1], [a_1, a_2], \cdots, [a_{k-1}, a_k]$. Here, obviously, $a_i \in V_i$, whence $a_k = t$, which is required. The correctness of the construction follows from the fact that $\forall v \in V(S)$, $v \neq s, t$ $\exists [x, v], [v, y] \in Q(S)$ (by the definition of the referent).

Analogously one can construct a path containing a given arc (or node) of S, building it up to t as well as to s.

Change of referent. We shall show how in almost all cases it is possible to change the referent S to obtain the referent S'.

Lemma. *Let the flow f' be obtained from the flow f by a change with the help of the shortest path L, and $l(s, t) = k$.*

Then: a) $l'(s, t) \geq k$; b) *if* $l'(s, t) = k$, *then* $S' \subset S$ (*as parts of the graph* Γ).

Let us consider the difference $\tilde{\Gamma}' \setminus \tilde{\Gamma}$ (in the graph Γ). By definition, $q \in Q(\tilde{\Gamma}' \setminus \tilde{\Gamma}) \Leftrightarrow q \in Q(\Gamma)$, $\delta(q) = 0$, $\delta'(q) > 0$. Since the quantity δ increases only on arcs opposite to the arcs of L, it can be asserted that $\tilde{\Gamma}' \setminus \tilde{\Gamma}$ consists only of arcs opposite to arcs of S. We shall prove that adjunction to the graph of one arc of such type does not change the referent; whence by induction it will follow that the referent of the graph $\tilde{\Gamma}' \cup \tilde{\Gamma}$ coincides with S.

Let us consider an arbitrary path from s to t in a new graph containing a new edge $[a, b]$. The segment of it from s to a consists only of old edges, and hence its length is not less than $l(s, a)$. Analogously the segment of the path from b to t is not shorter than $l(s, b)$. Hence the length of the path is not less than $l(s, a) + 1 + l(b, t) = l(s, a) + l(a, t) + 2 = k + 2$.

From the definition of referent the assertion follows.

In view of the inclusion $\tilde{\Gamma}' \subseteq \Gamma' \cup \Gamma$ we have the inclusion $S' \subseteq S$ if $l'(s, t) = l(s, t)$. It is strict, since in the graph $\tilde{\Gamma}'$ there will not be any of those arcs of L for which $\delta(q) = \min_{q \in L} \delta(q)$. The lemma is proved.

We shall show how to construct S' by successive diminution of S in case $l'(s, t) = l(s, t)$.

Obviously, it is necessary to throw out of S all arcs not lying in $\tilde{\Gamma}'$. In addition "dead ends" can appear in it, i.e. nodes from which do not leave, nor into which enter, any arcs. Such arcs cannot belong to the referent, hence it is necessary to discard them together with all the arcs incident to them. If "dead ends" again appear, then it is necessary to continue such a process until we obtain a part R of the graph Γ' without "dead ends". We note that it is possible to construct in it a path from s to t containing any arc or any node by the method described above for the referent. The length of such a path is equal to k, since it lies in S. Hence, if R is nonvoid, then $S' = R$. But if R is empty, then a path of length k from s to t does not exist, and $l'(s, t) \geq k + 1$. In such a case it is necessary to construct the referent S' anew.

Proof of convergence. We note that as a result of enlargement of the flow with the help of the shortest path from s to t either the number of arcs in the referent is decreased, or the distance from s to t is increased. Since the first is always finite, and the second does not exceed the number of nodes of the graph Γ, then we arrive in a finite number of steps at a flow $\bar{f}$ to which corresponds a graph Γ without paths from s to t. Hence, for any flow there are no paths enlarging it, and according to Ford and Fulkerson, it is maximal, which was required.

Estimate of the order of magnitude of the operations necessary for realization of the described algorithm.

The principal part of this magnitude Q is comprised by the operations necessary for search of the paths and for alteration of the function δ. At each step toward this a series of n operations is required. The number of arcs in the referent is not greater than p, and the number of different distances from s to t is not greater than n. Hence we have a term of Q of the order of n^2p. Here, possibly, the estimate of the mean number of steps necessary for degeneration of one referent is somewhat excessive.

In the construction of one referent there is required, generally speaking, on the order of p operations. Hence there appears a term of Q of the order of np.

In a change of referent the number of operations necessary for alteration of the information relating to a deleted arc or vertex does not exceed a certain constant, which does not depend on n or p. Hence it follows in each referent that there will be produced not more than $C(n + p)$ of such operations, and the term arising therefrom has order np.

Thus Q is bounded above by the quantity Cn^2p, and in the integer case the operations which are superfluous by comparison with the Ford-Fulkerson algorithm do not number more than C_1np.

Under careful scrutiny of the algorithm it can be noted that asymptotically C is a magnitude not greater than the zeroth (decimal) order (the operations are considered as machine operations).

Modification of the referent. We designate as a large referent of a network Γ a graph $\tilde{\Gamma}$ for each node of which $l(s, v)$ and $l(v, t)$ are calculated and a "utility" of each of its arcs defined in the following sense. The arc $[a, b]$ is called efficient as an emanating arc, if $l(s, b) = l(s, a) + 1$; efficient as an entering arc, if $l(a, t) = l(b, t) + 1$, and completely efficient if both conditions are satisfied. It is easy to prove that the set of completely efficient arcs coincides with the set of arcs of the referent graph $\tilde{\Gamma}$ described above.

Alteration of a large referent at each step of the algorithm consists in: a) alteration of the graph $\tilde{\Gamma}$; b) revision of the functions $l(s, v)$ and $l(v, t)$ for the nodes which were above called dead ends; c) corresponding alteration of the efficiency of the arcs identified by them.

Thus a large referent of the netword Γ successfully replaces the referent of the graph $\tilde{\Gamma}$, and at no time in the process of the algorithm is the construction of it required again.

Moscow State University

Received 18/FEB/70

Central-Research Institute for Patern Information

BIBLIOGRAPHY

[1] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in networks*, Princeton Univ. Press, Princeton, N. J., 1962; Russian transl., Moscow, 1966. MR 28 #2917.

Translated by R. F. Rinehart