AN O($|V|^3$) ALGORITHM FOR FINDING MAXIMUM FLOWS IN NETWORKS

V.M. MALHOTRA, M. Pramodh KUMAR and S.N. MAHESHWARI
Computer Science Program, Indian Institute of Technology, Kanpur 208016, India

Received 12 January 1978; revised version received 5 May 1978

Maximum flow, flow network, flow potential, reference node, reference potential

1. Introduction

We consider a flow network to be a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of directed arcs (arcs). The set of nodes V includes two distinguished nodes s and t called the source and sink respectively, and with each arc (u, v) in E is associated a non-negative real number $c(u, v)$ called the capacity of arc (u, v). A function f that maps each arc (u, v) in E into a non-negative real number $f(u, v)$ is said to be a flow if $0 \leqslant f(u, v) \leqslant c(u, v)$ for all (u, v) in E, and

$$\sum_{(u,w)\in E} f(u, w) - \sum_{(v,u)\in E} f(v, u) = 0$$

for all u in $V - \{s, t\}$.

The value of the flow is equal to $\Sigma_{(s,v)\in E}\, f(s, v)$, i.e., the net flow out of s.

The problem of how to determine the maximum flow in a network starting from a feasible flow is well understood [5]. The best known algorithms follow Dinic [2] in breaking the augmentation process into stages. For a detailed exposition the reader is referred to Dinic [2] or Even and Tarjan [4]. For our purposes it suffices to say that the per-stage flow problem can be formulated as follows: Given an acyclic flow network $G = (V, E)$ in which all the paths from s to t have the same length, determine a flow f such that on any path from s to t there exists an arc (u, v) for which $f(u, v) = c(u, v)$.

Dinic's algorithm requires O($|V| \cdot |E|$) computations to obtain such a flow. Karzanov [7] improved the bound to O($|V|^2$). For a readable English version of Karzanov's algorithm readers are referred to Even [3]. We give another algorithm, which is conceptually simpler than Karzanov's, to solve the per-stage flow problem in O($|V|^2$) steps.

In what follows the flow network $G = (V, E)$ is always understood to be the per-stage flow network.

2. Proposed method

Consider the flow network $G = (V, E)$ with some flow f. With each v in V we associate a flow potential $\rho_f(v)$, given by

$$\rho_f(v) = \min\{ \sum_{(v,w)\in E} (c(v, w) - f(v, w)) ,$$

$$\sum_{(u,v)\in E} (c(u, v) - f(u, v))\} \qquad (v \neq s, v \neq t)$$

$$\rho_f(s) = \sum_{(s,w)\in E} (c(s, w) - f(s, w)) ,$$

and

$$\rho_f(t) = \sum_{(u,t)\in E} (c(u, t) - f(u, t)) .$$

Intuitively, the flow potential is the maximum extra flow that can be forced through a node. We call a node r in V the *reference node* and the associated flow potential the *reference potential*, if

$$\rho_f(r) = \min_{v \in V} \{\rho_f(v)\} .$$

The importance of the reference node follows from the following lemma.

Lemma. *Let r be the reference node in a flow network $G = (V, E)$ with flow f. Then the flow f can be augmented by $\rho_f(r)$ to result in a flow f' such that $\rho_{f'}(r) = 0$.*

Proof. We need only show this for the case when $r = s$; every other case is reducible to this one. Clearly, $\rho_f(s)$ amount of flow can be distributed among the outgoing arcs of s. The extra flow now reaching nodes that are a unit distance away from s can be distributed among their outgoing arcs in any order because $\rho_f(v) \geqslant \rho_f(s)$ for all v in V. For the same reason, the extra flow reaching nodes at distance i away from s is less than or equal to their flow potential, and so it can be distributed among their outgoing arcs.

The algorithm for the per stage flow problem is based on the lemma. At any iteration, the reference node is determined and, starting from the reference node flow equal to the reference potential, is forced both towards the source and the sink. The flow is pushed from the reference node to the sink by processing the nodes from the reference node to the sink in topological order. When processing a node, the incoming flow is routed through outgoing arcs, saturating them one-one, so that at most one outgoing arc has flow added to it but remains unsaturated. The process is similar for pushing flow backwards from the reference node to the source. All the saturated arcs can be deleted from the network, since flow in them will not be affected by later iterations. Likewise, nodes that have had either all their incoming or outgoing arcs deleted can be deleted. Deletion of a node causes the deletion of all incoming or outgoing arcs. Since at any iteration all the incoming or the outgoing arcs at the reference node will become saturated there can be at most $|V|$ iterations.

If information about incoming and outgoing arcs at a node is kept in linked lists then the amount of effort in distributing the flow during the ith iteration is $O(|V| + |E_i|)$, where $|E_i|$ is the number of arcs deleted and $|V|$ is the number of arcs receiving extra flow but not saturated (maximum of one per node). Therefore the total effort is of the order of

$$O(\sum_i (|V| + |E_i|)) = O(|V|^2 + |E|)$$
$$= O(|V|^2) .$$

It should be noted that recomputation of flow potential at each node can be done while nodes and arcs are being deleted; here it requires no extra effort. Identification of the reference node, however, implies finding the minimum of at most $|V|$ numbers per iteration. Therefore the total effort is bounded by $O(|V|^2)$. Since, while determining the maximum flow in a network there will be at most $|V| - 1$ such stages [2], it follows that the maximum flow can be obtained using the method in $O(|V|^3)$ steps.

Recently Cherkasky [1] and Galil [6] have reported maximum flow algorithms of complexity $O(|V|^2 \cdot |E|^{1/2})$ and $O(|V|^{5/3} \cdot |E|^{2/3})$ respectively. These algorithms are more complex and on dense graphs have the same complexity as the proposed algorithm.

References

[1] B.V. Cherkasky, Algorithm of construction of maximal flow in networks with complexity of $O(|V|^2 \cdot |E|^{1/2}$ operations, Math. Methods of Solution of Econ. Problems, 7 (1977) 117–125.

[2] E.A. Dinic, Algorithm for solution of a problem of maximum flow in a network with power estimation, Soviet Math. Dokl. 11 (1970) 1277–1280.

[3] S. Even, The max flow algorithm of Dinic and Karzanov: An exposition, MIT Laboratory for Computer Science Technical Report No. MIT/LCS/TM-80 (1976).

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

[5] L.R. Ford and D.R. Fulkerson, Flows in Networks (Princeton University Press, Princeton, NJ, 1962).

[6] Z. Galil, A new algorithm for maximal flow problem: preliminary version, Dept. of Mathematical Sciences, Tel-Aviv University, Tel-Aviv, Israel (1978).

[7] A.V. Karzanov, Determining the maximal flow in a network by the method of preflows, Soviet Math. Dokl. 15 (1974) 434–437.