MAXIMUM FLOW IN (s, t) PLANAR NETWORKS

Refael HASSIN
Department of Statistics, Tel-Aviv University, Tel-Aviv, Israel

Received 29 October 1980; revised version received 14 September 1981

It is well known that a minimum cut of an (s, t) planar network can be found by constructing a shortest (s′, t′) path in a dual network [1, p. 156]. Itai and Shiloach [2] mention that implementation of this method requires $O(n \log n)$ time, however it does not produce the flow function itself. They therefore develop another $O(n \log n)$ algorithm which also finds the flow function.

This note shows that a tree of shortest paths rooted at s′, which can be found in $O(n \log n)$ time, defines not only a minimum cut but also a complete (maximum) flow function.

Let N be a network consisting of a (directed or undirected) graph $G = (V, E)$ with a source s and terminal t, and a capacity function c defined on E. Let $G^d = (V', E')$ be a dual graph of G with s′ and t′ as its source and terminal, respectively.

Each edge (i, j) in G is associated with an edge (i′, j′) in G^d, where node i corresponds to node i′ and j corresponds to j′. The procedure for dualizing graphs, including the correspondence between the endpoints of the edges, is described in [3, p. 33]. This correspondence is used to define the direction on the edges of the dual of a directed graph.

Let N′ be the network consisting of G^d, and a length function d defined on E′, such that $d(i', j')$ is equal to the capacity $c(i, j)$ of the corresponding edge in E. Let u(v) be the length of a shortest path from s′ to v, for every $v \in V'$.

A maximum flow function f can be constructed as follows: For each edge $(i, j) \in E$, let $(i', j') \in E'$ be the dual edge associated with it. Then let $f(i, j) = u(j') - u(i')$. The proof follows from the following observations:

(1) For every $(i', j') \in E'$ on a shortest (s′, t′) path, $u(j') - u(i') = c(i, j)$. Therefore f saturates the minimum cuts of N.

(2) For every $(i', j') \in E'$ $u(j') - u(i') \leq c(i, j)$, therefore $f(e) \leq c(e)$ for every $e \in E$.

(3) For every cycle $C \in E'$, $\Sigma_{(i',j') \in C}(u(j') - u(i')) = 0$. Therefore for every $v \in V$, $\Sigma_{(v,w) \in E} f(v, w) - \Sigma_{(w,v) \in E} f(w, v) = 0$.

The algorithm developed by Itai and Shiloach [2] has two steps:

(1) Search for the 'uppermost' augmenting path.

(2) Modification of the capacities.

Step 2 is in fact, an implementation of a shortest path algorithm in G^d. Step 1 requires $O(\log n)$ time in each iteration and $O(n \log n)$ time altogether. As I have suggested, this step can be replaced by explicitly constructing G^d at the beginning of the computation. This may be done in linear time.

References

[1] T.C. Hu, Integer Programming and Network Flows (Addison-Wesley, Reading, MA, 1969).
[2] A. Itai and Y. Shiloach, Maximum flow in planar networks, SIAM J. Comput. 8 (1979) 135–150.
[3] E.L. Lawler, Combinatorial Optimization: Network and Matroids (Holt, Rinehart and Winston, New York, 1970).