ALGORITHMS FOR MAXIMUM NETWORK FLOW

Robert E. TARJAN
AT & T Bell Laboratories, Murray Hill, NJ 07974, USA

Received 15 June 1983
Revised manuscript received 3 February 1984

This paper is a survey, from the point of view of a theoretical computer scientist, of efficient algorithms for the maximum flow problem. Included is a discussion of the most efficient known algorithm for sparse graphs, which makes use of a novel data structure for representing rooted trees. Also discussed are the potential practical significance of the algorithms and open problems.

Key word: Maxflow.

1. Introduction

The maximum network flow problem is one of the classical problems of network optimization. From the point of view of the complexity theorist, it is also one of the most intriguing, because of the number, variety, and rich structure of the algorithms that have been proposed to solve it. This paper is a survey of the known maximum flow algorithms and the techniques they use. Our emphasis is on theoretical efficiency, as measured by the worst-case running time of an algorithm on a random-access computer. We shall generally use the 'unit-cost' measure: any operation on real numbers is assumed to take unit time. We shall occasionally use the 'logarithmic-cost' measure, under which any operation on real numbers takes time proportional to the number of bits of precision. For more information on our theoretical framework and in particular on these cost measures, see the books by Aho, Hopcroft and Ullman [1], Papadimitriou and Steiglitz [16], and Tarjan [24].

In Section 2 we define the maximum flow problem and discuss algorithms for solving it. The best methods are based on algorithms for the blocking flow problem, which we consider in Section 3. On sparse graphs, the most efficient known algorithm is due to Sleator and Tarjan [20, 21]. It uses a novel data structure for representing and manipulating rooted trees. Recently Gabow [9] has discovered a new maximum flow algorithm that combines a scaling technique with Dinic's algorithm. If the edge capacities are integers of moderate size, Gabow's method is competitive with that of Sleator and Tarjan. We discuss Gabow's method in Section 4, which also contains additional remarks, including a discussion of the potential practical significance of the theoretically fast algorithms and some open problems. Throughout the paper we omit many of the details and all of the proofs; these may be found in [24].
2. Maximum flow algorithms

Let $G = (V, E)$ be a directed graph with two distinguished vertices, a source $s$ and a sink $t$, and a positive real-valued capacity $c(v, w)$ on every directed edge $[v, w]$. If $[v, w]$ is not an edge, we define $c(v, w) = 0$. We denote the number of vertices by $n$ and the number of edges by $m$. For ease in stating time bounds we assume $n = O(m)$. A flow on $G$ is a real-valued function $f$ on the vertex pairs satisfying

(i) (skew symmetry) $f(v, w) = -f(w, v)$ for all $v, w$.

(ii) (capacity constraint) $f(v, w) \leq c(v, w)$ for all $v, w$.

(iii) (flow conservation) For every vertex $v$ other than $s$ and $t$,

$$\sum_w f(v, w) = 0.$$ 

We impose skew symmetry merely for technical convenience. Note that skew symmetry and the capacity constraint imply $f(v, w) = 0$ if neither $[v, w]$ nor $[w, v]$ is an edge. If $f(v, w) > 0$, we say there is a flow from $v$ to $w$ of magnitude $f(v, w)$. Flow conservation states that the total flow into any vertex other than $s$ and $t$ equals the total outgoing flow. The value of a flow $f$, denoted by $|f|$, is the net flow out of the source, $\sum_v f(s, v)$. The maximum flow problem is that of finding a flow of maximum value, called a maximum flow.

The classical theory of network flows was developed by Ford and Fulkerson [8]. To understand it, we need one more concept, that of a cut. A cut $X, \bar{X}$ is a partition of the vertex set into two parts, such that $s \in X$ and $t \in \bar{X}$. The capacity of the cut is

$$c(X, \bar{X}) = \sum_{v \in X, w \in \bar{X}} c(v, w).$$

The net flow across the cut is

$$f(X, \bar{X}) = \sum_{v \in X, w \in \bar{X}} f(v, w).$$

Flow conservation implies the following lemma:

Lemma 1. For any flow $f$, the net flow across any cut $X, \bar{X}$ equals the flow value.

Ford and Fulkerson's main result is the max-flow, min-cut theorem:

Theorem 1. A flow $f$ is maximum if and only if there is a cut $X, \bar{X}$ such that $|f| = c(X, \bar{X})$.

The capacity constraint and Lemma 1 imply that, for any flow $f$ and any cut $X, \bar{X}$,

$$|f| = \sum_{v \in X, w \in \bar{X}} f(v, w) \leq c(X, \bar{X}).$$

This gives the easy half of theorem 1. Ford and Fulkerson proved the converse by giving a method to increase the value of a flow if it is not maximum. For any flow