A CHARACTERIZATION OF THE MINIMUM CYCLE MEAN IN A DIGRAPH

by

Richard M. Karp

Memorandum No. UCB/ERL M77/47

28 June 1977

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A CHARACTERIZATION OF THE MINIMUM CYCLE MEAN IN A DIGRAPH*

by

Richard M. Karp

Computer Science Division
Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California at Berkeley

Let $G = (V,E)$ be a digraph with n vertices. Let f be a function from E into the real numbers, associating with each edge $e \in E$ a <u>weight</u> $f(e)$. Given any sequence of edges $\sigma = e_1, e_2, \ldots, e_p$ define $w(\sigma)$, the <u>weight</u> of σ, as $\sum_{i=1}^{p} f(e_i)$, and define $m(\sigma)$, the <u>mean weight</u> of σ, as $\frac{1}{p} w(\sigma,f)$. Let $\lambda^* = \min_{C} m(C)$ where C ranges over all directed cycles in G; λ^* is called the <u>minimum cycle mean</u>. We shall give a simple characterization of λ^*, as well as an algorithm for computing it efficiently.

If G is not strongly connected then we can find the minimum cycle mean by determining the minimum cycle mean for each strong component of G, and then taking the least of these. The strong components can be found in $O(n+|E|)$ computational steps ([6]). Henceforth we assume that G is strongly connected.

Let s be an arbitrarily chosen vertex. For every $v \in V$, and every nonnegative integer k, define $F_k(v)$ as the minimum weight of an edge progression of length k from s to v; if no such edge progression exists, then $F_k(v) = \infty$.

<u>Theorem 1</u>.

(1) $$\lambda^* = \min_{v \in V} \max_{0 \le k \le n-1} \left[\frac{F_n(v) - F_k(v)}{n-k} \right]$$

*Research supported by National Science Foundation Grant MCS74-17680-A02.

The proof requires a lemma.

Lemma 1. If $\lambda^* = 0$ then $\min_{v \in V} \max_{0 \leq k \leq n-1} \left[\frac{F_n(v) - F_k(v)}{n-k}\right] = 0$

Proof. Since $\lambda^* = 0$ there exists a cycle of weight zero, and there exists no cycle of negative weight. Because there are no negative cycles there is a minimum-weight edge progression from s to v, and its length is less than n. Let this minimum weight be $\pi(v)$. Then $F_n(v) \geq \pi(v)$. Also, $\pi(v) = \min_{1 \leq k \leq n-1} F_k(v)$, so

$$F_n(v) - \pi(v) = \max_{0 \leq k \leq n-1} [F_n(v) - F_k(v)] \geq 0 ,$$

and

(2) $$\max_{0 \leq k \leq n-1} \left[\frac{F_n(v) - F_k(v)}{n-k}\right] \geq 0 .$$

Equality holds in (2) if and only if $F_n(v) = \pi(v)$. Hence we can complete the proof by showing that there exists a v such that $F_n(v) = \pi(v)$. Let C be a cycle of weight zero, and let w be a vertex in C. Let $P(w)$ be a path of weight $\pi(w)$ from s to w. Then $P(w)$, followed by any number of repetitions of C, is also a minimum-weight edge progression from s to w. Hence, any initial part of such an edge progression must be a minimum-weight edge progression from s to its end point. After sufficiently many repetitions of C, such an initial part of length n will occur; let its end point be w'. Then $F_n(w') = \pi(w')$. Choosing $v = w'$, the proof is complete.

Proof of Theorem 1. We study the effect of reducing each edge weight $f(e)$ by a constant c. Clearly λ^* is reduced by c, $F_k(v)$ is reduced by kc, $\frac{F_n(v)-F_k(v)}{n-k}$ is reduced by c, and $\min_{v \in V} \max_{0 \leq k \leq n-1} \left[\frac{F_n(v)-F_k(v)}{n-k}\right]$ is reduced by c. Hence both sides of (1) are affected equally when the function f is translated by a constant. Choosing that translation which makes λ^* zero, and then applying Lemma 1, the proof is complete.

We can compute the quantities $F_k(v)$ by the recurrence

$$F_k(v) = \min_{(u,v) \in E} [F_{k-1}(u) + f(u,v)] , \quad k = 1,2,\ldots,n$$

with the initial conditions

$$F_0(s) = 0 \; ; \; F_0(v) = \infty , \quad v \neq s .$$

The computation requires $O(n|E|)$ operations, and, once the quantities $F_k(v)$ have been tabulated, we can compute $\lambda^* = \min_{v \in V} \max_{0 \leq k \leq n-1} \left[\frac{F_n(v)-F_k(v)}{n-k}\right]$ in $O(n^2)$ further operations. Since G is strongly connected $n \leq |E|$, so the over-all computation time is $O(n|E|)$. If the actual cycle yielding the minimum cycle mean is desired, it can be computed by selecting the minimizing v and k in (1), finding a minimum-weight edge progression of length n from s to v, and extracting a cycle of length $n-k$ occurring within that edge progression.

The minimum cycle mean problem is closely related to the negative cycle problem; i.e., the problem of deciding whether a digraph with weighted edges has a cycle of negative weight. The best algorithms known for solving the negative cycle problem require time $O(n|E|)$ ([2],[4]). The best algorithm previously known for computing the minimum cycle mean ([3]) makes $O(\log n)$ calls on a subroutine for solving the negative cycle problem, and hence has a running time of $O(n|E| \log n)$. Any algorithm for the minimum cycle mean problem yields a solution to the negative cycle problem quite simply: a negative cycle exists if and only if $\lambda^* < 0$. Thus any improvement on the $O(n|E|)$ running time of our minimum cycle mean algorithm would also give an improved upper bound on the computational complexity of the negative cycle problem.

References

[1] G.B. Dantzig, W. Blattner and M.R. Rao, "Finding a Cycle in a Graph with Minimum Cost to Time Ratio with Application to a Ship Routing Problem," in P. Rosenstiehl (ed.), Theory of Graphs, Dunod, Paris and Gordon and Breach, New York (1967), 77-84.

[2] D.B. Johnson, "Algorithms for Shortest Paths," Ph.D. Thesis, Cornell University, Ithaca, New York (1973).

[3] E.L. Lawler, "Optimal Cycles in Doubly Weighted Linear Graphs," in P. Rosenstiehl (ed.), Theory of Graphs, Dunod, Paris and Gordon and Breach, New York (1967), 209-214.

[4] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York (1976).

[5] L.V. Romanovskii, "Optimization of Stationary Control of a Discrete Deterministic Process," Cybernetics **3** (1967), 52-62.

[6] R.E. Tarjan, "Depth-First Search and Linear Graph Algorithms, SIAM J. Computing **1** (1972), 146-160.