

A NOTE ON THE OPTIMALITY OF SOME ALL-SHORTEST-PATH ALGORITHMS

MARIO NAKAMORI

The University of Tokyo

(Received January 18, 1972 and Revised February 16, 1972)

Abstract

It is proved that any algorithm that determines the shortest distances between all node pairs in a network by repeated applications of the so-called triple-operations should contain at least $n(n-1)(n-2)$ such operations, where n is the number of nodes in the network. Hence, Floyd's algorithm [2], as well as Dantzig's [1] and Katayama and Watanabe's [5], has been shown to be optimal.

1. The All-Shortest-Path Problem

Let us consider a network with n nodes. We denote by $N = \{1, 2, \dots, n\}$ the set of nodes, and by d_{ij} the distance of the branch connecting node i to node j , where we set $d_{ii} = 0$ for every $i \in N$, and $d_{ij} = \infty$ if there is no branch between node i and node j . Although some of d_{ij} 's may possibly be negative, the sum of d_{ij} 's around a closed path should always be nonnegative.

We shall call a matrix $D = (d_{ij})$ satisfying the above conditions a distance matrix. Especially, we shall call a distance matrix with all the entries nonnegative a nonnegative distance matrix.

We put

$$(1) \quad d_{ij}^{\infty} = \min \sum_{p=0}^{m-1} d_{i_p i_{p+1}}$$

and call the matrix $D^\infty = (d_{ij}^\infty)$ the shortest distance matrix corresponding to a distance matrix D , where the minimum in (1) is taken over all the sequences of the form $(i_0=i, i_1, i_2, \dots, i_{m-1}, i_m=j)$. It is obvious that the d_{ij}^∞ is the distance along the shortest path from node i to node j in the network N . The all-shortest-path problem is the problem of obtaining the minimum distance matrix $D^\infty = (d_{ij}^\infty)$ for an arbitrarily given distance matrix $D = (d_{ij})$.

2. Definition of Algorithms and Optimal Algorithms

Numerous methods have been proposed for the all-shortest-path problem. Most of them can be described in the form of repeated applications of the so-called triple-operations, where a triple-operation with pivot k_0 on pair (i_0, j_0) (to be denoted by $\langle \begin{smallmatrix} k_0 \\ i_0 \ j_0 \end{smallmatrix} \rangle$ in abbreviation) of a matrix $A = (a_{ij})$ transforms, by definition, A into $A' = (a'_{ij})$:

$$(2) \quad \begin{cases} a'_{ij} = a_{ij}, & \text{for } (i, j) \neq (i_0, j_0), \\ a'_{i_0, j_0} = \min(a_{i_0, j_0}, a_{i_0, k_0} + a_{k_0, j_0}). \end{cases}$$

In the following, the term "algorithm" will mean one such method. An algorithm is said to be "optimal" if it contains the smallest number of occurrences of triple-operations among all the "algorithms". Among the known algorithms, R.W. Floyd's [2], G.B. Dantzig's [1] and H. Katayama and H. Watanabe's [5] contain the fewest triple-operations, *i.e.* $n(n-1)(n-2)$ in number.

The aim of this paper is to prove that any algorithm necessarily contains at least $n(n-1)(n-2)$ triple-operations and, consequently, the above-mentioned three algorithms are optimal.

3. Fundamental Theorem

Let i_0, j_0, k_0 be any three distinct nodes of N . Then an all-shortest-path algorithm must contain the triple-operation $\langle \begin{smallmatrix} k_0 \\ i_0 \ j_0 \end{smallmatrix} \rangle$.

In fact, for the (nonnegative) distance matrix $D=(d_{ij})$ such that

$$d_{ij} = \begin{cases} 0, & \text{if } i = j \quad \text{or } (i, j) = (i_0, k_0) \\ & \text{or } (i, j) = (k_0, j_0), \\ \infty, & \text{otherwise,} \end{cases}$$

if the algorithm did not contain $\langle \begin{smallmatrix} k_0 \\ i_0 \ j_0 \end{smallmatrix} \rangle$, the resulting D^∞ would be equal to D , because any triple-operation $\langle \begin{smallmatrix} k \\ i \ j \end{smallmatrix} \rangle$ distinct from $\langle \begin{smallmatrix} k_0 \\ i_0 \ j_0 \end{smallmatrix} \rangle$ changes no entry of D (cf. (1)). However, this obviously contradicts the fact that the minimum distance matrix D^∞ for the distance matrix D now in consideration should have one more null entry $d_{i_0 j_0}^\infty$.

Therefore, any algorithm must contain each of the triple-operations $\langle \begin{smallmatrix} k \\ i \ j \end{smallmatrix} \rangle$ (i, j, k being distinct from one another) at least once, hence at least $n(n-1)(n-2)$ operations in all.

Incidentally, it should be noted that the theorem is valid for "nonnegative" distance matrices as well as for distance matrices.

4. Discussions

(a) We have excluded from consideration the so-called sequential methods, where which of the possible operations to adopt at the next step depends on the current outcome. (For example, Dijkstra's method for the shortest-distance problem from a fixed node is of this kind.)

(b) The proof presented here was found as a by-product of our study of the algebraic structures of the semi-groups of operators and path-sets in connection with the shortest-path problems. The detailed results of the study will be published elsewhere [4], where the known algorithms for networks of special structure, such as dealt with by T.C. Hu and W.T. Torres [3] and J.Y. Yen [6], will be improved to get an optimal algorithm.

References

- [1] Dantzig, G.B., "All Shortest Routes in a Graph," *Théorie des Graphes*, pp. 91-92, (Proceedings of the International Symposium, held in Rome, Italy, July 1966) published by Dunod, Paris.
- [2] Floyd, R.W., "Algorithm 97: Shortest Path," *Communications of the Association for Computing Machinery*, **5** (1962), 345.
- [3] Hu, T.C. and W.T. Torres, "Shortcut in the Decomposition Algorithm for Shortest Paths in a Network," *IBM Journal of Research and Development*, **13** (1969), 387-390.
- [4] Iri, M. and M. Nakamori, "Path-Sets, Operator Semigroups and Shortest-Path Algorithms on a Network," *RAAG Research Notes, Third Series*, No. 185 (October 1972).
- [5] Katayama, H. and H. Watanabe, "An algorithm for the Shortest Path Problem in a Network," *TGCT Papers of the Institute of Electronics and Communication Engineers of Japan*, CT 68-15 (June 1968).
- [6] Yen, J.Y., "On Hu's Decomposition Algorithm for Shortest Paths in a Network," *Journal of the Operations Research Society of America*, **19** (1971), 983-985.