Journal of the Operations Research
Society of Japan
Vol. 24, No. 3, September 1981

© 1981 The Operations Research Society of Japan

ROUTING A VEHICLE WITH THE LIMITATION OF FUEL

Tetsuo Ichimori
Osaka University

Hiroaki Ishii
Osaka University

Toshio Nishida
Osaka University

(Received January 6, 1981)

Abstract The paper considers the problem of routing a vehicle with the limitation of fuel. The optimal route is a shortest path along which a vehicle can pass through a network, visiting some refueling vertices on the way for fear of running out of fuel. An efficient algorithm for it is presented whose computational complexity is $0 (pn^2)$ where n is the number of vertices in a network and p is that of refueling vertices.

1. Introduction

This paper is concerned with the following routing problem. We wish to route a vehicle through a traffic network in which there are only certain vertices at which it can refuel completely. Suppose that a vehicle can move a specified distance without refueling. How can we determine a shortest path between the origin and the destination for a vehicle along which it can move without running out of fuel?

The straightforward strategy for this problem is to compute the 1st, 2nd, 3rd, etc., shortest paths from the origin to the destination until the desired path is obtained. In view of the computational efficiency, however, the strategy posed above is not recommended. In this paper we develop a polynomially bounded algorithm for finding an optimal route.

The organization of this paper is as follows. In Section 2 the routing problem is defined. In Section 3 two difficulties in solving the routing problem are illustrated. Then in Section 4 we develop an efficient algorithm for solving the routing problem. The time complexity of it is $O(pn^2)$ where n is the number of vertices in a network and $p \le n$ is that of refueling vertices.

2. Definitions

Consider a traffic network $N=(V,A)$, where V is the set of *vertices*, A is the set of directed *arcs*. With each arc (i,j) from vertex i to vertex j is associated its length a_{ij}, which is here restricted to positive value. When a *path* is referred to, we intend a directed path which is permitted to traverse vertices and arcs more than once. The length of a path is defined to be the sum of the lengths of the arcs along that path.

A vehicle starts at a specified origin s and arrives at a specified destination t, visiting some refueling vertices on the way. Suppose that the vehicle can move at most distance L, after filling it up with fuel. Suppose in addition that there are p refueling vertices, among which are the origin and the destination. Then the routing problem is the problem of determining a shortest path between the origin and the destination along which a vehicle can move without running out of fuel. In the sequel we assume for simplicity that such a shortest path always exists.

3. Two Difficulties of the Routing Problem

Before describing our algorithm, we will present the following two difficulties which we may encouter in the routing problem, but not in the conventional shortest-path problems.

The first difficulty is illustrated with the example shown in Fig. 3.1, where $L=10$, s, r and t are refueling vertices, and the number on each arc is its length. A vehicle with $L=10$ can go along the path s-v-r-v-t with the cycle v-r-v, but can not along the *simple* path s-v-t because of the limitation of fuel.

The above illustration implies that an optimal route may contain cycles in the routing problem, unlike the conventional shortest-path problems.

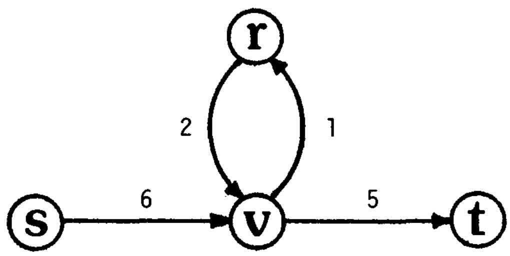

Fig. 3.1.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

The second difficulty is illustrated with the example shown in Fig. 3.2, where $L = 10$, s, r and t are refueling vertices, and the number of the arcs are their lengths. The optimal route between s and t is the path s-x-r-z-t. However the optimal route between s and z is not s-x-r-z, but s-x-y-z. The second illustration implies that the "Principle of Optimality", what is called, is not satisfied in this context. Here we note that these two difficulties are related to each other in some sense.

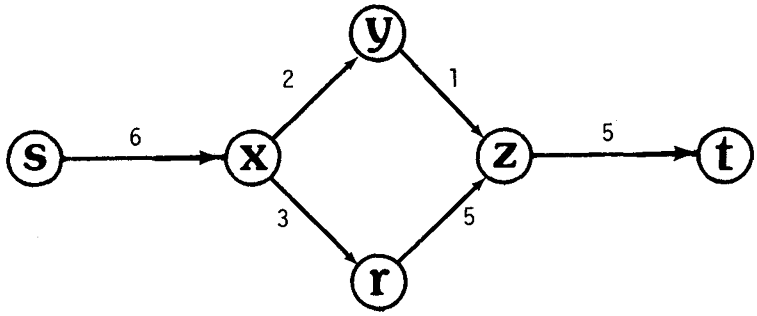

Fig. 3.2.

These two illustrations imply that it is impossible to solve the routing problem by applying one of the conventional shortest-path algorithms to it directly, i.e., by applying it only once.

4. A Solution Mehtod for the Routing Problem

Now we discuss a solution method for the routing problem. First consider the problem of finding shortest paths between all pairs of p refueling vertices in a traffic network $N = (V,A)$. This problem can be easily solved by applying the well-known Dijkstra's shortest-path algorithm [1] p times. Since the time complexity of the Dijkstra's algorithm is $O(n^2)$, the problem posed above is $O(pn^2)$, where $n=|V|$. Let u_{ij} denote the length of a shortest path from refueling vertex i to refueling vertex j.

Suppose $u_{ij} > L$ for some pair, i and j, for the moment. The value of u_{ij} corresponds to a shortest path P_{ij} from i to j. If P_{ij} does not contain any refueling vertices except for refueling vertices i and j, then the vehicle can not move from i to j, namely it must stop somewhere on the way from i to j. Otherwise, P_{ij} must contain at least one refueling vertex, say k. By "Principle of Optimality" we have

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

$$u_{ij} = u_{ik} + u_{kj} ,$$

i.e., the path P_{ij} consists of two shortest paths P_{ik} and P_{kj}.

From what is said above, all the values u_{ij} such that $u_{ij} > L$ can be discarded. After discarding $u_{ij} > L$, all the remaining u_{ij} are $\leq L$, which means that the vehicle can traverse any path without stopping which corresponds to $u_{ij} \leq L$.

Let us construct the following reduced network $N' = (V', A')$:
If each u_{ij} remains, then a directed arc $(i,j) \in A'$ and vertices i, $j \in V'$. Here we note that s, $t \in V'$ by assumption. In the network N' formed above, assign to each arc (i,j) its length $u_{ij} \leq L$. Then it is clear that the 1st shortest path from s to t in N' has its corresponding shortest path from s to t in the original network N and that the lengths of both shortest paths are equal. Since the 1st shortest path in N' contains only the arcs of lenths $\leq L$, its corresponding shortest path in N contains only the subpaths such that the lengths between successive refueling vertices on them are $\leq L$. Accordingly the vehicle can go through the network N along this path, which corresponds to the 1st shortest path in N'. The computation time for the 1st shortest path in N' is $O(p^2)$, noticing $|V'| = p$.

Algorithm 1 and time complexity

(1) Compute u_{ij} for all pairs of p refueling vertices. --- $O(pn^2)$.

(2) Discard u_{ij} if $u_{ij} > L$ and construct N'. --- $O(p^2)$.

(3) Compute the 1st shortest path in N'. --- $O(p^2)$. (This path corresponds to the desired one in N.)

Remark 1. The theoretical time bound for the entire algorithm is therefore proportional to pn^2.

Remark 2. Although the actual construction of the optimal route in N is not explicitly described here, such an issue is relatively straightforward and hence omitted.

Reference

[1] Dijkstra, E. W.: A Note on Two Problems in Connection with Graphs. *Numerishe Mathematik*, Vol. 1 (1959), 269.

Tetsuo ICHIMORI: Department of Applied Physics, Faculty of Engineering, Osaka University, Yamada-Oka, Suita, Osaka 565, Japan.

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.