Discrete Applied Mathematics 6 (1983) 85-89

North-Holland Publishing Company

NOTE

TWO ROUTING PROBLEMS WITH THE LIMITATION OF FUEL

Tetsuo ICHIMORI, Hiroaki ISHII and Toshio NISHIDA

Department of Applied Physics, Osaka University, Suita, Osaka 565, Japan

Received 22 September 1981

Revised 12 February 1982

Two routing problems are considered. Although these two are related to each other, one is polynomially solvable and, by contrast, the other is NP-complete. First an efficient solution procedure is developed for the polynomially solvable problem. Then we establish NP-completeness of the other problem.

Introduction

Consider a traffic network $N=(V,A)$ where V is the set of vertices, A is the set of undirected arcs. Assume N is connected. A length $a_{ij}>0$ is associated with each arc (i,j) between vertices i and j. There are p refueling vertices, including a depot. The others are referred to as non-refueling vertices. A vehicle starts from the depot and can move at most distance L after filling up with fuel.

On receiving a service call from any vertex, it travels from the depot to the vertex and returns to the depot. During this trip, the vehicle may visit some refueling vertices for fear of running out of fuel, even though the route may not be shortest.

The problem considered here, denoted by (P1), is to determine the minimum value of L, subject to the constraint that the vehicle can travel to any vertex of N and return to the depot without running out of fuel on the way. Or equivalently,

(P1) $\qquad \min L$

subject to the following constraints:

(i) each non-refueling vertex is connected to at least one refueling vertex through a path of length $\le \frac{1}{2}L$;

(ii) each refueling vertex is reachable from the depot along a path on which the lengths between successive refueling vertices are $\le L$.

There is a routing problem related to (P1), see [4]. The objective there is to find a shortest path between two specified refueling vertices along which the vehicle can travel without running out of fuel, when the value of L is fixed.

In Section 2 an efficient algorithm is developed for problem (P1) and the computational complexity is estimated. In Section 3 another routing problem (or

0166-218X/83/$03.00

location of refueling vertices) is considered and its NP-completeness is established. Finally a simple numerical example of problem (P1) is worked out in Section 4.

2. Solution procedure

Let p refueling vertices be $x_1, x_2, \ldots, x_p$, let x_1 be the depot, and let the other vertices, non-refueling vertices, be $y_1, y_2, \ldots, y_q$ where $q = |V| - p$.

First, consider the problem of finding shortest paths between all pairs of p refueling vertices in $N = (V, A)$. This problem can be easily solved by applying Dijkstra's shortest path algorithm [2] p times. Since the computational complexity of Dijkstra's algorithm is $O(n^2)$, the problem posed here is solvable in time $O(pn^2)$, where $n = |V|$.

Let u_{ij} denote the length of the shortest path between refueling vertices x_i and x_j which has been found above. Construct the reduced network $N' = (V', A)$ which contains an undirected arc (i, j) with length u_{ij} in A' and vertices i and j in V'.

Secondly, consider the problem of finding a minimum (or shortest) spanning tree of the reduced network N'. This problem can be solved in time $O(p^2)$ by implementing Prim–Dijkstra's minimum spanning tree algorithm [1, p. 138].

Let T be the resulting minimum spanning tree. And let $\max_{(i,j) \in T} u_{ij} = u^*$ and let L^* denote the minimum value of L in (P1).

Lemma 1. *L^* is greater than or equal to u^*.*

Proof. Obviously, we can see that the shortest path of N corresponding to each arc contained in the minimum spanning tree T of N' does not contain any refueling vertex except the endpoints of the path. Since the vehicle must traverse the shortest path of N without refueling which corresponds to the longest arc with length u^* in T and since a minimax (or bottleneck) spanning tree is exactly a minimum spanning tree (e.g. see [1, p. 139]), the conclusion follows. □

Now, consider the problem of finding a path between each non-refueling vertex y_k and its nearest refueling vertex x_k' in the original network N, letting v_k denote the path length. This problem is also solved by implementing Dijkstra's shortest path algorithm path algorithm q times, hence in time $O(qn^2)$.

Let $\max_{k=1,2,\ldots,q} v_k = v^*$. Then apparently L^* must be greater than or equal to $2v^*$. Hence we have the following important fact.

Theorem 2. $L^* = \max(u^*, 2v^*)$.

From what we have mentioned above, we can present the following algorithm.

Algorithm 1. (1) Compute u_{ij} for all pairs of p refueling vertices.
(2) Construct the reduced network N'.
(3) Compute the minimum spanning tree T.
(4) Set $u^* = \max_{(i,j) \in T} u_{ij}$.
(5) Compute v_k for q non-refueling vertices.
(6) Set $v^* = \max_{k=1,2,\ldots,q} v_k$.
(7) Set $L^* = \max(u^*, 2v^*)$.

Remark. The theoretical time bound for the entire algorithm is $O(pn^2) + O(qn^2) = O(n^3)$. Although the actual construction of routes for L^* in N is not explicitly described here, such an issue is relatively straightforward and hence omitted. (See Proposition 3 below.)

Finally we conclude this section by presenting the following proposition whose proof is omitted.

Proposition 3. *There exists a spanning tree of N such that the routes in the tree give* L^*. (See the result of the numerical example in Section 4.)

Remark. The routes identified by a spanning tree of N are not shortest paths in general (but feasible for L^*) from the depot to the vertices. Such shortest paths may constitute cycles (see [4] for more details).

3. Another routing problem

In the preceding sections, the number p of refueling vertices is fixed and L is a variable. Here we fix the value of L and minimize the number of refueling vertices, i.e., p is a variable. (In addition, the routing of the vehicle and location of the refueling vertices are desired to be found if possible. However, the following theorem strongly suggests that this will be intractale.) We will show that the following problem, denoted by (P2), is NP-complete. (P2) is as follows:

(P2) $\quad \min p$

subject to the same constraints as (P1).

Theorem 4. (P2) *is NP-complete.*

Proof. We will show that 'Vertex Cover' reduces to (P2). Note that Vertex Cover is NP-complete (see [3]).

Let G be an undirected connected graph in which each arc length is unity. let $d(i,j)$ denote the shortest path length between vertex i and vertex j as before. Let $C_1, C_2, \ldots, C_p$ be covering vertices, $D_1, D_2, \ldots, D_{n-p}$ be the others. Then Vertex

Cover can be written as follows:

(VC) min p

subject to the constraint that there is at least one covering vertex C_k' such that

$$d(D_k, C_k') = 1 \quad \text{for each } D_k.$$

Here it should be noted that the following fact holds in Vertex Cover: For each covering vertex C_i, there is at least one covering vertex C_j such that $d(C_i, C_j) = 1$ or 2 on any path emanating from C_i. Hence, Vertex Cover remains as it is if the above fact is imposed on it as an additional constraint.

We construct an instance of (P2) by letting $N = G$ (i.e. $a_{ij} = 1$ for all arcs), $L = 2$. The instance is as follows:

min p

subject to constraints:

(i) $d(y_k, x_k') = 1$ for $k = 1, 2, \ldots, n-p$,

(ii) each refueling vertex is reachable from the depot along a path on which the lengths between successive refueling vertices are one or two.

It is obvious that the instance of (P2) started above is the same as Vertex Cover with the additional constraint imposed, which implies (P2) is NP-complete. □

4. Numerical example of (P1)

Consider the numerical example of (P1) shown in Fig. 1, where the number on each arc is the length. The reduced network N' is as shown in Fig. 2. Since the minimum spanning tree T of N' is $\{(1,2), (1,3), (3,4)\}$, $u^* = 5$. Since the nearest refueling vertices of y_1, y_2, y_3, y_4, y_5 and y_6 are respectively x_2, x_1, x_1, x_1, x_2 and x_4, $v^* = 2$. Hence $L^* = 5$. The routes for $L^* = 5$ are shown in Fig. 3. Here note that a spanning tree results.

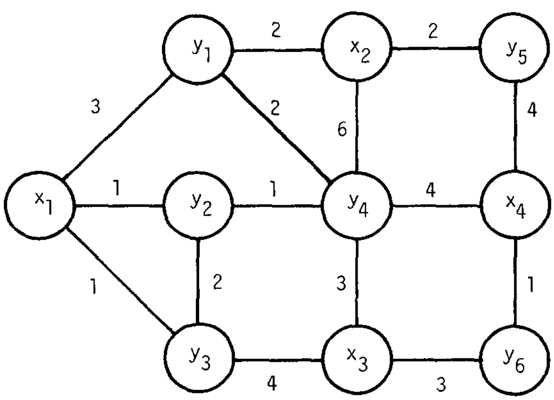

Fig. 1.

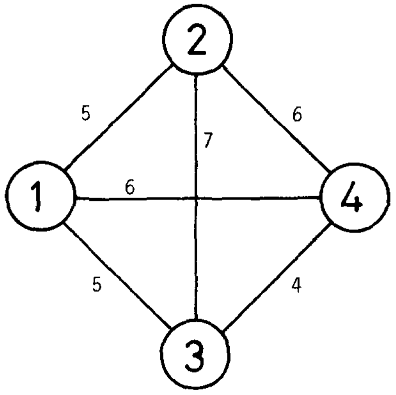

Fig. 2.

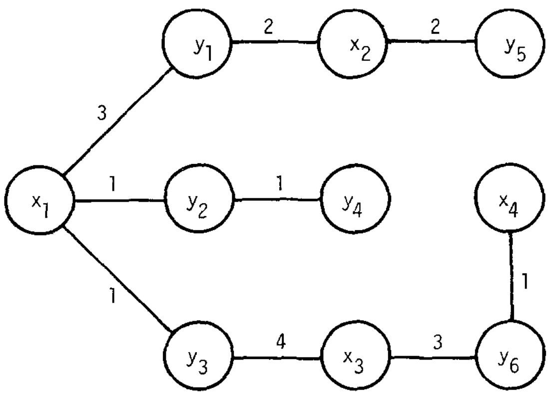

Fig. 3.

References

[1] N. Christofides, Graph Theory: An Algorithmic Approach (Academic Press, New York, 1975).

[2] E.W. Dijkstra, A note on two problems in connection with graphs, Numerische Mathematik 1 (1959) 269–271.

[3] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness (Freeman, San Francisco, CA, 1979).

[4] T. Ichimori, H. Ishii and T. Nishida, Routing a vehicle with the limitation of fuel, J. Operations Res. Japan 24 (1981) 277–281.