ELSEVIER

Operations Research Letters 16 (1994) 241–244

operations research letters

Solvable cases of the k-person Chinese postman problem

Wen Lea Pearn

Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan, ROC

(Received 1 August 1992; revised 1 July 1994)

Abstract

Given a network, the well-known Chinese Postman Problem (CPP) is to find a shortest postman tour traversing each arc of the network at least once and returning to the depot where the postman started. The CPP is NP-complete in general, but is polynomial-time solvable if the network is totally undirected, totally directed, mixed but even, windy with symmetric cycles, and windy but Eulerian. The k-person Chinese Postman Problem (k-CPP) is a multiple-vehicle extension of the CPP, which has many real-world applications. The intent of this paper is to generalize some of the above cited results to the k-CPP.

Key words: Chinese postman problem

1. Introduction

Given a network $G = (V, E)$ with V representing a set of nodes and E representing a set of arcs (streets), the celebrated Chinese Postman Problem (CPP) is that of finding a postman tour, which starts from the central depot (post office), traverses each arc in E at least once, then returns to the same depot node with the total distance traveled minimized. The CPP is originally proposed by Kwan [7]. Since its first appearance in *Chinese Mathematics*, it has been the focus of much research attention. As in the case of the Traveling Salesman Problem (TSP), the single-vehicle routing problems are rarely applicable in practical situations, but their multiple-vehicle variants, such as the Multiple Traveling Salesman Problem (MTSP), and the Vehicle Routing Problem (VRP), accommodate real-world situations more closely. Examples include the routing of solid waste collection vehicles, milk collecting trucks, the distribution of goods from warehouses to retail stores, and many others.

For the CPP we consider a multiple-vehicle extension, which we refer to as the k-person Chinese Postman Problem (k-CPP). The k-CPP may be briefly defined as follows. Given a network $G = (V, E)$, and a set of k postmen, then the k-CPP is to find a set of k postman tours so that: (1) each postman route begins and ends at a common node called the central depot, (2) each arc in G must be serviced by exactly one postman, (3) all the k postmen must be involved in the delivery service, and (4) the total distance traveled is minimized. Frederickson et al. [5] have considered another version of multi-vehicle extension of the CPP, which, may be called the *minimax k-CPP*. Instead of minimizing the total distance of the k tours, the minimax k-CPP seeks to minimize the length of the longest tour.

SSDI 0167-6377(94)00044-1

2. Polynomial-time solvable CPP

The CPP is NP-complete (see [6, 11]). But for certain cases of the CPP with suitable restrictions on network structure, polynomial-time algorithms have been developed to solve the problems exactly. These include the CPP on totally undirected networks [4], the CPP on totally directed networks [4, 2, 8], the CPP on mixed but even networks [4, 9], the CPP on windy networks with symmetric cycles [6], and the CPP on windy but Eulerian networks [15].

Case 1: CPP on totally undirected networks. For this class of CPP, Edmonds and Johnson [4] presented an efficient matching algorithm of $O(|V|^3)$ to solve the problem exactly. The algorithm first identifies the set of odd-degree nodes S in V, and constructs the matching network $N(S)$ consisting of S with distances defined as $d(i, j) =$ the shortest distance between i and j, for all $i, j \in S$. The algorithm then applies the minimal cost 1-matching algorithm over the matching network $N(S)$ generating a set of artificial arcs. Adding those artificial arcs to G, we obtain an optimal solution to the CPP.

Case 2: CPP on totally directed networks. For this class of CPP, several polynomial-time algorithms have been developed to solve the problem exactly. These include Edmonds and Johnson's algorithm [4], Beltrami and Bodin's algorithm [2], and Lin and Zhao's algorithm [8]. Edmonds and Johnson's algorithm transforms the original problem into a *minimal-cost flow* problem requiring an $O(|V|^3)$ of computations. Beltrami and Bodin's algorithm transforms the original problem into a standard *transportation problem* requiring an $O(|E| \cdot |V|^2)$ of computations. Lin and Zhao's algorithm is essentially based on the *Complementary Slackness* theorem in linear programming, requiring an $O(c \cdot |V|^2)$ of computations, where c is a constant depending on the network structure. All three algorithms are quite efficient and easy to implement.

Case 3: CPP on mixed but even networks. On mixed networks some arcs are undirected, and others are directed. For this class of CPP, Papadimitriou [11] showed it to be NP-complete. If the network is even (that is, every node has an even number of arcs incident to it), however, Edmonds and Johnson [4] provided a polynomial-time algorithm to solve the problem exactly. The algorithm basically transforms the original problem into a minimal-cost flow problem in which efficient solution procedures are available. The algorithm requires an $O(|V|^3)$ of computations.

Case 4: CPP on windy networks with symmetric cycles. On windy networks, an arc may be traversed in both directions with unequal distances. Guan [6] showed that this class of CPP is NP-complete in general, but is polynomial-time solvable if all the cycles in the network are symmetric. A cycle is symmetric if the length of the cycle in one direction equals that in the other direction. For this class of CPP, Guan [6] presented an efficient algorithm to solve the problem exactly. The algorithm simply transforms the problem into a standard CPP (Case 1) by redefining the distances as $d'(i, j) = d'(j, i) = (d(i, j) + d(j, i))/2$. The algorithm requires an $O(|V|^3)$ of computations.

Case 5: CPP on windy but Eulerian networks. On Eulerian networks every node is even, that is, every node has an even number of arcs incident to it. For this class of CPP, Win [15] presented a polynomial-time algorithm to solve the problem exactly. The algorithm first expands the original network by adding some new arcs with flow capacities, then applies the minimal-cost flow algorithm over the expanded network. The algorithm requires an $O(|V|^3)$ of computations.

3. Polynomial-time solvable k-CPP

The k-CPP is a multiple-vehicle extension of the CPP in which a set of k postman tours of minimal distance are to be found. Clearly, the CPP is a special case of the k-CPP. Since the CPP is NP-complete, the k-CPP must also be NP-complete. In Section 2, we reviewed several special cases of the CPP in which the problem may be solved exactly in polynomial time. In this section we generalize some of those results to the k-CPP.

Case 1': k-CPP on totally undirected networks. Benavent et al. [3] proposed a lower bound algorithm for the capacitated arc routing problem (CARP). The algorithm essentially applies the minimal-cost matching procedure [4] providing the following necessary conditions for the CARP solution: (1) the resulting network is even, (2) the number of maximal artificial paths between node i and the depot is no greater than the degree of node i (see [1]), and (3) at least $2M$ arcs are incident to the depot, where M = the minimal number of vehicles required to cover the service area. It is easy to show that Benavent et al.'s CARP lower bound algorithm solves this class of k-CPP optimally [12]. The complexity of this algorithm is of $O(|V|^3)$. We point out that the value of the k-CPP solution increases if the number of postmen, k, increases (see [1]).

Case 2′: k-CPP on totally directed networks. The algorithm for the *k-CPP on totally undirected networks* can be easily extended to the totally directed networks by recognizing that the number of directed maximal artificial paths from node i to the depot is no greater than $\deg^-(i)$, the degree into node i, and that the number of directed maximal artificial paths from the depot to node i is no greater than $\deg^+(i)$, the degree out of node i (see [12]).

Case 3′: k-CPP on mixed but even-and-symmetric networks. If G is even and symmetric, then G constitutes an optimal solution to the CPP. Let $G = G_u \cup G_d$, where G_u is the subnetwork of G consisting of undirected arcs and the corresponding nodes, and G_d is the subnetwork of G consisting of directed arcs and the corresponding nodes. Then, it is clear that G_u can be partitioned into a set of undirected cycles. Similarly, G_d can be partitioned into a set of directed cycles. Now, splice together all the cycles from G_u and G_d. The resulting big cycle is the desired postman route. To solve the k-CPP on this class of networks, we simply convert the original network into a totally directed one by assigning a direction to the subnetwork G_u following the postman route. Case 2′ now applies.

Case 4′: k-CPP on windy networks with symmetric cycles. On windy networks with symmetric cycles, the length of any cycle in one direction equals that in the other direction. Suppose we redefine the arc length as $d'(i, j) = d'(j, i) = (d(i,j) + d(j, i))/2$, and call the resulting network G' (which is totally undirected). Then, it is clear that the cycles in G are in one-to-one correspondence with the cycles in G'. Since all the cycles in G are symmetric, the length of any cycle in G must be equal to that of the corresponding cycle in G'. Therefore, the solution to the CPP on the windy network G (with symmetric cycles) must be the same as that to the CPP on G'. Further, both solutions have equal values. Consequently, the problem of solving the windy CPP on G (with symmetrical cycles) is equivalent to that of solving the CPP on G' [6]. The result can be easily extended to the k-CPP case. That is, the problem of solving the windy k-CPP on G (with symmetrical cycles) is equivalent to that of solving the k-CPP on G' (see [12]). And Case 1′ may now be applied.

On single-path networks, for the k-CPP to be polynomial-time solvable the networks have to be either totally undirected (Case 1′ applies) or windy. The solution techniques for both cases are the same. On single-cycle networks, for the k-CPP to be polynomial-time solvable the networks can be totally undirected (Case 1′ applies), totally directed (Case 2′ applies), mixed, or windy. The solution techniques for the mixed, and windy cases are trivial (if the solutions exist). On tree networks, for the k-CPP to be polynomial-time solvable the networks must be totally undirected (Case 1′ applies), or windy. Finding the solution for the windy case is, in fact, straightforward.

Acknowledgement

The author would like to thank the editor Dr. George L. Nemhauser, and anonymous referees for their helpful comments on an earlier draft.

References

[1] A. Assad, W.L. Pearn and B. Golden, "The capacitated Chinese postman problem: lower bounds and solvable cases", *Amer. J. Math. Management Sci.* 7(1/2), 63–88 (1987).

[2] E. Beltrami and L. Bodin, "Networks and vehicle routing for municipal waste collection", *Networks* **4**, 65–94 (1974).

[3] E. Benavent, V. Campos, A. Corberan and E. Mota, "The capacitated arc routing problem: lower bounds", *Networks* **22**, 669–690 (1992).

[4] J. Edmonds, and E. Johnson, "Matching, Euler tours, and the Chinese postman problem", *Math. Programming* **5**, 88–124 (1973).

[5] G. Frederickson, M. Hecht and C. Kim, "Approximation algorithms for some routing problems", *SIAM J. Comput.* **7**(2), 178–193 (1978).

[6] M. Guan, "On the windy postman problem", *Discrete Appl. Math.* **9**, 41–46 (1984).

[7] M. Kwan, "Graphic programming using odd or even points", *Chinese Math.* **1**, 273–277 (1962).

[8] Y. Lin, and Y. Zhao, "A new algorithm for the directed Chinese postman problem", *Comput. Oper. Res.* **15**(6), 577–584 (1988).

[9] E. Minieka, *Optimization Algorithms for Networks and Graphs*, Marcel Dekker, New York, 1978.

[10] E. Minieka, "The Chinese postman problem for mixed networks", *Management Sci.* **25**, 643–648 (1979).

[11] C. Papadimitriou, "On the complexity of edge traversing", *J. ACM* **23**(3), 544–554 (1976).

[12] W.L. Pearn, "On the k-person Chinese postman problem", Technical Report, National Chiao Tung University, Hsinchu, Taiwan, ROC 1991.

[13] W.L. Pearn and M.L. Li, "Algorithms for the windy postman problem", *Comput. Oper. Res.*, to appear.

[14] W.L. Pearn and G.M. Liu, "Algorithms for the Chinese postman problem on mixed networks", *Comput. Oper. Res.*, to appear.

[15] Z. Win, "On the windy postman problem on Eulerian graphs", *Math. Programming* **44**, 97–112 (1989).