



Combining ant colony optimization algorithm and dynamic programming technique for solving the covering salesman problem [☆]



Majid Salari ^{a,*}, Mohammad Reihaneh ^b, Mohammad S. Sabbagh ^c

^a Department of Industrial Engineering, Ferdowsi University of Mashhad, 91775-1111, Mashhad, Iran

^b Isenberg School of Management, 121 Presidents Drive, University of Massachusetts Amherst, MA 01003, United States

^c Department of Industrial and Systems Engineering, Isfahan University of Technology, 84156-83111, Isfahan, Iran

ARTICLE INFO

Article history:

Received 5 February 2014

Received in revised form 8 November 2014

Accepted 23 February 2015

Available online 5 March 2015

Keywords:

Traveling salesman problem

Covering salesman problem

Ant colony optimization

Dynamic programming

Heuristics

ABSTRACT

The covering salesman problem (CSP) is an extension of the well-known traveling salesman problem in which we are allowed to leave some vertices unvisited. The goal of the CSP is to construct a minimum length Hamiltonian cycle over a subset of vertices where those vertices not visited by the tour need to be within a pre-determined distance from at least one visited vertex. In this paper, we propose a mathematical formulation and a hybrid heuristic algorithm by combining ant colony optimization algorithm and dynamic programming technique to obtain high quality solutions. Comparing the results of the proposed algorithm with available methods in the literature clearly indicates the effectiveness of our proposed heuristic algorithm.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

The covering salesman problem (CSP) is a generalization of the traveling salesman problem (TSP) in which some vertices could remain unvisited (Current & Schilling, 1989). In particular, each vertex, say i , can cover a subset of vertices located within its pre-specified distance r_i . The goal of solving the problem is to construct a minimum length Hamiltonian cycle over a subset of vertices in which unvisited vertices need to be within the coverage radius of at least one visited vertex by the tour. The authors referred to potential applications of this problem arising in rural healthcare delivery problem. Essentially, the goal is to visit a given number of demand zones such that the people located in unvisited zones have to travel to their nearest visited zone to receive humanitarian functions (Current & Schilling, 1989).

Current and Schilling (1989) developed a simple heuristic algorithm for this problem. Their algorithm is composed of two steps. In the first step, a set covering problem (SCP) is solved to find the minimum number of vertices that cover all the vertices of the problem. In the second step, the optimal TSP tour is calculated

over the set of vertices obtained from first step. In case of having multiple optimal solutions for the SCP, all of them are used in the second step and the least costly TSP tour is selected.

Golden, Naji-Azimi, Raghavan, Salari, and Toth (2012) proposed two local search algorithms for the CSP. Their proposed algorithms take advantage of several moves like *swap*, *removal-reinsertion*, and *perturbation* to escape local optimal solutions. The first algorithm, denoted as *LS1*, improves the quality of the solutions by performing a set of removal and re-assignment moves. Essentially, during the first step of the *LS1* algorithm, some visited vertices are removed from the tour and in the second step, a new feasible solution is found by substituting the removed vertices with a better combination of unvisited vertices (if any). The second algorithm, denoted as *LS2*, applies more sophisticated moves to improve the solutions' quality. Particularly, *removal-reassignment* and *perturbation* procedures are applied to achieve this goal. Moreover, the *Lin-Kernighan* heuristic (Lin & Kernighan, 1973) is also used during this algorithm to find a better rearrangement of the visited vertices.

An integer linear programming (ILP) based heuristic algorithm was developed for the CSP (Salari & Naji-Azimi, 2012). Given an initial feasible solution, the algorithm follows a general destroy-and-repair paradigm to improve the tour length. To do this, some vertices are removed from the tour and reassigned, forming a new feasible solution resulting from solving an ILP-based model to optimality. The results show that ILP-based method is more efficient than other available CSP algorithms.

[☆] This manuscript was processed by Area Editor I-Lin Wang.

* Corresponding author at: Department of Industrial Engineering, Ferdowsi University of Mashhad, P.O. Box 91775-1111, Mashhad, Iran. Tel.: +98 0511 8805113.

E-mail addresses: msalari@um.ac.ir (M. Salari), mreihaneh@som.umass.edu (M. Reihaneh), sabbagh@cc.iut.ac.ir (M.S. Sabbagh).

Some researchers worked on a geometric variation of the CSP. In this version, each neighborhood is a compact set in the plane in a way that by intersecting this set, all the demands located at its neighborhood will be covered. Several approximation algorithms have also been proposed for this problem (see, e.g., Arkin & Hassin, 1994; Gulczynski, Heath, & Price, 2006 and Shuttleworth, Golden, Smith, & Wasil, 2008).

Several other variants of the CSP have been studied. The covering tour problem (CTP) was defined by Gendreau, Laporte, and Semet (1997), where the set of vertices is divided into three groups, namely those vertices that must be visited by the tour (S1), those that must be covered (S2) and vertices that can be visited or covered (S3). The other assumptions of the CTP are the same as those explained for the CSP. The goal of CTP is to cover all the vertices in S2 by constructing a tour over all the vertices in S1 and possibly a subset of vertices in S3. Gendreau et al. (1997) proposed a heuristic algorithm and an exact branch-and-cut method.

A multi-vehicle generalization of the CTP is introduced by Hachicha, Hodgson, Laporte, and Semet (2000), in which we are given a set of vehicles located at a central depot and, the goal is to visit or cover the demand of all available vertices at the minimum cost. In this problem, the classification of the vertices is the same as that defined for the CTP, while the only difference is that we have to cover the demand using more than one vehicle.

Golden et al. (2012) proposed some natural generalizations of the CSP, where the assumption of visiting or covering a vertex for only one time is relaxed. They referred to some applications arising in rural healthcare delivery, for which we have to visit or cover a vertex for more than once to satisfy its corresponding demand.

The ring star problem (RSP) is another related problem that has applications in telecommunications networks (Labbé, Laporte, Martín, & Salazar González, 2004). In this problem, each vertex is either visited by the tour (incurring a routing cost) or is assigned to the tour (incurring an assignment cost). The goal is to minimize the summation of routing and assignment cost. There is also a variation of the RSP called the median cycle problem, in which there is an upper bound on the maximum allocation cost (Labbé, Laporte, Martín, & Salazar González, 2005).

Another problem that is very closely related to CSP is the generalized traveling salesman problem (Fischetti, Salazar González, & Toth, 1997; Karapetyan & Reihaneh, 2012). In this problem, the set of vertices is partitioned into several disjoint clusters. The goal is to find the shortest Hamiltonian tour that visits exactly one vertex from each cluster. In fact, the generalized traveling salesman problem is a special case of the CSP, in which each vertex can be covered only by the vertices of the cluster to which it belongs.

In this paper, we propose a new and effective hybrid heuristic algorithm for the CSP. The proposed method combines Ant Colony Optimization (ACO) (Dorigo et al., 1992; Dorigo, Maniezzo, & Colomi, 1996) and dynamic programming technique to improve the solutions' quality. In addition, we propose a polynomial-size mathematical formulation for the studied problem.

The rest of the paper is organized as follows. Section 2 provides a formal description of the problem. The details of the hybrid heuristic algorithm are provided in Section 3. In Section 4, the efficiency of the proposed algorithm is investigated through extensive computational experiments. Finally, Section 5 concludes the paper.

2. Problem statement

In this section, we propose a polynomial-size mathematical formulation for the CSP. We are given a directed graph $G = (V, A)$ where $V = \{1, \dots, n\}$ is the set of vertices that must be either

visited or covered by the tour. In addition, $A = \{(i, j) | i, j \in V\}$ is the set of arcs. To each arc $(i, j) \in A$, a cost c_{ij} is associated. For each $i \in V$, we denote J_i as the set of vertices that can be covered by i and also J'_i as the set of vertices that can cover vertex i . Let $j^* = \arg \min_{i \in V} \{|J'_i|\}$. We create a copy i' of each $i \in J'_j$ and add it to the set of dummy vertices D . Then the augmented graph $G^* = (V^* = V \cup D, A^* = A \cup A')$ is constructed. Here, A' is the set of arcs between D and V , and contains the following arcs. For each dummy vertex $i' \in D$, we introduce only one arc (i', i) with $c_{i'i} = 0$ where i is the original version of i' (i' is a copy of i). Also, for each $i \in V$ and $j' \in D$ we add an arc (i, j') to A' with $c_{ij'} = c_{ij}$. The variables to model the problem are as follows:

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is traversed by the tour,} \\ 0 & \text{otherwise.} \end{cases}$$

$$y_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ is assigned to vertex } j, \\ 0 & \text{otherwise.} \end{cases}$$

For each arc $(i, j) \in A^*$ let u_{ij} be a continuous variable representing the load of the tour after leaving vertex i and before visiting vertex j when the tour passes along arc (i, j) . The model reads as follows:

$$\text{Minimize } \sum_{(i,j) \in A^*} c_{ij} x_{ij} \quad (1)$$

$$\text{subject to } \sum_{i \in D} y_{ii} = 1 \quad (2)$$

$$\sum_{j \in J'_i} y_{ij} = 1 \quad \forall i \in V \quad (3)$$

$$\sum_{(i,j) \in A^*} x_{ij} + \sum_{(j,i) \in A^*} x_{ji} = 2y_{ii} \quad \forall i \in V^* \quad (4)$$

$$\sum_{(i,j) \in A'} u_{ij} = ny_{ii} \quad \forall i \in D \quad (5)$$

$$\sum_{(j,i) \in A^*} u_{ji} - \sum_{(i,j) \in A^*} u_{ij} = \sum_{j \in J_i} y_{ji} \quad \forall i \in V \quad (6)$$

$$\sum_{i \in V} \sum_{j \in D} u_{ij} = 0 \quad (7)$$

$$u_{ij} \leq nx_{ij} \quad \forall (i, j) \in A^* \quad (8)$$

$$y_{ki} \geq y_{kj} - (1 - y_{ii}) \quad \forall k \in V, i, j \in J'_k: i < j \text{ and } i \neq j \neq k \quad (9)$$

$$x_{ij}, y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A^* \quad (10)$$

$$u_{ij} \geq 0 \quad \forall (i, j) \in A^* \quad (11)$$

The objective function (1) is to minimize the tour length. The first constraint (2) is used to assure that the salesman has to visit a subset of the vertices by starting from one of the dummy vertices. Constraint (3) ensures that each vertex is assigned to exactly one routed vertex. For each $i \in V^*$, constraint (4) represents the relation between routing (x) and allocation (y) variables. In particular, if vertex i is visited by the tour (i.e. $y_{ii} = 1$), there must be two edges incident to vertex i . Sub-tour elimination constraints are represented using constraints (5)–(7). Essentially, constraint (5) imposes that exactly n units of flow has to leave the dummy vertex that is visited by the tour. Constraint (6) stipulates that the load difference of the tour before and after visiting a vertex, say i , is exactly equal to the demand allocated to vertex i . Moreover, since the demand of all the n available customers has to be provided, the flow returned to the set of dummy vertices is equal to zero (given by (7)). The relation between the routing and flow variables is given in (8), in which we are not allowed to send flow along an unvisited arc. Constraints (9) are symmetry defeating constraints and are only introduced to enhance the model by breaking the symmetry caused by y . In fact, each vertex $k \in V$ can be covered by several vertices, and hence assigning k to any of them will create the same solution. This can

Download English Version:

<https://daneshyari.com/en/article/1133619>

Download Persian Version:

<https://daneshyari.com/article/1133619>

[Daneshyari.com](https://daneshyari.com)