TWMS J. App. Eng. Math. V.7, N.1, 2017, pp. 101-109

A NEW HEURISTIC ALGORITHM FOR MULTIPLE TRAVELING SALESMAN PROBLEM

F. NURIYEVA[1], G. KIZILATES[2], §

Abstract. The Multiple Traveling Salesman Problem (mTSP) is a combinatorial optimization problem in NP-hard class. The mTSP aims to acquire the minimum cost for traveling a given set of cities by assigning each of them to a different salesman in order to create m number of tours. This paper presents a new heuristic algorithm based on the shortest path algorithm to find a solution for the mTSP. The proposed method has been programmed in C language and its performance analysis has been carried out on the library instances. The computational results show the efficiency of this method.

Keywords: multiple traveling salesman problem, heuristic algorithms, shortest path algorithm, insertion heuristic, graph theory.

AMS Subject Classification: 90C27, 90C59, 90C09

1. Introduction

The Traveling Salesman Problem (TSP) is one of the most well-known and studied problems in the area of combinatorial optimization [1]. It aims to find the shortest tour or the tour with minimum cost for the given number of points, i.e., cities, pieces, vertices, etc., where all the distances between the points are known. Despite the simplicity of the definition of the TSP, its solution is quite difficult. As the number of points increases, the solution time and the difficulty of the problems increase correspondingly. Hence, instead of sweeping the whole solution space in order to guarantee the absolute solution, researchers mostly prefer to find quality solutions in short time periods by using heuristic methods.

The Multiple Traveling Salesman Problem (mTSP), on the other hand, is a derivation of the TSP for modelling daily life problems. In the mTSP, number of cities are divided into m number of tours by assigning each of these cities to a different salesman.

In this paper, the mTSP has been studied, a new heuristic algorithm has been proposed, and the results of the algorithm have been tested on the library instances. The rest of the paper will proceed as follows: The variants of the mTSP, the areas that use mTSP and the former studies on mTSP are mentioned in Section 2. The new algorithm that is proposed for the solution of mTSP is presented in Section 3. This proposed algorithm is tested on the sample library problems, and the efficiency of the method is discussed in Section 4. Finally, the complexity of the algorithm is given, and a goal for further remarks is indicated in the conclusion.

---

[1] Dokuz Eylul University, Faculty of Science, Department of Computer Science, Izmir-Turkey
e-mail: nuriyevafidan@gmail.com;

[2] Ege University, Faculty of Science, Department of Mathematics, Izmir-Turkey
e-mail: gozde.kizilates@gmail.com;

§ Manuscript received: April 11, 2016; accepted: June 09, 2016.
TWMS Journal of Applied and Engineering Mathematics, Vol.7, No.1;

2. Multiple Traveling Salesman Problem

The Multiple Traveling Salesman Problem (mTSP) is developed on the TSP. Unlike the TSP, there are m salesmen in the mTSP. In the mTSP, each city in a given set of n cities is divided into m tours by assigning each of these cities to a different salesman. The goal is to find the minimum cost of the tours in total. The cost can be defined as distance or time [2].

2.1. **Mathematical Model of the Problem.**

Many mathematical models have been proposed for the mTSP. In this paper, we will give some definitions before introducing the model. The mTSP is defined on a graph $G = (V, A)$, where V represents the set of vertices and A represents the set of edges. Let $C = (c_{ij})$ be the cost matrix defined on the set of A. If $c_{ij} = c_{ji}$ then the cost matrix is symmetric, otherwise it is asymmetric. If the cost matrix satisfies $c_{ij} \leq c_{ik} + c_{kj}$ for $\forall i, j, k$, then the matrix C satisfies the triangle inequality.

Among the proposed models for the mTSP in the literature, assignment based mathematical model, tree based mathematical model, and a three-index flow-based model have been mostly used.

The three-index flow-based model for the mTSP is as follows: Let n be the number of cities to be visited, and m be the number of salesmen (we assume $n \geq 3m + 1$). Then the variable x_{ij} is defined as follows:

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

Goal function:

$$minimize \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{1}$$

Constraints:

$$\sum_{j=2}^{n} x_{1j} = m \tag{2}$$

$$\sum_{j=2}^{n} x_{j1} = m \tag{3}$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 2, \ldots, n \tag{4}$$

$$\sum_{j=1}^{n} x_{ij} = 1, i = 2, \ldots, n \tag{5}$$

$$\sum_{i\in S} \sum_{j\in S} x_{ij} \leq |S| - 1, \forall S \subseteq V - 1, S \neq 0 \tag{6}$$

$$x_{ij} = 0 \vee 1, (i, j) \in A \tag{7}$$

In this model, constraints (4), (5), and (7) satisfy the assignment problem constraints. Constraints (2) and (3) ensure the return of each salesman to their starting point. Constraint (6) is used to prevent sub-tours [3].

2.2. **Problem Types and Application Areas.** There are different types of the mTSP. The possible variations of the problem are as follows [2]:

- Single and multiple depots: In single depot mTSP, every salesman begins his tour from the point with others and returns to this same point at the end of his tour. In multiple depot mTSP, on the other hand, every salesman is located on a different point, and at the end of his tour, he returns to the same point he started.
- Number of salesmen: The number of salesmen may be a bound variable or an initially fixed variable.
- Time window: In this variation, certain points must be visited in specific time periods. This variation of the mTSP is important and widely used. It is used for making the schedules of school shuttles, buses, and flights.
- Other special constraints: These are special constraints such as constraining the number of vertices that a salesman will visit or constraining the minimum or maximum length of the path that a salesman will use.

The mTSP can also be considered as relaxation of vehicle routing problem. Hence, the models and solution approaches proposed for vehicle routing problem may also be used for the mTSP [4].

The mTSP has a variety of application areas. These areas are basically the routing and scheduling problems. The following includes some of the application areas of the mTSP indicated in the literature [5, 6]:

- Computer electrical installation setup
- Path and route planning
- Job planning
- Robot control
- Drilling of printed circuit boards
- Chronological order

In this paper, single depot symmetrical mTSP is studied and a new algorithm is proposed.

2.3. **The Solution Methods for the Problem.** There have been many studies in solving the mTSP since 1970. These studies have generally done by evolutionary computation techniques.

Many methods that have been used for the mTSP focus on specifically vehicle routing problem [7].

Another significant method for solving the mTSP is to draw on the mathematical model of the problem. In mTSP studies, single and multiple depot variants are used, information about the solution algorithms is given, and new formulas that can be used to solve the problem are proposed [2].

3. A New Method for Single Depot mTSP

The method we proposed for single depot mTSP consists of two steps based on Shortest Path Algorithm and Insertion Heuristic. In the first step, algorithm creates tours as many as the number of salesmen. In the second step, the vertices that have not been used are inserted to the tours which have been created in the first step, through Insertion Heuristic technique.

3.1. **Shortest Path Algorithm.** Shortest path algorithm is used for finding the shortest path between two vertices on a given graph. There are a number of algorithms that are used for finding the shortest path between two vertices on a graph. The shortest path between two vertices (except an edge between these vertices) in a complete graph, is a path with 2-length. In our proposed algorithm, the length of paths is calculated by adding all vertices to the central vertex and the farthest vertex to it. Then the shortest length among these paths is selected as a shortest path. The complexity of this method is $O(n)$.

3.2. **Insertion Heuristic.** Insertion Heuristic techniques are easy to understand and have different variants. It simply starts with a tour created with a subset of given cities. Then new vertices are inserted to this tour by some heuristic methods. The initial sub tour is generally a triangle. Yet, some insertion heuristics start the algorithm with a single edge [5].

The insertion heuristic in proposed algorithm inserts the tours that have not been inserted yet into m tours including four vertices that have been created by the shortest path algorithm with an $O(n^2)$ complexity.

3.3. **The New Method.** Algorithm first starts with finding the central vertex and the farthest vertex from the central vertex. Central vertex can be found as follows:

$$x = \frac{\sum_{i=1}^{n} x_i}{n};\ y = \frac{\sum_{i=1}^{n} y_i}{n}$$

After finding (x, y) coordinates, we define the nearest vertex to these point as central vertex. Then the edge between these two vertices is removed from the graph. Hence, the edge that may probably cause a high cost in a tour is removed from the graph before the tours are formed. Then the shortest paths between the central vertex and its farthest vertex are found, and all the vertices (except the central vertex and its farthest vertex) and edges used on that path are removed from the graph. In this new graph, the second shortest path between the central vertex and its furthest vertex are found, and the vertices and edges used on that path are also removed from the graph. Hence, a tour between the central vertex and its furthest vertex is formed. This process continues until each salesman is assigned a tour. After assigning tours to salesmen is accomplished, the vertices that have not been included in any tours yet are included in the proper tour by insertion heuristic method. Then the algorithm is ended.

The main steps of the algorithm are as follows:

(1) Find the farthest vertex to the central vertex. Remove the edge between the central vertex and its farthest vertex. Find the shortest path between these two vertices and remove all the vertices (except the central vertex and a farthest vertex from it) and edges on that path from the graph.
(2) Find the second shortest path between the central vertex and its farthest vertex. Then, remove all the vertices (except the central vertex) and edges on that path from the graph.
(3) Assign the tour that has been formed by these two shortest paths (the tour that begins and ends with the central vertex) to a salesman.
(4) If the number of salesmen to whom a tour has been assigned is less than m, then go to Step 1.
(5) Otherwise, if there are any unassigned vertices left, then insert them to the tours by insertion heuristic method.
(6) STOP.

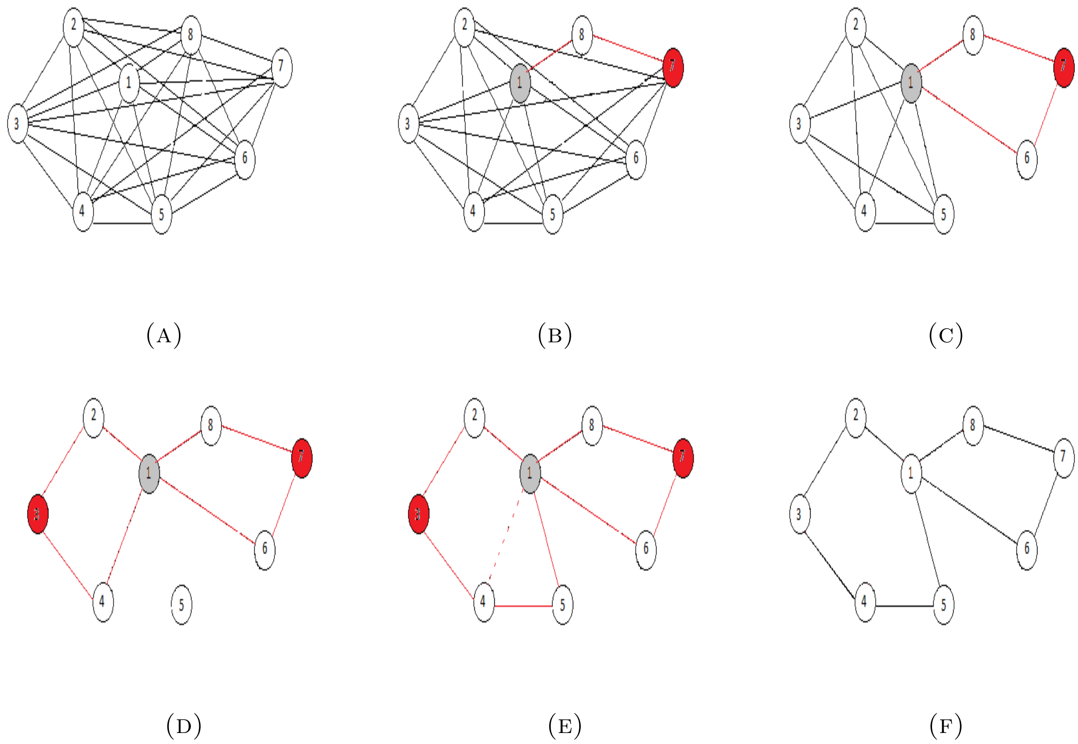

FIGURE 1. An example for the algorithm. (a) An example for the proposed algorithm. (b) Finding a central vertex and a farthest vertex to it. (c) A tour obtained by finding two shortest paths between the central vertex and farthest vertex to it (except an edge between them). (d) A tour obtained by finding two shortest paths between the central vertex and the second fathest vertex to it (except an edge between them). (e) Adding a vertices which are not included to the tour using insertion heuristic. (f) An obtained tour.

In Figure 1, these main steps of the algorithm are explained. In the example, vertex 1 is the central vertex, and the number of salesmen is two.

3.4. **The Time Complexity of the Proposed Method.** In the proposed algorithm, the shortest path algorithm is applied twice between the central vertex and m vertices far from the central vertex. Since the complexity of the shortest path algorithm is $O(n)$, the time complexity of forming m tours is $(2m * n)$ which is $O(n^2)$. Then, the algorithm inserts the residual vertices to the formed tours by insertion heuristic method. Since the complexity of insertion heuristic is $O(n^2)$, and insertion heuristic method will be used for maximum n times, the time complexity of Step 5 is $O(n^3)$.

The overall time complexity of the algorithm is $O(n^3)$.

4. Computational Results

In this section, some computational experiments have been done in order to evaluate the performance of the algorithm that is proposed for the mTSP. There is no library for the mTSP. Hence, the computational experiments for evaluating the proposed algorithm are done with the instances in TSPLIB that are created for the symmetric TSP [8].

TABLE 1. Computational Results

		Eil51	St70	Eil76	Rat99	KroA100	KroB100	Eil101	KroB150	KroA200	A280	Lin318
	Dimension	51	70	76	99	100	100	101	150	200	280	318
	Optimum	426	675	538	1211	21282	22141	629	26130	29368	2579	42029
m=2	Result	530	792	643	1486	24911	25573	721	30121	34711	3290	54131
	Time (s)	0,015	0,036	0,026	0,138	0,063	0,086	0,076	0,328	0,516	1,299	2,236
m=3	Result	594	970	711	1817	29059	30167	733	34188	39116	3653	59496
	Time (s)	0,015	0,041	0,046	0,176	0,097	0,104	0,121	0,468	0,703	2,015	3,087
m=4	Result	678	1072	766	2066	33739	34751	775	37673	43743	3948	68272
	Time (s)	0,031	0,047	0,06	0,265	0,134	0,159	0,144	0,562	0,953	2,343	4,809
m=5	Result	708	1248	873	2433	38523	40646	859	42970	48510	4494	75182
	Time (s)	0,031	0,062	0,093	0,281	0,188	0,169	0,182	0,734	1,187	2,953	5,525
m=6	Result	763	1368	929	2846	43501	46645	925	47606	53226	4954	81691
	Time (s)	0,031	0,125	0,094	0,343	0,23	0,193	0,25	0,843	1,359	3,547	6,082
m=7	Result	841	1539	975	3257	47979	52219	1013	51572	58448	5486	89194
	Time (s)	0,046	0,14	0,098	0,375	0,26	0,213	0,296	1,078	1,75	4,25	7,017
m=8	Result	890	1723	1065	3663	52857	58010	1095	56556	63565	6052	96468
	Time (s)	0,062	0,164	0,109	0,391	0,272	0,281	0,301	1,141	1,828	4,64	9,282
m=9	Result	979	1878	1146	4066	57854	63846	1176	61193	68356	6523	104737
	Time (s)	0,093	0,187	0,156	0,421	0,328	0,326	0,329	1,218	2	5,015	9,469
m=10	Result	1021	2058	1236	4468	62604	69645	1247	65200	73473	7071	111732
	Time (s)	0,078	0,141	0,161	0,453	0,343	0,359	0,331	1,406	2,379	5,826	10,933

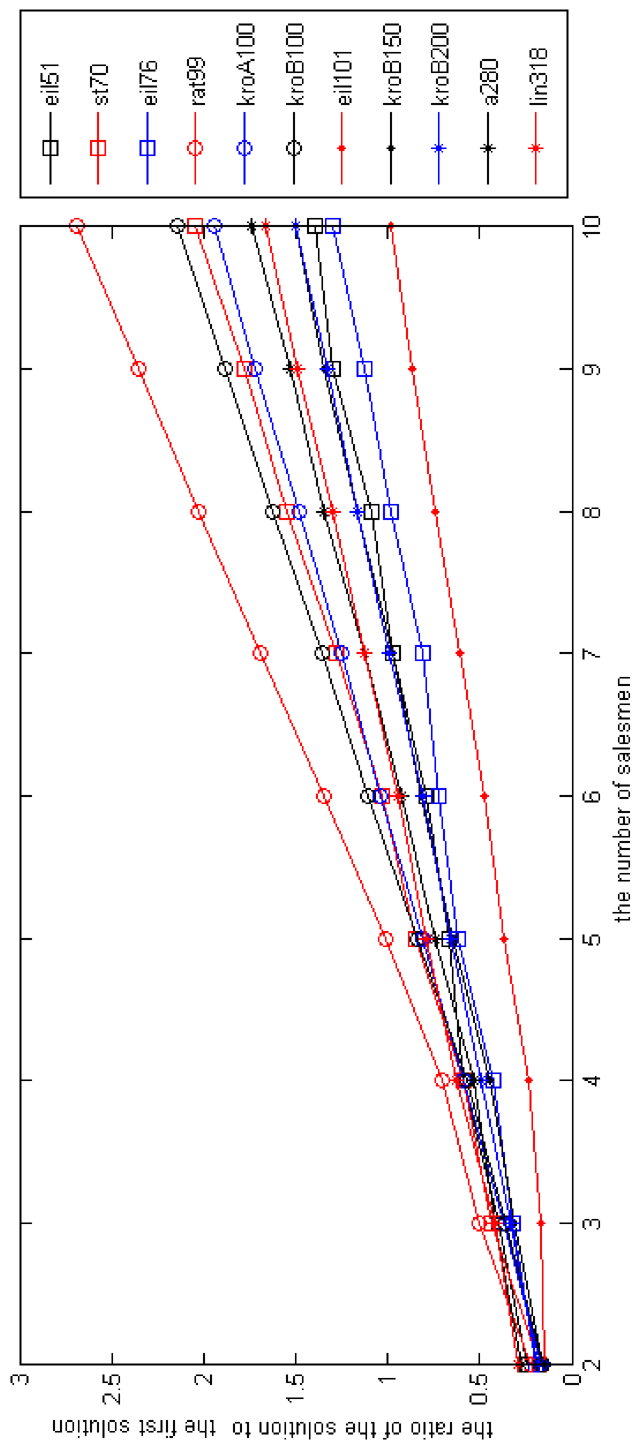

FIGURE 2. The performance of the computational results

The proposed algorithm has been programmed in C language. The processing time of the algorithm does not include the time for reading the graph and creating distance matrix.

The Table 1 includes the results and processing times of the program for each library instance in which the number of salesman, m, ranges from 2 to 10. The optimum row in the table shows the optimum results for the symmetric TSP in the TSPLIB. Furthermore, the change in the results of the problem depending on the increase of the number of salesmen is also shown in Figure 2.

5. Conclusion

The TSP is an optimization problem, which is seen in almost every area of the real life. As being one of the most well-known and common graph problems, the TSP belongs to the class of hard problems. In this paper, the mTSP, which is a developed version of TSP, has been studied, and a new heuristic algorithm that takes the shortest path algorithm as a basis has been proposed. The time complexity of this proposed algorithm is $O(n^4)$.

This proposed algorithm has been programmed in C language, and its performance has been analyzed by the frequently used test problems in the literature. After the results that have been obtained from the tests are examined, it is seen that the algorithm is efficient and produces results in a short time period.

There are many different variants of mTSP in the literature. For further study, it is aimed to propose new methods for these problems.

References

[1] Lawler,E., Lenstra,J., Kan,A., and Shmoys,D., (1985), The traveling salesman problem: a guided tour of combinatorial optimization, Chichester, Wiley, 11, pp.201-209.

[2] Kara,I. and Bektas,T., (2006), Integer linear programming formulations of multiple salesman problems and its variations, European Journal of Operational Research, 174(3), pp.1449-1458.

[3] Dantzig,G., Fulkerson,D., and Johnson,S.M., (1954), Solution of a large-scale traveling salesman problem, Operational Research 2, pp. 393-410.

[4] Bektas,T., (2006), The multiple traveling salesman problem: an overview of formulations and solution procedures, Omega, 34(3), pp.209-219.

[5] Matai,R., Singh,P.S., and Mittal,L.M., (2010), Traveling salesman problem: An overview of applications, formulations, and solution approaches, traveling salesman problem, theory and applications, pp.1-24.

[6] Averbakh,I., Lebedev,V., and Tsurkov,V., (2008), Nash equilibria solutions in the competitive salesmen problem on a network, Applied and Computational Mathematics, An International Journal, 7(1), pp.54-65.

[7] Park,Y., (2001), A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines, Int. Journal of Productions Econ., 73(2), pp.175-188.

[8] Traveling salesman problem and related problems library doi: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95.

Fidan Nuriyeva graduated from Ege University and works as an assistant professor at Dokuz Eylul University, Faculty of Science, Department of Computer Science. Her research interests include discrete mathematics, combinatorial optimization, graph theory, and theory of algorithms.

Gozde Kizilates is a Ph.D. Student at Ege University, Faculty of Science, Department of Mathematics. Her research interests include graph theory, theory of algorithms, and bioinformatics.