An Effective Genetic Algorithm for Solving the Multiple Traveling Salesman Problem

Mohammad Sedighpour[a,*], Majid Yousefikhoshbakht[b], Narges Mahmoodi Darani[c]

[a] Instructor, Department of Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran

[b] Instructor, Young Researchers Club, Hamedan Branch, Islamic Azad University, Hamedan, Iran

[c] Instructor, Department of Mathematics, Malayer Branch, Islamic Azad University, Malayer, Iran

Received 22 January, 2011; Revised 18 February, 2011; Accepted 5 March, 2011

Abstract

The multiple traveling salesman problem (MTSP) involves scheduling m > 1 salesmen to visit a set of n > m nodes so that each node is visited exactly once. The objective is to minimize the total distance traveled by all the salesmen. The MTSP is an example of combinatorial optimization problems, and has a multiplicity of applications, mostly in the areas of routing and scheduling. In this paper, a modified hybrid metaheuristic algorithm called GA2OPT for solving the MTSP is proposed. In this algorithm, at the first stage, the MTSP is solved by the modified genetic Algorithm (GA) in each iteration, and, at the second stage, the 2-Opt local search algorithm is used for improving solutions for that iteration. The proposed algorithm was tested on a set of 6 benchmark instances from the TSPLIB and in all but four instances the best known solution was improved. For the rest instances, the quality of the produced solution deviates less than 0.01% from the best known solutions ever.

Keywords: Genetic algorithm; Multiple traveling salesman problem; NP-Hard problems; 2-Opt local search algorithm.

1. Introduction

The multiple traveling salesman problems (MTSP) is a generalization of the well-known traveling salesman problem (TSP) (Carter [6]), where more than one salesman can be used in the solution. Besides, it is an example of combinatorial optimization problems, and has a multiplicity of applications mostly in the areas of routing and scheduling such as the School Bus Routing Problem (Angel [1], Orloff [14]). and the Pickup and Delivery Problem (Christofides [7], Savelsbergh [16]). Therefore, finding an efficient algorithm for the MTSP is important and induces to improve the solution of any other complex routing problems. The MTSP can in general be defined as follows:

Given *n>1* nodes, let there are *m* salesmen located at a single depot node. The remaining nodes that are to be visited are called intermediate nodes. Then, the MTSP consists of finding tours for all *m* salesmen, who all start and end at the depot, such that each intermediate node is visited exactly once and the total cost of visiting all nodes is minimized. In the MTSP, the *n* nodes must be partitioned into *m* tours, with each tour resulting in a TSP for one salesman. The MTSP is more difficult than the TSP because it requires assigning nodes to each salesman, as well as the optimal ordering of the nodes within each salesman's tour (Shi [17]).

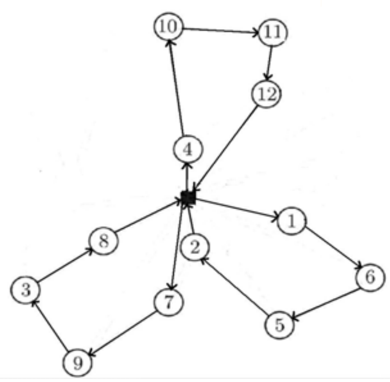

Fig. 1. A sample of solving the MTSP

*Corresponding author's E-mail: Sedighpour@iauh.ac.ir

The techniques used for solving the MTSP can be categorized into exact, heuristic and metaheuristic algorithms. Exact approaches for solving the MTSP are successfully used only for relatively small problems, but they guarantee optimality based on different techniques. These techniques apply algorithms that generate both a lower and an upper bound on the true minimum value of the problem instance. If the upper and lower bound coincide, a proof of optimality is achieved. There have been many studies in the literature proposing exact algorithms to solve the MTSP. These algorithms are based on lagrangean relaxation algorithm (Yadlapalli [19]), branch-and-cut method (Cordeau [8]), etc.

Although the MTSP is conceptually simple, it is difficult to obtain an optimal solution (Bektas [3]). In other words, when the problem size is increased, the exact methods cannot solve it. So, heuristic or metaheuristic methods are necessary to be used for solving them in a reasonable amount of time particularly with large sizes. Some of the well-known heuristic algorithms are gravitational emulation search (Balachandar [2]), local search (Bianchi [4]), and lin-kernighan (Karapetyan [12]).

A new kind of emerged algorithm basically tries to combine basic heuristic methods in higher level frameworks aimed at efficient and effective exploration of a search space in the last 30 years. These methods are nowadays commonly called *metaheuristics*. The term *metaheuristic*, first introduced in (Glover [10]), derives from the composition of two Greek words. *Heuristic* stems from the verb *heuriskein* which means "to find", while the prefix meta means "beyond in an upper level". Before this term was widely adopted, metaheuristics were often called *modern heuristics* (Reeves [15]). In general, it is incredibly urgent to use metaheuristic algorithms to solve complex optimization problems when dealing with them. Since the metaheuristic approaches are very efficient for escaping from local optimum, they are one of the best group algorithms for solving combinatorial optimization problem. That is why the recent publications are all based on metaheuristic approaches such as genetic algorithm (GA) (Kaur [13]), memetic algorithm (MA) (Bontoux [5]), ant system (AS) (Ghafurian [9]) and particle Swarm optimization (PSO) (Zhong [20]).

While the TSP is considered to be one of the standard problems in the operation research and management science literature, the MTSP has not yet received extensive attention. So, in this paper, a modified GA is used for solving the MTSP. Furthermore, the 2-Opt local search is applied to increase performance of the proposed algorithm.

In the following parts of this paper, a mathematical model of MTSP is presented in Section 2. In Section 3, the basic GA and the proposed idea are especially explained. In Section 4, the proposed algorithm is compared with some of the other algorithms on standard problems. Finally in Section 5, the conclusions are presented.

2. A Mathematical Model

Let *G(V,A)* be a perfect undirected connected graph with a vertex set $V = \{0, 1, ..., n\}$ and an edge set $A = \{(i, j) : i, j \in V, i \neq j\}$. If the graph is not perfect, the lack of any edge is replaced with an edge that has an infinite size. For presenting the integer linear programming model for MTSP, the following variables are introduced:

n = the number of nodes for each instance.

m = the number of salesmen used for each instance.

C= the cost matrix on graph G is symmetric firstly, and it is true in triangle inequality secondly. It means that $c_{ij} = c_{ji}$ and $c_{ij} + c_{jk} \geq c_{ik}$ for each $(i, j, k = 1, 2, ..., n)$.

$$x_{ij} = \begin{cases} 1 & \textit{if the salesman travels directly for } i \textit{ to } j \\ 0 & \textit{otherwise.} \end{cases}$$

Hence, one of the common integer programming formulations for the MTSP can be written as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

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

$$\sum_{i=1}^{n} x_{ij} = m \qquad j = 1 \tag{3}$$

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

$$\sum_{j=1}^{n} x_{ij} = m \qquad i = 1 \tag{5}$$

$$\sum_{i \in S} \sum_{j \in N-S} x_{ij} \geq 1 \ (\phi \neq S \subset N = \{2, ..., n\}), |S| \geq 2 \tag{6}$$

$$\sum_{i \in N-S} \sum_{j \in S} x_{ij} \geq 1 \ (\phi \neq S \subset N = \{2, ..., n\}), |S| \geq 2 \tag{7}$$

$$x_{ij} \in \{0, 1\} \tag{8}$$

The objective function (1) minimizes the total distance traveled in a tour. Constraint sets (2) and (3) ensure that the salesmen arrive once at each node and m times at the depot. Constraint sets (4) and (5) ensure that the salesmen leave each node once and the depot m times. Constraint sets (6) and (7) are to avoid the presence of sub-tours for each salesman. Finally, Constraint set (8) defines binary conditions on the variables.

3. The Presented Algorithm

Metaheuristic algorithms such as memetic algorithms, simulated annealing, particle swarm optimization, tabu search and so on have been successfully applied to many difficult optimization problems including traveling salesman problem, vehicle routing problem, quadratic assignment problem and job-shop scheduling problem, etc. In this section, first the Genetic Algorithm (GA) is explained, and then the proposed algorithm is analyzed in more details.

The Genetic Algorithm

The GA is one of the oldest metaheuristic algorithms that have received much attention by researchers' worldwide. This algorithm is an adaptive searching procedure for solving combinatorial optimization problems based on the mechanics of natural genetics and natural selection.

The GA starts from a group of initial solutions called the initial population. Furthermore, a fitness function is used to evaluate the performance of the solutions. Each time two solutions, called parent solutions, are chosen from the population according to the selection probability which is proportional to their fitness value. Then, the two parent solutions crossover to produce two new solutions of the next generation. These new solutions will replace the old solutions if they have better fitness.

Then, a mutation operation is applied to the newly-generated solutions based on a mutation probability. Repeat selection, crossover, and mutation operations to produce more new solutions until the population size of the new generation is the same as that of the old one. The iteration then starts from the new population. Since better solutions have a larger probability to be selected for crossover and the new solutions produced carry the features of their parents, it is hoped that the new generation will be better than the old one. The procedure continues until the number of generations is reached to n or the solution quality cannot be easily improved.

Proposed Algorithm

In the proposed algorithm, only one kind of chromosome is commonly employed for solving the MTSP. Fig. 2 illustrates a method for representing solutions to a MTSP (where $n = 11$ and $m = 3$). This technique involves using a single chromosome of length $n + m$ and is referred to as the ''one chromosome'' technique. In this technique, the n nodes are represented by a permutation of the integers from 1 to n. This permutation is partitioned into m sub-tours by the insertion of m negative integers (from 1 to m) that represent the change from one salesman to the next. In the example illustrated in Fig. 2, the first salesman would visit nodes 1 and 9 (in that order), the second salesman would visit nodes 10, 3, 11, 5, 4 and 2 (in that order), and the third salesman would visit nodes 6, 7 and 8 (in that order).

1	9	10	3	11	5	4	2	6	7	8	2	6	3

Fig. 2. Representation of the one chromosome in the proposed algorithm

One of the best crossovers in terms of quality and speed is Order crossover yet. So, this method which is simple to implement has been considered here, and the modified crossover is proposed based on the Order crossover. In this crossover, a randomly chosen crossover point divides the parent strings in left and right substrings. The right substrings of the parents are selected. After selection of nodes the process is the same as the order crossover. The only difference is that instead of selecting random several positions in a parent tour, all the positions to the right of the randomly chosen crossover point are selected (Fig. 3). Clearly, this method allows only the generation of valid strings.

Moreover, two mutations are used in the proposed algorithm. These operators select randomly two points in the string, and it replaces together (Fig. 4-1) or reverses the substring between these two cut points (Fig. 4-2). Furthermore, the vast literature on metaheuristics indicates that a promising approach to obtaining high-quality solutions is to couple a local search such as 2-opt algorithm when attaining a better solution compared to the previous iterations. In fact, the probability of finding better solutions near a good solution is relatively high in this situation. The 2-opt heuristic tries to improve route by replacing its two non-adjacent edges by two other edges (Fig. 5). It should be noted that there are several routes for connecting nodes and producing the tour again, but a state that satisfies the problem's constraints is acceptable. So, this unique tour will be accepted only if, first, the above constraints are not violated and, second, the new tour produces a better value for the problem than the previous solution. The process is repeated until no further reduction of route length is possible.

2 3 1 6 5 4 | 2 4 1 3 4 2 5 6 | 3 3	⇨	2 * 1 * * 4 | 2 4 1 3 * * * 6 | 3 3	⇨	2 3 1 5 6 4 | 2 4 1 3 2 5 4 6 | 3 3
(1)		(2)		(3)

Fig. 3. a. Randomized selection number of genes in each chromosome
b. finding arrangement these genes in another chromosome
c. replacement these genes on based new arrangement

2 3 1 6 5 4 | 2 4 $\Rightarrow$ 2 5 1 6 3 4 | 2 4 (1)

2 5 6 1 3 4 | 2 4 (2)

Fig. 4. Two mutations used for a chromosome

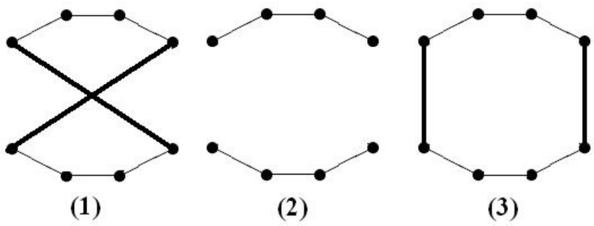

Fig. 5. The 2-opt algorithm

A pseudo-code of the proposed algorithm for the MTSP is presented in the Fig. 6 below.

procedure GA2OPT algorithm
S:= none; // *S is population of solutions* //
n= the number of nodes;
if n is even then nn:=n else nn:=n+1; // *the number of chromosomes* //
l:= int[n/10] ; // *l is the number of used mutations in each iteration* //
s^* *is the random solution;* // s^* *is the best solution found yet* //
f^* *is value of* s^* *;* // f^* *is the best value found yet* //
for i:= 1 to n do // *main cycle* //
for j:= 1 to nn do
Construct a solution s_i *as Fig. 2;*
Find value of the s_i *and call it* $f(s_i)$ *;*
S=S$\cup s_i$ *;*
end
for j := 1 to nn do
if j is odd then
begin
do crossover for s_j *and* s_{j+1} *as Fig. 3 and called them* s_i^* *and* s_{i+1}^* *;*
if values of s_j^* *and* s_{j+1}^* *are better than* s_j *and* s_{j+1} *respectively, then replaced them;*
end
end
end
select l number chromosomes from S and do mutation for them;
if new solutions based on mutations are better than before, then replace new solutions and their values;
find the best solution and value of S and called s_i^* *and* $f(s_i^*)$
if $f(s_i^*) < f^*$ *then*
begin
apply 2-opt local search to s_i^* *;*
$f^* := f(s_i^*)$ *;*
$s^* = s_i^*$*;*
end // *save the best so far solution* //
end
show s^* *and* f^*
end // *procedure* //

Fig. 6. Pseudo-code of the proposed algorithm

4. Experiments and Computational Results

Some numerical results of comparison between the proposed algorithm and several metaheuristic algorithms are presented in this section. The proposed algorithm is coded in Matlab and implemented on a Pentium 4 3 GHZ (512 MB RAM), operation system is windows XP. The proposed algorithm will be stopped after *n* iterations. To reveal the variability of the GA2OPT's performance from one run to another, 10 runs are carried out for each instance with different random numbers.

In these tests, the efficiency and performance of the proposed algorithm (PA) is compared with some of the best techniques designed including modified genetic Algorithm (MGA) [18] and Modified Ant Colony Algorithm (MACA) [11]. These algorithms are applied and tested on several instances from TSP problems available on the TSPLIB including Pr76, Pr152, Pr226, Pr299, Pr439 and Pr1002.

Table 1 illustrates the characteristics of the six problem instances pointed out above. These problems range in size from n=76 to n=1002 nodes. All problems are Euclidean, and their distances are compared with real numbers. Columns 2-6 show the problem size *n*, the number of salesmen *m*, the max number of customers that a salesman can visit *l*, the number of runs that carried out for each instance *tb*, the number of iterations that the proposed algorithm will be stopped after no improvement *T*. Additionally, in order to recognize the performance of the method, the best solutions published in the literature and also on the web *BKS*, are presented in column 7.

Table 1
The characteristics of the six problem instances

BKS	T	tb	l	M	n	Instance
157444	76	10	20	5	76	Pr76
127839	152	10	40	5	152	Pr152
166827	226	10	50	5	226	Pr226
82106	299	10	70	5	299	Pr299
161955	439	10	100	5	439	Pr439
382198	1002	10	220	5	1002	Pr1002

Table 2 shows the comparison of the proposed algorithm with the published results. The first column depicts the various instances, whereas columns 2-3 specify the two well-known and best published results obtained using metaheuristic algorithms. Furthermore, column 4 refers to the best result of the proposed method for these instances; column 5 presents the best result published in the literature and also on the web for these instances and finally; column 6 shows the mean gap values. It is noted that the gap is defined as the percentage of deviation from the best known solution in the literature. In other words, the gap is equal to $100[c(s^{**})-c(s^{*})]/c(s^{*})$, where s^{**} is the best solution found by the algorithm for a given instance, and s^{*} is the overall best known solution for the same instance on the Web.

The results of the comparison show that the GA2OPT has the ability for escaping from local optimum points and find the best solutions for most of the instances. Besides, the proposed algorithm yields better solutions than the MGA and MACO for some of the instances. More specifically, the results of this comparison show that the proposed algorithm gains worse solutions than the MGA in Pr152, and it gains better solutions than the MGA in the other problems from Pr76 to Pr1002.

Table 2
Comparison of algorithms for standard problems of MTSP

Gap	BKS	GA2OPT	MACO[11]	MGA [18]	instance
+0.00	157444	157440	178597	157444	Pr76
-0.01	127839	127852	130953	127839	Pr152
+0.00	166827	166817	167646	166827	Pr226
+0.01	82106	82095	82106	82176	Pr299
-0.01	161955	162150	161955	173839	Pr439
+0.00	382198	382185	382198	427269	Pr1002

Furthermore, the results indicate that although the MACO gives a better solution than the proposed algorithm for one instance in the name of Pr439, this algorithm cannot gain optimal solutions for the others and yields less solution than the proposed algorithm. The Computational experiments also show that, in general, the proposed algorithm gives better results compared to other two algorithms including the MGA and the MACO algorithms in terms of quality of solution.

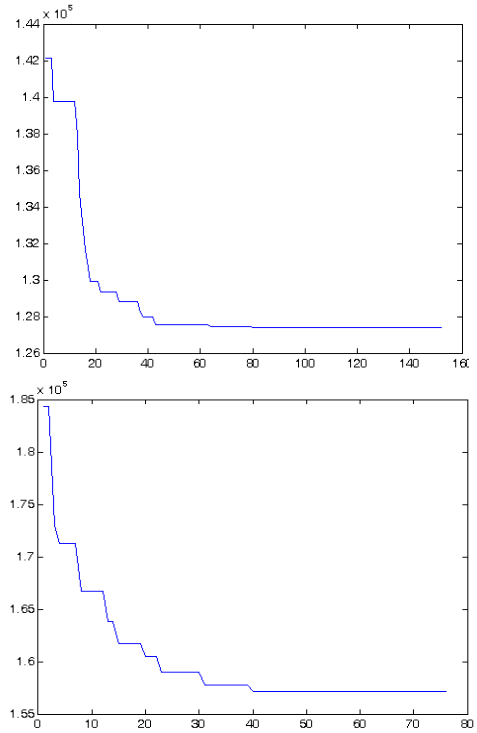

Fig. 7. Result of the proposed algorithm for two instances including Pr152 (left curve) and Pr76 (right curve)

To show exactly the correctness and effectiveness of the proposed algorithm, Figure 7 illustrates 2 traveling salesmen problems with 76 and 152 nodes. This figure represents the results and the convergence curve of the proposed algorithm. In the two curves, the vertical axis shows the total distance gained by the algorithm and the horizontal axis shows the number of iteration as termination conditions of the proposed algorithm.

In Figure 8, the values obtained by the proposed algorithm, MACO and MGA are shown. The results indicate that although the MGA has a very good ability for small problems, the MACO has an approximately similar behavior for large problems, and can converge to the best solutions. However, this ability for the proposed algorithm is satisfied for all of the instances.

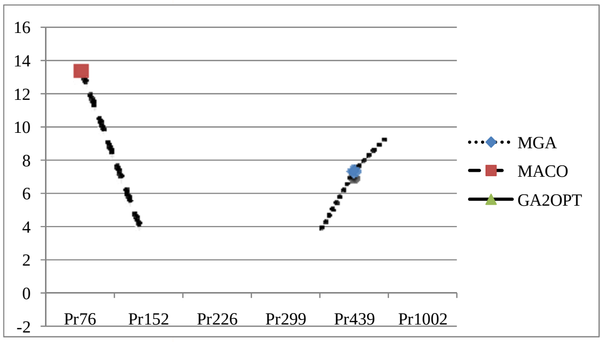

Fig. 8. Comparison between gap of the proposed algorithm and other metaheuristic algorithms

5. Conclusions

In this paper, a hybrid algorithm combining modified GA and 2-opt local search was proposed for solving the MTSP. The GA2OPT is more efficient than the modified ant colony optimization and modified genetic algorithm. For large-size problems, in particular, this algorithm yields better solutions compared with the previous algorithms. Also it seems that the combination of the proposed algorithm with other metaheuristics algorithms including simulated annealing, ant colony optimization, tabu search, etc. will yield better results. Furthermore, using this proposed algorithm for all versions of the vehicle routing problem is suggested for future research.

6. Acknowledgements

The authors would like to extend their gratitude to Hamedan Branch, Islamic Azad University for the financial support of this work.

7. References

[1] R. D. Angel, W. L. Caudle, R. Noonan, A. Whinson, Computer assisted school bus scheduling. Management Science, 18, 279-288, 1972.
[2] S. R. Balachandar, K. Kannan, Randomized gravitational emulation search algorithm for symmetric traveling salesman problem. Applied Mathematics and Computation. 192(2), 413-421, 2007.
[3] T. Bektas, The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega. 34, 209–219, 2006.
[4] L. Bianchi, J. Knowles, N. Bowler, Local search for the probabilistic traveling salesman problem: Correction to the 2-p-opt and 1-shift algorithms. European Journal of Operational Research. 162(1), 206-219, 2005.
[5] B. Bontoux, C. Artigues, D. Feillet, A Memetic algorithm with a large neighborhood crossover operator for the generalized traveling salesman problem. Computers & Operations Research. 37(11), 1844-1852, 2010.
[6] A. E. Carter, C. T. Ragsdale, A new approach to solving the multiple traveling salesperson problem using genetic algorithms. European Journal of Operational Research. 175, 246–257, 2006.
[7] N. Christofides, S. Eilon, An algorithm for the vehicle dispatching problem. Operations Research Quarterly. 20, 309-318, 1969.
[8] J. F. Cordeau, M. Dell'Amico, M. Iori, Branch-and-cut for the pickup and delivery traveling salesman problem with FIFO loading. Computers & Operations Research. 37(5), 970-980, 2010.
[9] S. Ghafurian, N. Javadian, An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems. Applied Soft Computing. 11(1), 1256-1262, 2011.
[10] F. Glover, Future paths for integer programming and links to artificial intelligence. Computers Operations Research. 13(5), 533.549, 1986.
[11] P. Junjie, W. Dingwei, An ant colony optimization algorithm for multiple traveling salesman Problem, In ICICIC '06: Proceedings of the First International Conference on Innovative Computing. Information and Control, 210–213, 2006.
[12] D. Karapetyan, G. Gutin, Lin-Kernighan heuristic adaptations for the generalized traveling salesman problem. European Journal of Operational Research. 208, 221–232, 2011.
[13] D. Kaur, M. M. Murugappan, Performance enhancement in solving traveling salesman problem using hybrid genetic algorithm. Fuzzy Information Processing Society, NAFIPS. 1-6, 2008.
[14] D. S. Orloff, Routing a fleet of M vehicles to/from a central facility. Networks. 4, 147-162, 1974.
[15] C. R. Reeves, editor. Modern heuristic techniques for combinatorial problems. Blackwell Scientific Publishing, Oxford, England, 1993.
[16] M. W. P. Savelsbergh, The general pickup and delivery problem. Transactions of Science. 29, 17-29, 1995.
[17] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lub, Q. X. Wang, Particle swarm optimization-based algorithms for TSP and generalized TSP. Information Processing Letters. 103, 169–176, 2007.

[18] T. Tang, J. Liu, multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex. European Journal of Operational Research. 24, 267- 282, 2000.
[19] S. Yadlapalli, W. A. Alik, S. Darbha, M. Pachter, A Lagrangian-based algorithm for a Multiple Depot. Multiple Traveling Salesmen Problem, Nonlinear Analysis: Real World Applications. 10(4), 1990-1999, 2009.
[20] W. Zhong, J. Zhang, W. Chen, A novel discrete particle swarm optimization to solve traveling salesman problem. Proc. Evolutionary Computation. 3283–3287, 2007.