J. Ind. Eng. Int., 7(15), 68-75, Fall 2011
ISSN: 1735-5702
© IAU, South Tehran Branch

A modified elite ACO based avoiding premature convergence for traveling salesmen problem

Majid Yousefikhoshbakht[1*]; *Ebrahim Mahmoodabadi*[2]; *Mohammad Sedighpour*[3]

[1] *M.Sc., Young Researchers Club, Hamedan Branch, Islamic Azad University, Hamedan, Iran*

[2] *M.Sc., Dept. of Mathematics, Faculty of Science, Bu-Ali Sina University, Hamedan, Iran*

[3]*M.Sc., Hamedan Branch, Islamic Azad University, Hamedan, Iran*

Received: 3 October 2009; Revised: 19 September 2010; Accepted: 19 December 2010

Abstract: The Traveling Salesman Problem (TSP) is one of the most important and famous combinational optimization problems that aims to find the shortest tour. In this problem, the salesman starts to move from an arbitrary place called depot and after visiting all nodes, finally comes back to depot. Solving this problem seems hard despite of program statement is simple and leads this problem belonging to NP-hard problems. In this paper, the researchers present a modified Elite Ant System (EAS) which is different from common EAS. There is a linear function used here for increasing coefficient pheromone of the best route activated when a better solution is achieved. This process avoids the premature convergence and makes better solutions. The results on several standard instances show that this new algorithm gains more efficient solutions compared to other algorithms.

Keywords: *Ant colony optimization; Traveling salesman problem; NP-hard problems; Meta-heuristic algorithms*

1. Introduction

For many years, researchers have paid so much attention to the traveling salesman problem (TSP) while there are many problems developed out of it, like the vehicle routing problem and the multiple traveling salesmen problem and so on. Besides, it has many applications in other problems including the computer wiring, sequencing job, designing hardware devices and radio electronic devices in communications, architecture of computational networks, etc. (Kureichik and Kureichik, 2006). Therefore, many algorithms involving exact, heuristic and meta-heuristic have been applied and so much effort has been made to find a better solution. There have been many papers proposing exact algorithms for solving the TSP. These algorithms are based on Lagrange relaxation (Qinghong and Zhang, 1999), branch-and-cut method (Cordeau *et al.*, 2010) and branch and bound (Carpaneto and Toth, 1980).

As the TSP is known to be NP-hard, (Garey and Johnson, 1979) exact algorithms are not capable of solving problems for large dimensions. On the other hand, heuristics are thought to be more efficient for complex TSPs and have become very popular for some researchers. There are three types of heuristics in the literature:

Construction algorithms: these algorithms probably are the most well-known optimization strategies for solving the combinatorial optimization problems. Construction heuristics start from a null solution and generate feasible solutions by accomplishing sequences of simple steps. Some of the well-known tour construction procedures are the randomized gravitational emulation search (Balachandar and Kannan, 2007) and the partitioning approach (Karp, 1977).

Improvement algorithms: perhaps one of the best known iterative improvement algorithms for the TSP is the branch exchanges such as the 2-opt and 3-opt heuristics (Lin, 1965) and the local search (Bianchi *et al.*, 2005). The algorithm proposed by Lin and Kernighan (1973) made a great improvement in the quality of tours that can be obtained by heuristic methods. This algorithm is still widely used to generate initial solutions for other algorithms.

Meta-heuristics algorithms: Other new kinds of popular approximate methods are meta-heuristics arisen in the last 20 years. In general, it is incredibly needed to use meta-heuristics algorithms to solve complex optimization problems when dealing with them. Since meta-heuristic approaches are very efficient for escaping local optimum values (Fig. 1), they are one of the best algorithms for solving combinatorial optimization problems. That is why the recent publications are all based on meta-heuristic approaches such as genetic algorithm (GA) (Choi

*Corresponding Author Email: khoshbakht@iauh.ac.ir
Tel.: +98 2164542540

et al., 2003), neural network (Masutti and Castro, 2009), memetic algorithm (Bontoux *et al.*, 2010) and ant system (AS) (Ghafurian and Javadian, 2011).

The TSP in investigations is defined as follows;

Let $V = \{1,...,n\}$ and $A = \{(i, j): i, j \in V, i \neq j\}$ respectively be node and arc collection of complete indirection graph $G = (V, A)$ (if the graph is not complete, we can instead lack of each arc with the arc that has an infinite size). In this problem, if $C = (c_{ij})$ shows cost matrix on graph G, then aimed to minimize cost for a traveling salesman who starts from arbitrary node as depot and after visiting other nodes exactly and finally comes back to depot.

Furthermore, TSP can be presented as several various states of integer linear programming problem. Then we only point formula presented by Dantzig in 1954. We use the following variables for modeling:

C that shows cost matrix on graph G first is symmetric matrix and secondly it is true in triangle inequality. It means $c_{ij} + c_{jk} \geq c_{ik}$ 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, the TSP model is to:

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

subject to

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

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

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad (S \subset \{0,...,n\}, |S| \geq 2) \qquad (3)$$

$$x_{ij} \in \{0,1\} \qquad (4)$$

In this formulation, Constraint (1) means that only one arc can be entered into each node; however, Constraint (2) shows that only one arc can be exited from each node. Generally, these two constraints guarantee gaining a solution that is true in Hamilton circle.

Constraint (3) is the standard sub-tour elimination constraint to prevent from creating sub tour (Dantzig *et al.*, 1954). Finally, Constraint (4) defines binary condition for x_{ij}.

In the following parts of this paper, history of TSP is presented in Section 2. In Section 3, we are going to explain the main idea of ant colony algorithm (ACO) and review briefly the works performed on this algorithm. Then we especially explain EAS and express its weakness in Section 4. In Section 5, the proposed idea will be explained in detail and its efficiency and performance reasons will be described. In Section 6, the proposed algorithm will be compared with some of the other algorithms on standard problems, which are included in TSP library, and finally in Section 7, the conclusions are presented.

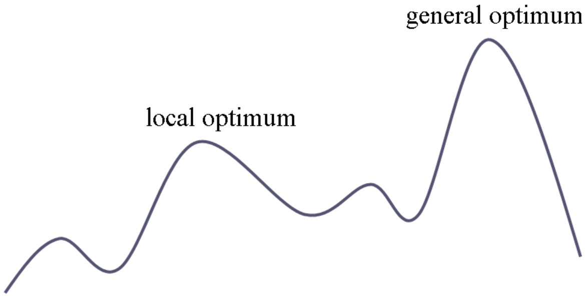

Figure 1: Local and general optimums.

2. Literature review

The TSP is one of the most popular classical problems, which is important in most of the sciences, while many of scientific problems can be converted to TSP and be solved. Although this problem has simple and understandable formulation, it is one of the elements of NP-hard problems. In other words, when the problem size grows, the exact methods cannot solve it, so we have to use heuristic methods and settle for the suboptimal solutions. Another feature that attracts more attention to this problem is the usage of this problem as a standard instance for testing and comparing the efficiency of new algorithms. It is noted that the algorithms successfully used in this problem can usually perform efficiently in other problems as well.

At first, Euler and Vandermonde formally discussed the TSP as the Knight's Tour problem in the mid-1700s. In the 1800s, this problem was recognized as one of the graph theory problems and was studied by Irish and British mathematicians Sir William Rowan Hamilton and Thomas Penyngton Kirkman, respectively. Specifically, Hamilton was the creator of the Icosian Game in 1857 based on TSP. At almost the same time as Hamilton, the problem was also acknowledged by the mathematician Thomas Penyngton Kirkman.

The German manual published first presented TSP in 1832, entitled "The Traveling Salesman", how he should be and what he should do to get commissions and to be successful in his business by a veteran Traveling Salesman. It was also followed by several applied appearances of the problem in the late 19th century and early 20th century.

However, the general form of TSP was first studied by Karl Menger in Vienna and was the first to introduce it to the Unites States in the late 1920’s or early 1930’s when visiting as a lecturer at Harvard University. Another mathematician called of Whitney was doing his PhD research in graph theory coincidently with Karl Menger. A few years later, a presentation was made by him at Princeton University where he posted the problem of finding the shortest route among the 48 states of America.

Beginning in 1954, researchers such as Dantzig, Fulkerson, and Johnson were able to find a solution to the TSP for a maximum of 49 cities (that is, Washington D.C. and the 48 capitals of the 48 states of the USA) (Dantzig *et al*., 1954). At this time, they introduced a new method for solving the TSP, the cutting plane method, which became a prototype in integer linear programming. Since that time TSP has been considered as one of the classic models in combinatorial optimization, it is used as a test case for virtually all advancements in solution procedures. TSP problem for 120 cities is solved with only 13 iterations without using a branch and bound techniques.

Over the past 29 years, the record for the largest nontrivial TSP instance solved to optimality has increased from 318 cities by Crowder & Padberg (1980) to 532 cities by Padberg & Rinaldi (1987) to 666 cities by Grötschel & Holland (1991) to 1002 and 2392 cities by Padberg & Rinaldi (1991) to 3038, 4461, 7397, 13509, 15112 and 24978 cities by Applegate *et al*. (1995) to 33810 by Concorde Team, which is the most recent ‘TSP world record’ that has been published so far.

3. Ant colony optimization

One of the most important algorithms which proved its performance for TSP is the ACO that is used by Dorigo *et al.* to solve the combinational optimization problems in 1991 and 1992 (Dorigo, 1992). This population-based approach has been successfully applied to several NP-hard combinatorial optimization problems such as vehicle routing problem (Bullnheimer *et al*., 1999) and communications networks (Di Caro and Dorigo, 1998). The ACO, inspired by the nature, simulates the natural ant treatment for food finding and applies it for solving the combinational optimization problem for which has not been found any effective algorithm yet. Studies on real ant show that despite the ants do not have the sense of seeing; they can find the shortest path from the food sources to the nest (Figure 2). Some evaporated material called pheromone, secreted by the ants when they move from one place to another to find the shortest path. Ants secrete this chemical material first for guiding other ants, which are going to exit the nest later, secondly, for recognition of the return path to the nest. Hence the route that ants travelled is marked by this chemical material so after a few times, at the same time, more ants pass this shorter path and remain much more pheromone on this shorter path. In addition, the ants' instinct with more probability select the route, which has more pheromone than others.

This experience shows that the simple swarm intelligence, which is used by ants for finding food, leads to solve the hard combinational problems and reach a solution, which is near to the optimal situation.

The ACO has been developed well lately; modifying the method of updating the local and global pheromones, and the distribution of ants on the nodes, are some examples. These developments lead to more efficient algorithms like EAS, ant colony system (ACS), rank based ant system (RAS), and max-min ant system (MMAS). On the other hand, application and efficiency of these algorithms have gained more attention, compared to some other meta-heuristic algorithms including GA, Simulated Annealing, etc. Despite good advantages of ACO algorithm compared to the other methods, the large distances between the solutions cause difficulties in developing better solutions.

Many researchers have tried to modify this algorithm as much as possible to overcome these difficulties. The investigations by Qinghong and Zang (1999), Bin and Zhongzhi (2001) and Yousefikhoshbakht *et al.* (2009; Yousefikhoshbakht and Zafari, 2008) are some examples of these efforts.

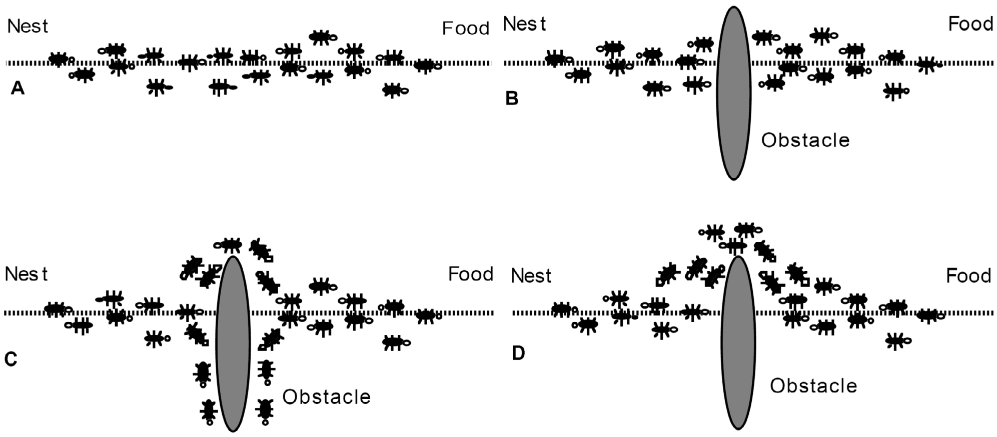

Figure 2: (A) Real ants follow a path between nest and food source,
(B) An obstacle appears on the path: Ants choose whether
to turn left or right with equal probability,
(C) Pheromone is deposited more quickly on the shorter path,
(D) All ants have chosen the shorter path.

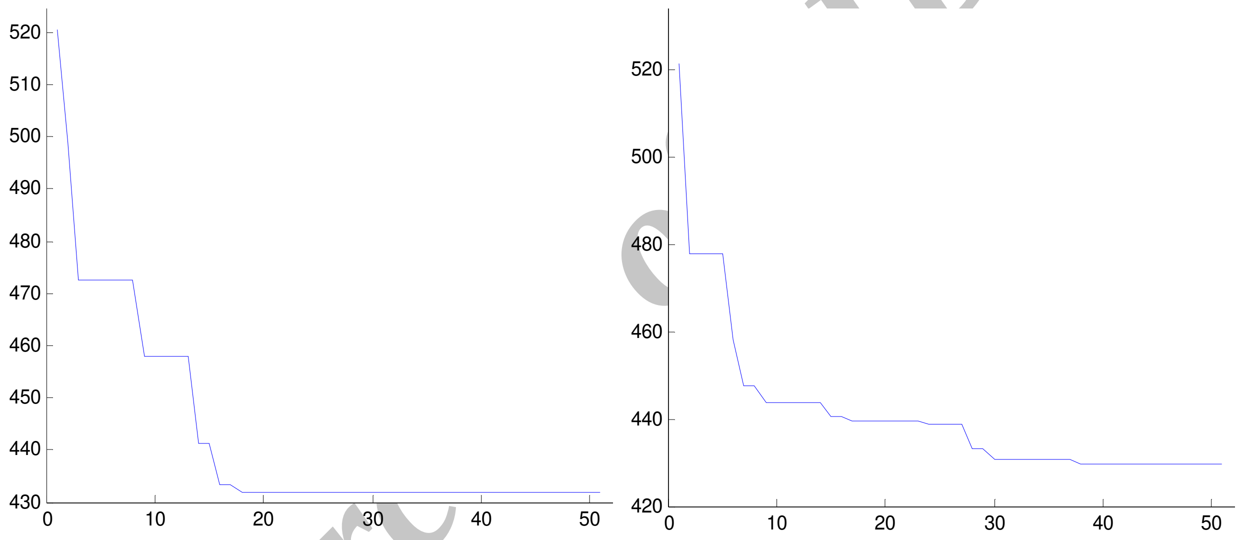

Figure 3: Comparison between the EAS and new EAS for solving Eli51.

4. Elite ant system

The first algorithm of ACO family was the AS used for solving small scale TSP instances. In the face of developing good solutions for small scale problems of TSP type, AS algorithm failed to reach an acceptable efficiency in large scale problems in comparison with the famous algorithms of that time. Much effort was performed to solve the problem; the first modification applied on the AS algorithm was the usage of elitist strategy, published by Dorigo *et al.* (1996). Based on this algorithm, in addition to the local releasing of pheromone on the arcs which the ants have passed through, the arcs belonging to the best route (T^*) are released with pheromone and are encouraged with the constant coefficient e in the following way;

$$\Delta\tau_{ij}^{gb}(t) = \begin{cases} e / L^{gb}(t) & (i,j) \in T^* \\ 0 & (i,j) \notin T^* \end{cases} \qquad (5)$$

This process causes that the arcs belonging to the best route in any iteration are more highlighted, and to be updated according to the value of the best route L^{gb}. Note that, the above operator indicates; the less the value of L^{gb}, the more pheromone released on the arcs.

In spite of better results obtained by EAS algorithm, premature convergence occurred during the calculation process. In other words, releasing global pheromone with constant coefficient leads to a very fast and early concentration of searching procedure around suboptimal solutions. In addition, this will cause an early stagnation of premature search and settling for a local optimal. Premature convergence makes ants trail similar route and gains similar solutions after several times. In this way we cannot get to better solutions. It shows that system stops from finding possibly better routes and cannot gain a better tour.

Updating the pheromone simulates the changes in values of pheromone in any iteration and mainly it is one of the reasons that algorithms are different. Generally, two operations motivate this updating procedure in EAS algorithm:

First, releasing new pheromone on the arcs, locally and globally; this operation leads to increase pheromone on the arcs. Second, evaporation of pheromone; this operation leads to decrease pheromone on the arcs with constant rate ρ , in other words, at the end of each iteration of algorithm, the value of pheromone left on the arcs is decreased by the constant coefficient ρ . Thus, the new footprint of pheromone has an average weight between the value of the pheromone left on the arcs, and the value of new pheromone released in the arcs. Thus the formula of updating pheromone in the EAS algorithm is:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k}(t) + \Delta\tau_{ij}{}^{gb}(t) \quad (6)$$

where ρ is the evaporation rate, which is a constant value in [0,1] domain. It regulates the decrease of phermone on arcs.

$$\Delta\tau_{ij}{}^{k}(t) = \begin{cases} 1/L^{k}(t) & (i,j)\in T^{k} \\ 0 & (i,j)\notin T^{k} \end{cases} \quad (7)$$

T^{k} :The collection of arcs passed over by the ant '*k*'.

$\Delta\tau_{ij}{}^{k}(t)$: As mentioned above, is the formula of updating the pheromone. Thus ants passing over the arc between nodes *i* and *j*, release some pheromone on it. The value of released pheromone is one over the value passed yet.

5. Presented algorithm

In EAS algorithm, the probability of movement from node *i* to node *j* – which is not visited yet - by ant '*k'* is presented as follows:

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{r\in J_i^k}\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)} & j\in j^{*} \\ 0 & j\notin j^{*} \end{cases} \quad (8)$$

Where

j^{*} :Collection of nodes that has not been visited by ant '*k*' yet.

$\tau_{ij}(t)$: The value of phermone on the arc joining *i* to *j*.

$\eta_{ij}(t)$: The heuristic information for the ant visibility measure (e.g., defined as the reciprocal of the distance between node *i* and node *j* for the TSP).

α,β: The controling parameters by user, which determine the ratio of importance of ant's visiblity measure, compared to the value of pheromone releaseed on arc (*i,j*) on one hand, and compared to the heuristic information on arc (*i,j*) on the other hand.

So, the value of pheromone released on an arc is important in selecting it by an ant. On the other hand, the construction of ACO algorithm is such that the precision of early solutions are low but it increases gradually by the increase of iterations and releasing pheromone by ants. Therefore, a constant coefficient like *e* cannot be the appropriate formula for encouraging the best route that has been gained so far, because it does not signify in which iteration and in which level of precision, we get to the best result. In developing and modifying EAS algorithm, the researchers found that the usage of an increasing linear function like *e*+*n*/2, when a better solution is gained, in addition to small value like *e* in beginning of algorithm, has better conclusions. In the next section, an extensive comparison in efficiency and performance of modified algorithm to the previous algorithms is presented.

Moreover, the linear function seems to be a good choice because first, it is an increasing function and second, it has a constant gradient. In other words, as the iterations increase and better results are obtained, the corresponding function grows linearly and appreciates the best route obtained.

It is noticeable that the low value of the linear function at the beginning of algorithm, here *e*=1, lead to less influence of released pheromone on the selection of routes in the next algorithm iteration. In other words, if ants gain weak solutions, they forget them, but as the algorithm proceeds and the solutions are more precise, the pheromone is increased and this causes the best solution to be encouraged more powerfully.

Most successful meta-heuristic methods the same as ACO have paid attention to global search and search in the whole solution space as far as possible. As the algorithm proceeds, they move to better solutions and alter the global search for local. We have considered the point too and chosen this linear function. Note that encouraging new footprints if happened at the beginning of algorithm, might lead to an early stagnation and premature convergence to suboptimal regions. In other words, before the algorithm finishes a complete global search, tends to the local search and consequently relatively weak results are attained. Thus, as mentioned before, encouraging the best solution during the first steps of the ACO algorithm, where decisions are almost by chance, should be little but gradually increase.

Moreover, in order to help local search during the algorithm steps, not leading to any stagnation, the researchers used a local searching algorithm when attaining a better solution compared to previous iterations. In fact, the probability of finding better solutions near a good solution is relatively high. There exist many algorithms for the local search and they have of course pros and cons. Here if we use the 2-opt local search, as not only it is simple, but also does not influence the algorithm's complications.

6. The numerical calculations

In this section, some numerical results of comparison between the proposed modified algorithm and some previous algorithms are presented. These algorithms are applied and tested on several instances from TSP problems of TSPLIB including Eli51, KroA100, KroB100, Pr152 and Krob200. The following number in these instances shows the number of related cities. Note that the EAS algorithm is selected as the basic algorithm. In these tests we compared the efficiency and performance of the EAS algorithm and the new EAS algorithm with GA, AS, ACS and mixed GA and ACO. As expected, the results of this comparison show that the EAS method gains better solutions than the GA, and it gains worse solutions than the ACS algorithm. But better solutions are developed applying this modification, and also the results are better compared to the ACS algorithm. Additionally, in order to recognize the performance of the method, the best solutions published in the literature and also on the web, are presented in table 1. The pack of optional parameters obtained through several tests is as follows:

$$\alpha = 1, \beta = 5, \rho = 0.5, \ Q = 100, e = 1 \qquad (9)$$

Furthermore, two solutions for Eli51 instance is presented in Figure 3 for better showing the difference between two mentioned algorithms. On the other hand, the new algorithm devotes more time than another algorithm for global searching based on presented analysis and employs local search with more powerful after that. For example in iteration 20, the old algorithm is completely stopped and then cannot gain a better solution and it is trapped in local optimum, but although the new algorithm has a less solution in this time, it can escape from local optimum because of using mentioned solution after this iteration. Thus, it can gain better solutions than the old algorithm can. So, albeit the proposed algorithm has less speed for finding solution compared to the old algorithm, it can overcome local optimums and tends to better solutions.

7. Conclusion

In this paper, a new modification on the EAS algorithm is presented which is different from the common EAS algorithm in the updating method of the pheromone. In this algorithm, when the number of iterations of the algorithm is increased and better solutions are attained, the encouragement of the best route is performed more powerfully. Note that here the researchers have used a linear function to encourage the best route. Presenting other functions instead of this linear function, improving the algorithm more, and also applying this method in other combinational optimization problems including the vehicle routing problem, multiple traveling salesmen problem and the sequencing of jobs are postponed to the next papers.

Table 1: Comparison of algorithms for standard problems of TSP (the best conclusions for 20 test has presented).

Instance	New EAS	EAS	GA+ACO	ACS	GA	AS	Best
Eli51	426	430.35	427	427	431	441	426
KroA100	21282	21740.78	21521	21521	21802	21799	21282
KroB100	22141	22588.56	22170	22274	22641	23005	22141
Pr152	73682	74135.10	74121	73988	74976	74429	73682
Krob200	29437	30110	29576	29655	30277	30421	29437

References

Applegate, D.; Bixby, R.; Chvatal, V.; Cook, W., (1995), *Finding cuts in the TSP (A preliminary report)*. DIMACS Technical Report 95-05, March 1995.

Balachandar, S. R.; Kannan, K., (2007), Randomized gravitational emulation search algorithm for symmetric traveling salesman problem. *Applied Mathematics and Computation*, 192(2), 413-421.

Bianchi, L.; Knowles, J.; Bowler, N., (2005), 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.

Bin, W. U.; Zhongzhi, SHI., (2001), A kind of subsection algorithm based on ACO solving the TSP problem. *Computer transaction*, 1328-1333.

Bontoux, B.; Artigues, C.; Feillet, D., (2010), A memetic algorithm with a large neighbourhood crossover operator for the generalized traveling salesman problem. *Computers & Operations Research*, 37(11), 1844-1852.

Bullnheimer, B.; Hartl, R. F.; Strauss, C., (1999), An improved ant system algorithm for the vehicle routing problem. *Annals Operations Research*, 89, 319-328.

Carpaneto, G.; Toth, P., (1980), Some new branching and bounding criteria for the asymmetric traveling salesman problem. *Management Science*, 26, 736-743.

Choi, I. C.; Kim, S. I.; Kim, H. S., (2003), A genetic algorithm with a mixed region search for the asymmetric traveling salesman problem. *Computers & Operations Research*, 30(5), 773-786.

Cordeau, J. F.; Dell'Amico, M.; Iori, M., (2010), Branch-and-cut for the pickup and delivery traveling salesman problem with FIFO loading. *Computers & Operations Research*, 37(5), 970-980.

Crowder, H.; Padberg, M. W., (1980), Solving large-scale symmetric traveling salesman problems to optimality. *Management Science*, 26, 495-509.

Dantzig, G. B.; Fulketson, D. R.; Johnson, S. M., (1954), Solution of a large-scale traveling-salesman problem. *Operations Research*, 2, 393-410.

Di Caro, G.; Dorigo, M., (1998), Ant net: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9, 317-365.

Dorigo, M., (1992), *Optimization, learning and natural algorithms*. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 140.

Garey, M. R.; Johnson, D. S., (1979), *Computers and intractability: A guide to the theory of NP-completeness*. San Francisco: W. H. Freeman.

Ghafurian, S.; Javadian, N., (2011), An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems. *Applied Soft Computing*, 11(1), 1256-1262.

Grotschel, M.; Holland, O., (1991), Solution of large-scale symmetric traveling salesman problems. *Mathematical Programming*, 51, 141-202

Hashemi, S. M.; Yousefi Khoshbakht, M.; Zafari, A., (2009), *Solving multiple traveling salesman problem by a modified elite ant colony system*. The 40th Annual Iranian Mathematics Conference (AIMC), Sharif University of Technology, Tehran, Iran.

Karp, R. M., (1977), Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane. *Mathematics of Operations Research*, 2, 209-224.

Kureichik, V. V.; Kureichik, V. M., (2006), A genetic algorithm for finding a salesman's route. *System Analysis and Operations Research,* 45(1), 89–95.

Lin, S., (1965), Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44, 2245-2269.

Lin, S.; Kernighan B. W., (1973), An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21, 498-516.

Malandraki, C.; Dial, R. B., (1996), A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *European Journal of Operation Research*, 90, 45–55.

Masutti, T. A. S.; Castro, L. N. D., (2009), A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem. *Information Sciences*, 179(10), 1454-1468.

Padberg, M. W.; Rinaldi, G., (1987), Optimization of a 532-city symmetric traveling salesman problem. *Operation Research Letters*, 6, 1-7.

Padberg, M.; Rinaldi, G., (1991), A branch-and-cut algorithm for the resolution of large-scale traveling salesman problems. *SIAM Review*, 33, 60-100.

Qinghong, W. U., Zhang, J., (1999), ACO with the characteristic of mutation. *Computer research and development*, 240-245.

Yadlapalli, S.; Malik, W. A.; Darbha, S.; Pachter, M., (2009), A lagrangian-based algorithm for a multiple depot, multiple traveling salesmen problem. *Nonlinear Analysis: Real World Applications*, 10(4), 1990-1999.

Yousefi Khoshbakht, M.; Zafari, A., (2008), *A new ant colony algorithm for solving multiple traveling salesman problem.* The 2nd Joint Congress on Intelligent and Fuzzy Systems (ISFS2008), Malek-Ashtar University of Technology, Tehran, Iran.