ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **16**, Number 1, 2013, 65–80

Modification of the Ant Colony Optimization for Solving the Multiple Traveling Salesman Problem

Majid YOUSEFIKHOSHBAKHT, Farzad DIDEHVAR, Farhad RAHMATI

Department of Mathematics and Computer Science,
Amirkabir University of Technology
No. 424, Hafez Avenue, Tehran 15914, Iran

E-mail: `didehvar@aut.ac.ir`

Abstract. This article presents a new modified version of the ant colony optimization (ACO) mixed with insert, swap and 2-opt algorithm called NMACO for solving the multiple traveling salesman problem (MTSP) which utilizes an effective criterion for escaping from the local optimum points. The MTSP is one of the most important combinatorial optimization problems in which the objective is to minimize the distance traveled by several salesmen for servicing a set of nodes. Since this problem belongs to NP-hard Problems, some meta-heuristic approaches have been used to solve it in recent years. In contrast to the classical ACO, the proposed algorithm uses only a global updating for the current best solution and the best found solution until now. Furthermore, a new state transition rule and an efficient candidate list are used in order to assess the efficiency of the proposed algorithm. The proposed algorithm is tested on some standard instances available from the literature and their results were compared with other well-known meta-heuristic algorithms. The results indicate that the NMACO has been able to improve the efficiency of the ACO in all instances and is quite competitive with other meta-heuristic algorithms.

Key words: Multiple Traveling Salesman Problem, Ant Colony Optimization, Updating Pheromone, Combinatorial Optimization Problems.

1. Introduction

Ant colony optimization (ACO) is one of the most popular algorithms in the research field of swarm intelligence. ACO has been inspired by the behavior of real ants seeking a path between their colony and a source of food (Fig. 1).

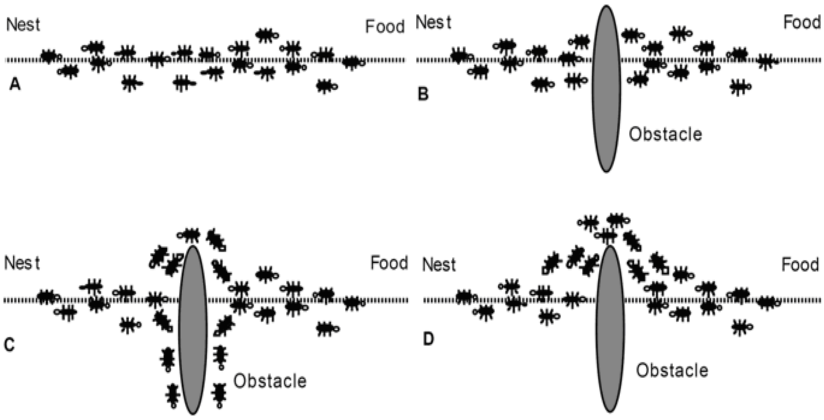

Fig. 1. (A) Real ants follow a path between the nest and a food source; (B) An obstacle appears on the path: Ants choose whether to turn left or right with equal probability; (C) Pheromone is deposited more quickly on the shorter path; (D) All ants have chosen the shorter path.

The ACO is an iterative stochastic search technique which was proposed by Marco Dorigo in 1992 [1]. While walking between their colony and the food source, ants deposit pheromones along the paths they move. The pheromone level on the paths increases with the number of ants passing through and decreases with the evaporation of pheromone. As time passes, shorter paths attract more pheromone. Consequently, pheromone intensity helps ants to identify shorter paths to the food source. The first version of ACO called Ant System (AS) aimed at searching for an optimal path between two nodes in a graph. It should noted that a problem like traveling salesman problem (TSP) is divided into some sub-problems in which the simulated ants are expected to select the next node based on the amount of the pheromone in a trail and the distance to the next node. The probability of movement from node i to node j which is not visited yet by ant 'k' is presented by formula (1):

$$P_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum\limits_{j\in N_i}\tau_{ij}^{\alpha}\eta_{ij}^{\beta}} & \text{if} \quad j \in N_i^k \\ 0 & \text{if} \quad j \notin N_i^k \end{cases} \tag{1}$$

where:
N_i^k: The collection of nodes that hasn't been visited by ant 'k' yet;
τ_{ij}: The value of pheromone on the arc joining i to j;

η_{ij}: The inverse distance between node i and j. However, both are powered by α and β which can be changed by the user. Therefore, their relative importance can be altered;

α, β: The controlling parameters which are set by the user and determine the ratio of the importance of ant's visibility measure compared to the value of pheromone release on arc (i, j).

Ants release $\Delta\tau_{ij}$ called "pheromone information" on the respective path while moving from node i to node j. This value can be calculated by formula (2).

$$\tau_{ij}(t) \leftarrow \tau_{ij}(t) + \Delta\tau_{ij} \tag{2}$$

If τ is the matrix for the existing pheromone on the edges of the respective graph, the algorithm like its natural version makes use of pheromone evaporation $(1 - \rho)$ in order to prevent the rapid convergence of ants to a sub-optimal path by formula (3). In other words, pheromone density on each edge is reduced in each iteration by $0 \leq \rho \leq 1$ which is set by the user.

$$\tau \leftarrow (1 - \rho)\tau \qquad \rho \in [0, 1] \tag{3}$$

In the face of gaining good solutions for small scale problems of TSP by AS (Fig. 2 [2]), this algorithm could not produce satisfactory results in large scale problems of TSP compared with meta-heuristic algorithms of the time. Therefore, scientists attempted to develop a newer version of the algorithm which was able to produce better results. This algorithm has attracted the attention of an increasing number of researchers and several versions of ACO algorithms have been proposed in the literature such as elitist strategy AS (EAS) [3], ant colony system (ACS) [4], max-min AS (MMAS) [5], rank-based AS (RAS) [6]. These algorithms are different in some ways, but all of them are based on a stronger exploitation of the search history to direct the ants' search process and share the same main idea which is the indirect communication between the individuals from a colony of ants, based on an analogy with the trails of pheromone.

The MTSP is a major problem in combinatorial optimization problems. In contrast to TSP, MTSP has less attracted significant research attention and few algorithms have been proposed for its solution. Since there is no known polynomial algorithm which can find the optimal solution in every instance, the MTSP is considered NP-hard. For such problems, the use of heuristics such as ant colony optimization (ACO) is considered a reasonable approach in finding solutions. Moreover, Since the ACO has some shortcomings like its slow computing speed and local-convergence, it cannot be directly applied to the problem with acceptable performance. Therefore, we have proposed a new modified ant colony optimization called NMACO in order to improve both the performance of the algorithm and the quality of the solutions in this paper. The proposed algorithm improved the global ability of the algorithm through importing new probability function of movement for constructing solutions by utilizing the new candidate list, updating pheromone method and several effective local searches. Therefore, the NMACO explores different parts of the solution space so that the search method is not trapped at the local optimum. The experimental results have shown that the proposed algorithm is very efficient and competitive in terms of solution quality.

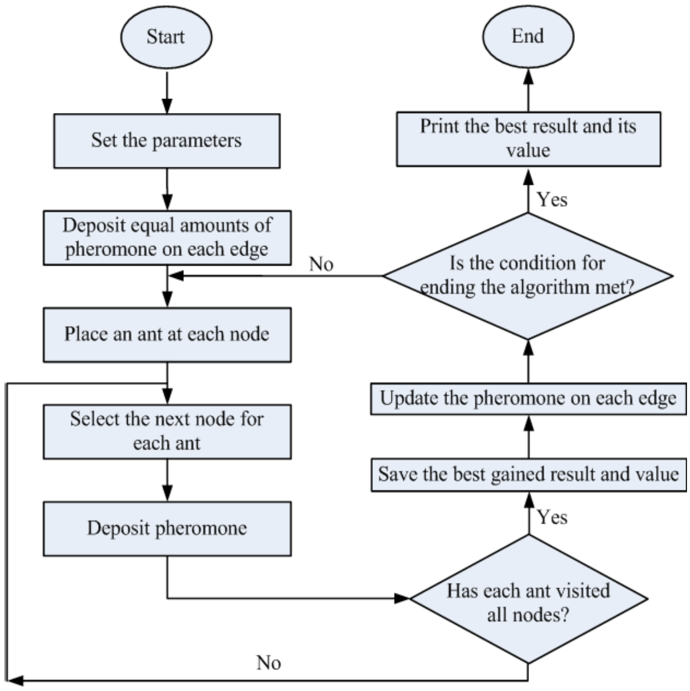

Fig. 2. The algorithm for solving the TSP.

The structure of the remainder of the paper is as follows. In the next section, related works on MTSP are presented. In Section 3, the proposed idea based on ant colony optimization (ACO) is explained in great detail. In addition, using a new candidate list, building the solution simultaneously using a new transition rule, applying three powerful local search techniques to improve the solution, and proposing a new method for updating global pheromone information which are four main steps of NMACO are also described in more detail in the same section. In Section 4, the proposed algorithm is compared with some of the other algorithms on standard problems belonging to OVRP library. Finally, some concluding remarks are given in Section 5.

2. Multiple Traveling Salesman Problem

The TSP is a well-known optimization problem in operations research and is great importance in other fields due to its widespread application in other problems. In this problem a salesman starts to move from a arbitrary node called node depot and returns after visiting n nodes so that each node is visited only once. The objective is to find the shortest minimum cycle. Although TSP has been considered in the research arena since the 1920s, this problem nowadays is receiving much attention by researchers and scientists. One of the most important reasons for this attention is the fact that the TSP belongs to non-deterministic polynomial (NP-hard) problems in nature. It means that it requires a nonpolynomial time complexity at runtime to produce a solution. When the size of such problems increases, it requires an impractically enormous time is required to solve them.. For this reason, numerous approaches have been proposed to produce near-optimal solutions for TSP in an acceptable CPU time.

The multiple traveling salesman problem (MTSP) is a generalization of the well-known TSP [3], where more than one salesman can be used in the solution. This problem is defined as follows. Let G = (V, E) be a completely undirected connected graph with $V = \{0, 1, ..., n\}$as the set of vertexes and the set of arcs $E = \{(i, j) : 0 \leq i, j \leq n\}$(if the graph is not complete, we can replace each missing arc with an arc which has an infinite size). Vertex 0 denotes the depot, and each vertex $i \in \{1, ..., n\}$ is a node. The cost of travel from vertex i to vertex j is denoted byc_{ij} ($c_{ii} = 0,\ 0 \leq i \leq n$), and it is assumed that costs are symmetric (i.e c$_{ij}$ = c$_{ji}$). There are m¿1 salesmen at the depot. Furthermore, each node must be serviced by a salesman and each salesman route must start at the depot and end in the depot. The objective is to define the set of cycles which minimizes the total costs. An example is depicted in Fig. 3, where $m = 5$ and $n = 50$.

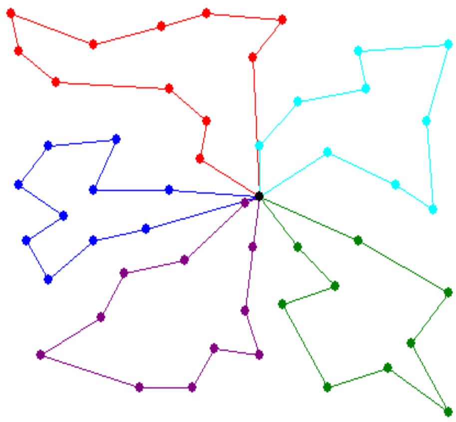

Fig. 3. Example solution of MTSP.

There is a wide variety of MTSPs and an extensive literature on this class of problems. The variants include MTSP with pickup and delivery (If the salesmen need to pick up loads) [7], time windows (If the services have time constraint) [8], multi-depot MTSP (if the problem has a single depot or multiple depots) [9], open ends (if the salesmen have not returned to the depot) [10] and others. The methods for solving MTSP can be classified as exact, heuristic, metaheuristic and hybrid algorithms. Although there are exact algorithms which investigate fewer solutions and can obtain optimal solutions for small size problems, it is almost impossible even in such cases to find an optimum solution within a satisfactory time limit. There are several exact algorithms of the MTSP such as Cutting Planes algorithm [11], Branch-and-Bound algorithm [12], and Lagrangian algorithm [13]. For instance, many researchers have tried to relax the MTSP to the TSP and apply algorithms to resolve it, but the results have always ended in degeneration [14–16]. In this method, the MTSP with m salesmen and n cities is transformed into a TSP with $n + m - 1$ cities through introducing $m - 1$ artificial depots $(n + 1, \ldots, n + m - 1)$.

The TSP belongs to the class of NP-hard problems [17] and MTSP is more difficult than TSP because it involves finding a set of Hamilton circuits without sub-tours for $m(m > 1)$ salesmen who serve a set of $n(n > m)$ cities so that each one is visited by exactly one salesman. Consequently, it is obvious that MTSP also belongs to NP-hard problems. This means that the MTSP solution time grows exponentially with the increase in distribution points. As a result, that classical optimization procedures are not adequate for this problem especially for large dimensions. In other words, as the size of the problem increases, the complexity of the problem also increases rapidly as well. A lot of algorithms have been performed on the MTSP including heuristic approaches such as k-opt approach [18], minimum spanning tree [19], and self-organizing NN approach [20].

Although heuristic methods solve NP-hard problems, they become trapped in local optima and cannot obtain a high quality solution. Consequently, due to MTSPS's high computational complexity, good metaheuristic techniques namely genetic algorithms (GA), neural network (NN), tabu search (TS), simultaneous generalized hill climbing algorithms (SGHC), and ACO are necessary to solve it.. These algorithms basically try to combine basic heuristic methods into higher level frameworks in order to explore a search space efficiently. For example, a powerful tabu search on the MTSP was used by Ryanet et al. [21] and Thompson et al. proposed a class of neighborhood search algorithms named cyclic transfers for solving the MTSP [22]. Furthermore, Somhom et.al. have applied competition-based neural network to solve MTSP with minmax objective[23] and a simultaneous generalized hill climbing algorithm (SGHC) for the MTSP was used by Jacobson *et al.* [24]. The most frequently used algorithm for solving the MTSP is genetic algorithm. Tang *et al.* have used the modified genetic algorithm to solve hot rolling scheduling problem which is an example of MTSP [25] and Brown [26] and Carter [27] *et al.* researched on chromosome representation and related genetic operators applicable to the MTSP. Finally, Ghafurian *et al.* proposed an ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems [28].

Recently, many researchers have found that the employment of hybridization in optimization problems can improve the quality of problem solving in comparison with heuristic and meta-heuristic approaches. Since hybrid algorithms have greater ability to find an optimal solution, they have been considered by researchers and scientists in recent years. Yousefikhoshbakht et al. proposed a hybrid two-phase algorithm called SW+ EAS for solving the MTSP [29]. At the first stage, the MTSP is solved by the sweep algorithm and at the second stage, the elite ant system and 3-opt local search are used for improving solutions. Furthermore, a hybrid genetic algorithm based on tabu search for the MTSP was proposed by Liaw [30].

4. The Proposed Method

In this section the new modified ACO called NMACO algorithm is described. There are several modifications in the NMACO compared to ACO including transition rule, candidate list, global pheromone updating rules and several local search techniques. In contrast to local searches, using the candidate lists leads to the concentration of the search on promising nodes in order to reduce the computational time. These modifications help to avoid premature convergence and then to search over the subspace in order to find the global optimum. The steps of the proposed algorithm are mainly described as follows:

Step 1: Build n solution of the MTSP by using a new transition rule and a candidate list.

Step 2: Use insert, swap and 2-opt moves for the current best solution and the best solution until now in order to improve them more.

Step 3: Update the global pheromone information.

At the first stage of the NMACO, $n$ants are initially positioned on the depot and each ant builds a tour for each salesman by using a candidate list and formula (4). This step is iterated for n times. Therefore, n feasible solutions for the MTSP are partially obtained in this step. The NMACO differs from the ACO due to its strategy of new transition rule. In the proposed algorithm, the next node j from node i in the route is selected by ant k, among the unvisited candidate nodesO_i^k with $n*$ number, according to the following transition rule which shows the probability of each city being visited. In other words, the ant k at point i maintains a tabu list N_i^kin memory which defines the set of points still to be visited when it is at this point.

$$P_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^\alpha \eta_{ij}^\beta \kappa_{ij}^\lambda}{\sum\limits_{j \in N_i^k} \tau_{ij}^\alpha \eta_{ij}^\beta \kappa_{ij}^\lambda} & \text{if} \quad j \in O_i^k \\ 0 & \text{if} \quad j \notin O_i^k \end{cases} \tag{4}$$

where $\tau_{ij}(t)$, $\eta_{ij}(t)$, α and β are defined in formula (1) as well. Moreover, κ_{ij} is defined as the savings of combining two nodes on one tour as opposed to serving them on two different tours and λ is a control parameter set by the user. The savings of combining any two nodes i and j are computed by$\kappa_{ij} = \alpha' c_{i0} + \beta' c_{0j} - \lambda' c_{ij}$ where node 0 is the depot and c_{ij} denotes the distance between nodes i and j.

A candidate list is used to improve the quality of the solutions and computational time in the optimization problems. For example, it is used to determine the next customer which is not visited until now and is selected in a vehicle route based on the distance to all other customers in the customer set. In the proposed algorithm, the closest unvisited nodes are only considered in the candidate list for the current node i and are made available for selection as the next node to be visited in the route. It should be noted that the size of the candidate list has been determined by restricting its size to a fraction of the total number of nodes in the problem. It is noted that this restriction prevents the algorithm from wasting its efforts to consider moves to nodes which are a great distance from the current node and have very little chance of creating an improved solution to the problem.

In the proposed algorithm, the candidate list size is set equal to 30% of the total number of nodes. For example, in problems with 50 nodes the candidate list is limited to the rounded integer value of fifteen and in problems with 200 nodes which are common in different versions of the MTSP, candidate lists might include as many as sixty nodes which is still a large number. As a result, in order to decrease the computational time and increase the probability of obtaining a higher-quality solution, upper limit 20 is fixed to the number of candidate list n^*. If the size of this list is more than the maximum 20, then 20 is replaced by 0.30 n^* using formula (5).

$$n^* = \begin{cases} 20 & 0.30\ n > 20 \\ 0.30\ n & \text{otherwise} \end{cases} \tag{5}$$

For some optimization problems, ACO even can obtain the best solutions and outperform other metaheuristics. All these powerful ACO algorithms have been applied in these problems with local search methods. This is usually achieved by improving the best solution through local search. In the interest of effectiveness and efficiency, three simple procedures including insert, swap and 2-opt moves are adopted in this study. After all ants have constructed their routes and before updating global pheromone, several local searches are performed in an attempt to increase the length of the routes. Local searches are conducted in single and multiple routes.

In single insert and single swap moves, the nodes and destination positions are selected in a way that operation causes the greatest improvement in the cost of the solution.

In a multiple insert, a candidate node i is removed from its current route (origin) and a trial insertion is made in the other route if the destination route contains at least one of the δ-nearest neighbors of i (δ is defined by the user). Furthermore, in multiple swap move, a candidate node i of R_1 is exchanged with j of R_2 if one of the destination routes contains at least one of the δ-nearest neighbor to the acceptable selected node.

The 2-opt move is implemented in one route or is applied to two routes. This algorithm is one of the most commonly encountered moves in which two edges belong to the same route or different routes which form a criss-cross are selected and two new edges are replaced. It also should be noted that the new solution will be only accepted when constraints of the MTSP are not violated and the novel tour will gain better value for problem than the previous solution.

For example, the single moves of these local searches are demonstrated in Fig. 4. It is noted that the local searches stop until no additional improvement can be obtained. Furthermore, a local search is a time-consuming procedure of ACO. To save the computation time, we will only apply a local search to the best current solution and the best known solution until now. The idea here is that better solutions may have a better chance of finding a global optimum.

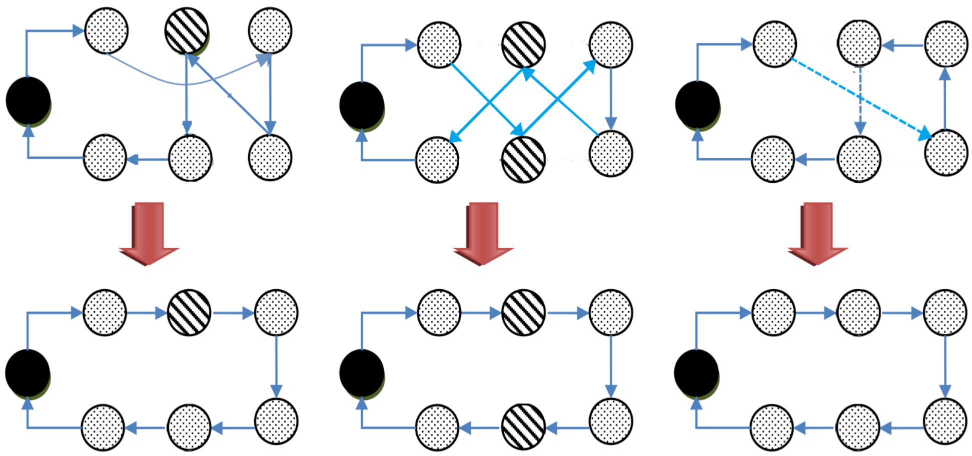

Fig. 4. Single insert (left), swap (middle) and 2-opt moves (right).

The pheromone updating formula was used to simulate the change in the amount of pheromone due to both the addition of a new pheromone deposited by ants on the visited edges and the pheromone evaporation. Using this rule, ants will search in a wide neighborhood of the best previous schedule. Another problem with all of the versions of the ACO is the fact that pheromone is deposited only on the best current route or only on the best known solution in global pheromone release. Due to the random nature of ACO, the value of the best solution function might decrease or increase in different iterations. Therefore, it is likely that in a few iterations the value of objective function does not increase and despite the fact that a solution is not good enough (especially at the start of algorithm) it is deposited with pheromone repeatedly in several iterations. Consequently, in following iterations this additional pheromone on an inappropriate solution will cause the algorithm to waste a lot of time on changing the search direction in solution space without being able to find a better solution. As a result, the pheromone of all edges belonging to the route taken by ants will be updated in the proposed NMACO. Besides, the pheromone is increased only on edges which belong to two best found solutions in current iteration and since the beginning. The idea of the proposed strategy in the context of the proposed algorithm is to place extra emphasis on the two best paths found after each iteration. This modification leads to a balance between exploitation (through emphasizing the global best ant) as well as exploration (through the emphasis to iteration best ant).

As mentioned above, in the proposed algorithm, pheromone is released on the edges of the current best solution (T^{cbs}) and the best known solution until now (T^{bks}) with coefficient e in each iteration based on the formula (6):

$$\Delta\tau_{ij}^{cbs(bks)}(t) = \begin{cases} \dfrac{e}{L^{cbs}(t)} \left(\dfrac{e}{L^{bks}(t)}\right), & (i,j) \in T^{cbs(bks)} \\ 0, & (i,j) \notin T^{cbs}(\text{bks}) \end{cases} \tag{6}$$

In this way, the edges of the shortest path up to the current iteration including T^{cbs} and T^{bks} become more attractive and are updated based on the values of the best L^{cbs} and L^{bks} tours. It should be noted that the lower the value of the tours in formula (6), the greater the pheromone deposited on the visited edges. The updating formula of the pheromone in the proposed algorithm is:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^{cbs}(t) + \Delta\tau_{ij}^{bks}(t), \tag{7}$$

where the parameter $0 < \alpha < 1$ is the persistence of the pheromone trail so that $1-\alpha$ indicates the evaporation rate and is the amount of pheromone which an ant deposits on the trail. The pheromone updating reflects the desirable factors on that trail such as shorter distance, better performance, etc., depending on the application problem. Furthermore, it results in the new pheromone trail being a weighted average between the old pheromone value and the amount of pheromone deposited. It should be noted that if the best solution till now does not improve within a given ten generations in the proposed algorithm, the algorithm will be stopped. A pseudo-code of the NMACO for the MTSP is presented in the Fig. 5.

1) Set the parameters.
2) Place all ants at the depot.
3) Select a nest node for each ant based on a new proposed formula and efficient candidate list.
4) Deposit pheromone.
5) if there is a node that has not been visited, go to step 3.
6) Save the best solution and its value obtained in current iteration.
7) Update the best solution and its value obtained until now.
8) Apply the local searches for two best solutions in current iteration and obtained until now.
9) Update global pheromone on each edge.
10) If the best solution till now is improved within ten iterations, go to step 2.
11) Print the best solution and its value.

Fig. 5. The NMACO for MTSP.

5. Results

In this section, first, the best results of NMACO are compared with the best results of classic ACO for some standard MTSPs. These results have been presented in Table 1. Then, in order to assess the effectiveness of the proposed algorithm, its results and some meta-heuristic algorithms have been compared and presented in Tables 2 and 3. Table 1 contains six standard MTSPs which possess an acceptable number of nodes whose sizes are between 76 and 1002. These instances which are shown in Table 1 belong to TSP problems of TSPLIB including Pr76, Pr152, Pr226, Pr299, Pr439 and Pr1002. For each instance, the number of nodes, n, the number of salesmen, m, the number of independent algorithm implementation to find the best value, *tbest*, and the max number of nodes which a salesman can visit, u, is presented. Besides, the algorithms are coded by Matlab and implemented on a Pentium 4 PC at 3.5GHZ (8 GB RAM). The pack of optional parameters obtained through several tests is as follows:

$$\alpha = 1,\ \beta = 3,\ \rho = 0.1,\ \mathrm{Q} = 100,\ \mathrm{e} = 3.$$

For more information regarding the provided examples visit:
`http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`.

As the results in Table 1 indicate, ACO has not been able to find better solutions in any of the fourteen examples. However, NMACO has been able to find better solutions through using some modifications and has come up with the best solution in 6 examples.

Table 1. Comparison of algorithms between ACO and NMACO

Instance	n	m	tbest	u	NMACO	ACO
Pr76	76	5	10	20	157413	158425
Pr152	152	5	10	40	127083	127993
Pr226	226	5	10	50	167239	168631
Pr299	299	5	10	70	81261	82871
Pr439	439	5	10	100	160298	164941
Pr1002	1002	5	10	220	379042	384901

In Table 2, we compared the efficiency and performance of NMACO with the modified genetic Algorithm (MGA) [31], sweep algorithm and elite ant system (SA+EAS) [29], and Modified Ant Colony Algorithm (MACA) [32]. In this table, for each algorithm the average, AS, and the best of obtained solutions, BS, over 10 runs are reported. This table shows that the proposed algorithm escapes from the local optimum points and converges to the best solution. Among the 6 examples, NMACO has improved the solutions in 5 examples and has been able to find the high quality for the Pr76. It has been able to find better solutions in all 6 examples except Pr76. Moreoevr, these new best known solutions are highlighted in bold in Table 2. The results of comparison with other algorithms show that the proposed algorithm obtains worse BS than the SA+EAS in Pr152, which are not large scale problems compared to other instances in Table 2and it gains better solutions than the SA+EAS in following scale problems such as Pr76, Pr226, Pr229, Pr439 and Pr1002 both in AS and BS. Furthermore, the results indicate that although the MGA produces an almost equal BS with the proposed algorithm for Pr76, this algorithm cannot maintain competition in other five examples and the NMACO yields better AS and BS than the MGA for other instances including Pr152, Pr226, Pr299, Pr439 and Pr1002. Results indicate that MACO has produced better solutions in most of the instances than the MGA. Computational experiment has shown that in general the NMACO produces better results compared to the existing solution methods for MTSP and the MGA is considered to be the weakest algorithm among the 4 presented algorithms.

Table 2. Comparison of the NMACO with other metaheuristic algorithms

Instance	n	M	u	NMACO		SA+EAS		MACO	
				BS	AS	BS	AS	BS	AS
Pr76	76	5	20	157413	157499	157482	157511	178597	180690
Pr152	152	5	40	127781	127988	127755	128004	130953	136341
Pr226	226	5	50	167239	167821	167655	168043	167646	170877
Pr299	299	5	70	81261	82012	81922	82099	82106	83845
Pr439	439	5	100	160298	161687	161698	162602	161955	165035
Pr1002	1002	5	220	379042	381108	379865	381614	382198	387205

Instance	n	M	u	NMACO		MGA		BKS	
				BS	AS	BS	AS	BS	AS
Pr76	76	5	20	157413	157499	157444	160574	157444	157511
Pr152	152	5	40	127781	127988	127839	133337	127755	128004
Pr226	226	5	50	167239	167821	166827	178501	167646	168043
Pr299	299	5	70	81261	82012	82176	85796	81922	82099
Pr439	439	5	100	160298	161687	173839	183698	161698	162602
Pr1002	1002	5	220	379042	381108	427269	459179	379865	381614

We applied the algorithm to the twenty one instances from the second dataset in [33]. These instances including depot are with 51, 100, 150 cities. In Table 3, the experimental 21 instances including MTSP-51, MTSP-100-I, MTSP-100-II, MTSP-150-I and MTSP-150-II are summarized. In Table 3, the best, average and the worst results obtained from calculations by the NMACO are compared with average results of other algorithms including genetic algorithm (GA) [33], grouping genetic algorithm (GGA) [34], new grouping genetic algorithm (NGGA) [32] and classic ACO. From Table 3 we conclude that the proposed method is efficient for solving the MTSP instances. In more detail, the NMACO not only found the best known solutions for all 3 examples including 5, 6 and 18 but also could improve the BKS for eleven examples. However, in other instances, the proposed algorithm finds very similar solutions to the BKS, i.e. the percentage deviation (Gap) is about as high as -1.5%. On the whole, the average Gap for the proposed algorithm is 0.82%. Gap is computed by formula (8) where$c(s^{**})$is the solution found by the proposed algorithm for a given instance, and $c(s^{*})$is the overall BKS for the same instance on the Web. The Gap is a simple criterion to measure the efficiency and the quality of an algorithm and is used to compute the relative average of the percentage deviation of its solution from the BKS on specific benchmark instances. Figure 6 shows the Gap of the best, average and the worst results of our algorithm.

$$Gap = \frac{c(s^{**}) - c(s^{*})}{c(s^{*})} \times 100 \tag{8}$$

Based on the results of comparison between NMACO with other metaheuristic algorithms in Table 3, we conclude that the NMACO gains better solutions than the GGA, NGGA and ACO in all of the 21 instances. Compared with GA, the proposed algorithm obtains better 11 solutions including 1, 2, 7, 10, 11, 14, 15, 16, 19, 20 and 21 than GA. Besides, the NMACO yields worse solutions than the GA in 3, 4, 8, 9, 12, 13 and 17. In other words, the performance of the proposed algorithm is better in reaching the sub-optimal solution than the GA.

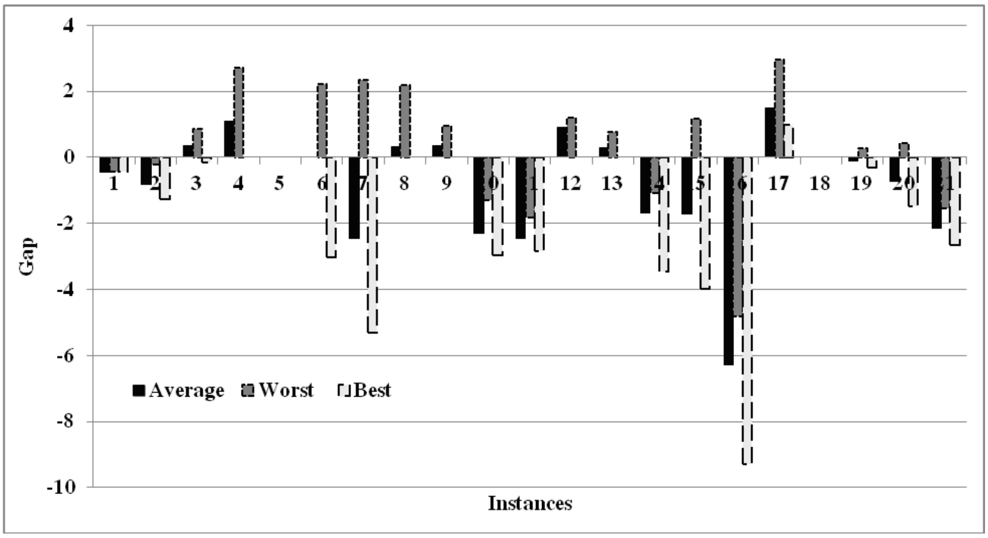

Fig. 6. Gap between average, worst and best results of the NMACO.

Table 3. Comparison of algorithms for standard problems of MTSP

Instance	Name	n	m	GGA	NGGA	GA	ACO	NMACO			BKS
								Average	Worst	Best	
MTSP-51	1	51	3	924	543	449	449	447	447	447	449
	2	51	5	882	586	479	481	475	478	473	479
	3	51	10	1001	723	584	592	586	589	583	584
MTSP-100-I	4	100	3	79347	-	22100	22492	22342	22700	22100	22100
	5	100	5	70871	-	23398	23791	23398	23398	23398	23398
	6	100	10	89778	-	28356	28701	28356	28982	27501	28356
	7	100	20	137805	-	41554	40992	40534	42534	39342	41554
MTSP-100-II	8	100	3	-	26653	22051	22241	22123	22532	22051	22051
	9	100	5	-	30408	23678	23901	23759	23902	23678	23678
	10	100	10	-	31227	28488	27981	27830	28121	27643	28488
	11	100	20	-	54700	40892	39991	39882	40149	39731	40892
MTSP-150-I	12	150	3	33888	-	6632	6699	6692	6711	6632	6632
	13	150	5	26851	-	6751	6823	6771	6802	6751	6751
	14	150	10	37771	-	7885	7788	7753	7799	7612	7885
	15	150	20	43699	-	10399	10299	10221	10521	9984	10399
	16	150	30	52564	-	14929	14137	13989	14210	13543	14929
MTSP-150-II	17	150	3	-	47418	38434	39451	39006	39573	38811	38434
	18	150	5	-	49947	39962	40671	39962	39962	39962	39962
	19	150	10	-	54958	44274	44889	44225	44399	44139	44274
	20	150	20	-	73934	56412	56619	56001	56659	55578	56412
	21	150	30	-	99547	72783	71858	71223	71658	70854	72783

6. Conclusion and Future Works

In this paper, a new modified version of ACO called NMACO which employs several effective modifications in order to solve the MTSP was presented. The modifications improved the performance of the classic ACO in escaping from local optimum points and finding better solutions in comparison with the other metaheuristic algorithms. It seems that combining the NMACO with other meta-heuristic algorithms like particle swarm optimization or tabu search and making use of strong local algorithms like Lin-Kernighan algorithm can lead to obtaining better results for the MTSP. Furthermore, the proposed algorithm can be used for other combinatorial optimization problems like vehicle routing problem and the capacitated clustering problem.

References

[1] Dorigo M., *Optimization, Learning and natural algorithms*, Ph.D Thesis, Dip. Electtronica e Informazion, Politecnico di Milano Italy, 1992.

[2] Sayyah M., Larki H., Yousefikhoshbakht M., *Solving the Vehicle Routing Problem with Simultaneous Pickup and Delivery by an Effective Ant Colony Optimization*, Transportation Research Journal, **1**, pp. 1–12, 2013.

[3] Dorigo M., Maniezzo V., Colorni A., *The ant system: optimization by a colony of cooperating agents*, IEEE Transactions on Systems, Man, and Cybernetics–Part B, **26**(1), pp. 29–41, 1996.

[4] Gambardella L.M., Dorigo M., *Solving symmetric and asymmetric TSPs by ant colonies*, Proceedings of 1996 IEEE international conference on evolutionary computation, Nagoya, Japan, pp. 622–627, 1996.

[5] Bullnheimer B., Hartl R.F., Strauss C., *A new rank based version of the ant system – A computational study*, Central European Journal for Operations Research and Economics, 7(1), pp. 25–38, 1997.

[6] Stutzle T., Hoos H.H., MAX-MIN ant system, Future Generation Computer System, pp. 889–914, 2000.

[7] Wang X., Regan A.C., "Local truckload pickup and delivery with hard time window constraints", Transportation Research Part B, 36, 2002, pp. 97-112.

[8] Kim K.H., Park Y.M., *A crane scheduling method for port container terminals*, European Journal of Operational Research, **156**, 2004, pp. 752–768.

[9] Rathinam S., Sengupta R., *Lower and Upper Bounds for a Symmetric Multiple Depots, Multiple Travelling Salesman Problem*, Research Report, UCB-ITS-RR-2006-2, 2006.

[10] Sariel S., Balch T., *Real Time Auction Based Allocation of Tasks for Multi-Robot Exploration Problem in Dynamic Environments*, The 20th National Conference on Artificial Intelligence (AAAI-05) Pittsburgh, Pennsylvania, 2005.

[11] Laporte G., Nobert Y.A., *Cutting planes algorithm for the m-salesmen problem*, Journal of the Operational Research Society, **31**, 1980, pp. 1017–1023.

[12] Ai A., Kennington J.L., *Exact solution of multiple traveling salesman problems*, Discrete Applied Mathematics, **13**, 1986, pp. 259–276.

[13] Yadlapalli S., Malik W.A., Darbha S., Pachter M., *A Lagrangian-based algorithm for a Multiple Depot, Multiple Traveling Salesmen Problem*, Nonlinear Analysis: Real World Applications, **10**(4), 2009, pp. 1990–1999.

[14] Bektas T., *The multiple traveling salesman problem: an over view of formulations and solution procedures*, Omega, **34**, 2006, pp. 209–219.

[15] Sveska J.A., Huckfeldt V.E., *Computational experience with an m-salesman traveling salesman algorithm*, Management Science, 1973, **19**, pp. 790–799.

[16] Bellmore M., Hong S., *Transformation of multi salesmen problem to the standard traveling salesman problem*, Journal of the Association Computer Machinery, 1974, **21**, pp. 500–504.

[17] Russell R.A., *An effective heuristic for the m-tour traveling salesman problem with some side conditions*, Operations Research, **25**(3), 1977, pp. 517–524.

[18] Potvin J., Lapalme G., Rousseau J., *A generalized k-opt exchange procedure for the MTSP*, Information Systems and Operations Research, **27**(4), 1989, pp. 474–481.

[19] Malik W., Rathinam S., Darbha S., *An approximation algorithm for a symmetric Generalized Multiple Depot, Multiple Travelling Salesman Problem*, Operations Research Letters, **35**(6), 2007, pp. 747–753.

[20] Vakhutinsky I.A., Golden L.B., *Solving vehicle routing problems using elastic net*, Proceedings of the IEEE international conference on neural network, 1994, pp. 4535–4540.

[21] Ryan J.L., Bailey T.G., Moore J.T., Carlton W.B., *Reactive Tabu search in unmanned aerial reconnaissance simulations*, Proceeding of the 1998 winter simulation conference, 1998, pp. 873–882.

[22] Thompson P.M., Psaraftis H.N., *Cyclic transfer algorithms for mutivehicle routing and scheduling problems*, Operations Research, Vol. **41**, No. 5, September-October 1993, pp. 935–946.

[23] Somhom S., Modares A., Enkawa T., *Competition-based neural network for the multiple travelling salesmen problem with minmax objective*, Computers & Operations Research, **26**, 1999, pp. 395–407.

[24] Jacobson S.H., McLay L.A., Hall S.N., Henderson D., Vaughan D.E., *Optimal search strategies using simultaneous generalized hill climbing algorithms*, Mathematical and Computer Modelling, **43**, 2006, pp. 1061–1073.

[25] Tang L., Liu J., *Multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex*, European Journal of Operational Research , 2000, **24**, pp. 267–282.

[26] Brown E.C., Ragsdale C.T., Carter A.E., *A grouping genetic algorithm for the multiple traveling salesperson problem*, International Journal of Information Technology & Decision Making, Vol. **6**, No. 2, 2007, pp. 333–347.

[27] Carter A.E., Ragsdale C.T., *A new approach to solving the multiple traveling salesperson problem using genetic algorithms*, European Journal of Operational Research, **175**, 2006, pp. 246–257.

[28] Ghafurian S., Javadian N., *An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems.*

[29] YousefiKhoshbakht M., Sedighpour M., *A Combination of Sweep Algorithm and Elite Ant Colony Optimization for Solving the Multiple Traveling Salesman Problem*, Proceedings of the Romanian Academy A, Vol. **13**, No. 4, 2012, pp. 295–302.

[30] Liaw C.F., *A hybrid genetic algorithm for the open shop scheduling problem*, European Journal of Operational Research, **124**, 2000, pp. 28–42.

[31] Tang T., Liu J., *Multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex*, European Journal of Operational Research, **24**, pp. 267–282, 2000.

[32] Junjie P., Dingwei W., *An ant colony optimization algorithm for multiple traveling salesman Problem*, ICICIC '06: Proceedings of the First International Conference on Innovative Computing, Information and Control, pp. 210–213, Washington, DC, USA, 2006. IEEE Computer Society.

[33] Gilbert K.C., Hofstra R.B., *A new multiperiod multiple traveling salesman problem with heuristic and application to a scheduling problem*, Decision Sciences, **23**, 1992, pp. 250–259.

[34] Angel R.D., Caudle W.L., Noonan R., Whinston A., *Computer assisted school bus scheduling*, Management Science, **18**, 1972, pp. 279–288.