Journal of Industrial and Systems Engineering
Vol. 9, No. 3, pp 57-69
Summer (July) 2016

JISE

A hybrid modified meta-heuristic algorithm for solving the traveling salesman problem

Hassan Zarei[1], Majid YousefiKhoshbakht[2]*,Esmaile Khorram[3]

[1]*Department of Mathematics, Payame Noor University, Tehran, Iran*
[2]*Young Researchers & Elite Club, Hamedan Branch, Islamic Azad University, Hamedan, Iran*
[3]*Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran*

zarei2003@gmail.com , khoshbakht@iauh.ac.ir, eskhor@aut.ac.ir

Abstract

The traveling salesman problem (TSP) is one of the most important combinational optimization problems that has nowadays received much attention because of its practical applications in industrial and service problems. In this paper, a hybrid two-phase meta-heuristic algorithm called MACSGA used forsolving the TSPis presented. At the first stage, the TSP is solved by themodified ant colony system (MACS) in each iteration, and at the second stage, the modified genetic algorithm (GA) and 2-opt local searchare used for improving the solutions of the ants for that iteration. This process avoids the premature convergence and makes better solutions. Computational results on several standard instances of TSP show the efficiency of theproposedalgorithm comparedwith the GA, ant colony optimization and other meta-heuristic algorithms.

Keywords: Genetic Algorithm, Ant Colony system, Traveling Salesman Problem, Premature Convergence

1- Introduction

One of the most studied problems in combinatorial optimization is the traveling salesman problem (TSP) and its generalizations such as the multiple traveling salesmen problem (MTSP) and the vehicle routing problem (VRP) (Yousefikhoshbakht, Didehvar and Rahmati, 2014). Their importance relies upon the fact that they are difficult to be solvedbut are intuitively used for modeling several real world problems.This problem can be simply described as follows.Suppose that there is one salesman who wants to visit *n* cities, and his object is finding out the shortest Hamilton cycle through which he can visit all the cities once and only once, and finally return to the start one. In practice, the basic TSPis extended with constraints, for instance, on the allowed capacity of the salesman, the length of the route, arrival, departure and service time, the time of collection and delivery of goods.

The techniques used for solving the TSP can be categorized into exact, heuristic and meta-heuristic methods. Exact approaches for solving the TSP are successfully used only for relatively small-sized problems, but they can guarantee optimality based on different techniques. These techniques use 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.

*Corresponding author.

There have been some papers such as exact exponential method (Xiao and Nagamochi, 2016), branch and bound (Battarraand Pessoa, 2014)proposing exact algorithms for solving the TSP. Although the TSP is conceptually simple, it is difficult to obtain an optimal solution. In *n*-city situation, any permutation of *n* cities yields a possible solution. As a consequence, *n*!possible tours must be evaluated in the search space. In other words, when the problem size increases, the exact methods cannot solve it efficiently in a reasonable computing time.Therefore, we have to use heuristic or meta-heuristicmethods for solving them and settle for the suboptimal solutions in a reasonable amount of time for instances with large sizes. Some of the famousheuristic algorithms are gravitational emulation search (Balachandar and Kannan, 2007), 2-opt, 3-opt heuristics (Lin, 1965) and Lin-Kernighan (Karapetyanand Gutin, 2011).

In the last 30 years, a new kind of algorithm called *metaheuristic* has emerged which basically tries to combine basic heuristic methods in higher level frameworks aimed at efficiently and effectively exploring a search space.In general; it is absolutely essentialto use meta-heuristics algorithms in order to solve complex optimization problems. Since the meta-heuristic approaches are very efficient for escaping from local optimum, they are one of the best algorithms for solving combinatorial optimization problems. That is why the recent publications are more based on meta-heuristic approaches such as genetic algorithm (GA) (Sedighpour, YousefiKhoshbakht and MahmoodiDarani, 2011), african Buffalo Optimization (Odili and MohmadKahar, 2016), ant system (AS) (YousefiKhoshbakht, Didehvar and Rahmati, 2013), particle swarm optimization (Anantathanavit and Munlin, 2015) and imperialist competitive algorithm (ICA)(YousefiKhoshbakht and Sedighpour, 2012).

Recently, many researchers have found that the employment of hybridization in optimization problems can improve the quality of problem solving in comparison with heuristics and meta-heuristics. ICA with tabu search(Ahmadvand, YousefiKhoshbakht and MahmoodiDarani, 2012), combination of particle swarm optimization, greedy randomized adaptive search procedure (GRASP) and expanding neighborhood search strategy (Marinakis and Marinaki, 2010), elite ant system and local search (YousefiKhoshbakht, Sedighpour and Mahmoodabadi, 2011), variable neighborhood descent search and GRASP (Hernandez-Perez, Rodríguez-Martín and Salazar-Gonzalez, 2009) and so on are examples of hybrid algorithms.

The TSP is an important NP-hard optimization problem that arises in several applications including the computer wiring, sequencing job, designing hardware devices and radio electronic devices, communications, architecture of computational networks, etc. This means that a polynomial time algorithm does not exist for it and the computational attempt required to solve this problem increases exponentially with the size of the problem. Consequently, these kinds of problems are often solved with metaheuristic techniques. Besides, because hybrid algorithms have greater ability for finding an optimal solution, they have been considered by researchers and scientists in recent years. For these reasons, in this paper, a hybrid modified two-phase meta-heuristic algorithm called MACSGA is proposed for solving the TSP. In this algorithm, several modifications including a new state transition rule, a modified Order crossover, an efficient mutation and several effective local search techniques are used in order to achieve better solutions. The algorithm was applied on a set of benchmark instances from TSPLIB and is able to produce very satisfactory results which are very close to best-known solutions (BKS) for most testedproblems.

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

2- Mathematical model

The TSPis one of the most famous combinatorial optimization problems that has nowadays received much attention because of its practical applications in industrial and service problems.The TSPis defined in investigations as follows:

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, we can replace the lack of any edge with an edge that has an infinite size.In this problem, if c_{ij} represents the distance from node *i* to node *j*, and by introducing variables x_{ij} to represent the tour of the salesman from node *i* to node *j*, one of the common integer programming formulations for the TSP can be written as:

$$\min \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ij} \quad (1)$$

subject to

$$\sum_{i=0}^{n} x_{ij} = 1 \qquad (j = 0,...,n) \quad (2)$$

$$\sum_{j=0}^{n} x_{ij} = 1 \qquad (i = 0,...,n) \quad (3)$$

$$\sum_{i \in S} \sum_{j \in N-S} x_{ij} \geq 1 \quad (S \subset N = \{1,...,n\}, |S| \geq 2) \quad (4)$$

$$\sum_{i \in N-S} \sum_{j \in S} x_{ij} \geq 1 \ (S \subset N = \{1,...,n\}, |S| \geq 2) \quad (5)$$

$$x_{ij} \in \{0,1\} \quad (6)$$

In this formulation, the objective function (1) is simple to minimize the total distance traveled in a tour. Constraint set (2) ensures that the salesman arrives once at each city. Constraint set (3) ensures that the salesman leaves each city once. Constraint set (4) and (5) are to avoid the presence of sub-tour. Finally, Constraint set (6) defines binary conditions on the variables. Generally, the TSP formulated is known as the Euclidean TSP, in which the distance matrix $C = c_{ij}$ is expected to be symmetric, and to satisfy the triangle inequality.

3- Proposed algorithm

In this section, first the ACOand GAare explained and thenour algorithm is analyzed in more detail.

3-1- Ant colony optimization

The ACO is an iterative stochastic search technique whichwas inspired by the food foraging behavior of real ant colonies. Furthermore, this algorithmis used for solving combinatorial optimization problems that do not have a known efficient algorithm. 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 accumulate more pheromone. Therefore, pheromone intensity helps ant to identify shorter paths to the food source. Consequently, a positive feedback and autocatalytic mechanism based on pheromone is formed.This technique is based on AS introduced by Dorigo et al. who used it to solve TSP problems (Dorigo, Maniezzo and 1996). It led the simple intelligence body of the artificial ant colony into the optimization algorithms to solve the problems more efficiently.

Since the first ACO was proposed in 1991, this algorithmhas attracted the attention of an increasing number of researchers and several ACO algorithms have been proposed in the literature. The first ACO algorithm called AS was applied to the TSP. Besides, many improved algorithms have also been tested on the TSP. Those improvements 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 among the individuals from a colony of ants, based on an analogy with the trails of pheromone.

3-2- Genetic algorithm

Meta-heuristic algorithms such as memetic algorithms simulated annealing, particle swarm optimization, Tabu search and soon 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. The GA is one of the oldest meta-heuristicalgorithms thathasreceived much attention from researchers and scientists. This algorithm is adaptive searching procedure for solving 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. In each iteration, 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 compared with the old ones. Later, a mutation operation is applied to the newly generated solutions based on a mutation probability. Repeat these operations (selection, crossover, and mutation) 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 reachedto *n*or the solution quality cannot be easily improved.

3-3- The proposed algorithm

Although the various versions of ACO have some advantages like enjoying distributed calculation and being robust, this algorithm is still one of the best methods to solve combinatorial optimization problems more effectively considering time and quantitycompared to other meta-heuristic algorithms. However, there are some disadvantages such as long interval in operation and easily-occurring stagnation in ACO. In order to overcome these shortcomings, many researchers have improved the ACO (Tsai and Tsai, 2002).In the following subsections, transition probabilities, crossover and mutation of GA, local search and pheromone updating rulesare presented respectively.

3-3-1 State transition rule

At each iteration of the modified ACS, each ant builds a solution of the TSP step by step. At each step the ant makes a move in order to complete the actual partial solution by choosing between elements of a set of A_k expansion states through following a probability function. The probability of ant *k* which moves from city *i* to city *j* which has not been visited yet is presented as follows:

$$P_{ij}^{k}(t)=\begin{cases}1 & \text{if } q \le q_0 \ \&\ j = j^{*}\\ 0 & \text{if } q \le q_0 \ \&\ j \ne j^{*}\\ \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)\gamma_{ij}^{\lambda}(t)}{\sum_{r\in J_i^k}\tau_{ir}^{\alpha}(t)\eta_{ir}^{\beta}(t)\gamma_{ij}^{\lambda}(t)} & \text{Otherwise}\end{cases} \tag{7}$$

Where

$j^{*}=\arg\max_{r\in J_i^k}\tau_{ir}^{\alpha}(t)\eta_{ir}^{\beta}(t)\gamma_{ij}^{\lambda}(t)$ identifies the unvisited node in J_i^k that maximizes $P_{ij}^{k}(t)$.

$\tau_{ij}(t)$: The value of phermone on the arc (i,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).

$\gamma_{ij}(t)$: The value of savings algorithm computed by $\gamma_{i,j} = c_{i0} + c_{0j} - c_{ij}$.

q: A uniformly distributed random number between 0 to 1.

q_0: A parameter belonging to [0,1]: the smaller q_0, the higher the probability to make a random choice.

α, β, λ: The controling parameters by user.

3-3-2- Crossover of GA

Genetic algorithms were inspired by the evolution process, observed in nature. The system encodes theparameters of a solution into a chromosome, where the basic element of a chromosome is the gene. In the proposed algorithm, three operations are performed, i.e., selection operations, crossover operations and mutation operations in order to search the near optimal solution. After building *n* solutions of the TSP in each iteration by ACS, they are considered as a group of initial population for GA. In the proposed GA, only one kind of feasible solutions is commonly employed for solving the TSP. Therefore, a bisection array shown in Figure 1 is used (where *n = 11* and fitness function = *29.34*). In this array, the visited nodes are ordered from left to right in the first section and the value of fitness function of this feasible solution is shown in the second section. Therefore, the selection operation selects those feasible solution built by ACS. The fitness degree of a chromosome is evaluated by Euclidian distance between nodes of this produced cycle.

1	9	10	3	11	5	4	2	6	7	8	29.34

Fig. 1. A chromosome in the proposed algorithm

Then crossover and mutation operationsare applied to them for improving the obtained solution of ACS.Order is still one of the best crossovers in terms of quality and speed. Therefore, this method that is simple to implement has been considered hereand the modified Order crossover is proposed based on this crossover. A randomly chosen crossover point divides the parent strings into left and right substrings. The right substrings of the parents are selected and the same arrangement of genes in another chromosome is found. Finally, the selected genes in each chromosome based on their arrangement in another chromosome are replaced. In the proposed crossover, after selecting a randomized number of genes from each chromosome, the process is the same as the Order crossover. The only difference is that instead of selecting all the positions to the right of the randomly chosen crossover point, several random positions in a parent, are selected (Figure2). Clearly this method allows only the generation of valid strings.

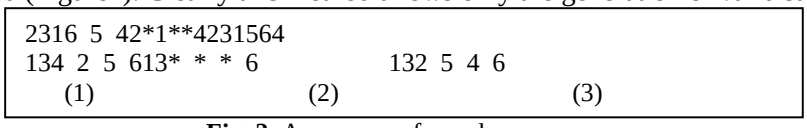

Fig. 2. A crossover for a chromosome

3-3-3- Mutation of GA

The aim of the mutation operation is to prevent the chromosomes in the new population to fall into a local minimum. It should be noted that the system performs the mutation operation after the crossover operation is performed and some chromosomes perform the mutation operation in the evolution process. For achieving this purpose, a random number between 0 and 1 generated by the system is smaller than a predefined mutation rate (MR) belonged to (0,1], then the system performs the mutation operation. Mutation method is to randomly select a gene of the selected chromosome, and then randomly generate a new value to replace it. Furthermore, the mutation operation continues until every chromosome is certain to perform the mutation operation or not. Therefore, two mutations are used in the proposed algorithm.

These operators randomly select one point in the string. Then, this node is moved to another position (Figure3 (left)) or substrings of nodes are reversed after this node (Figure3 (right)).

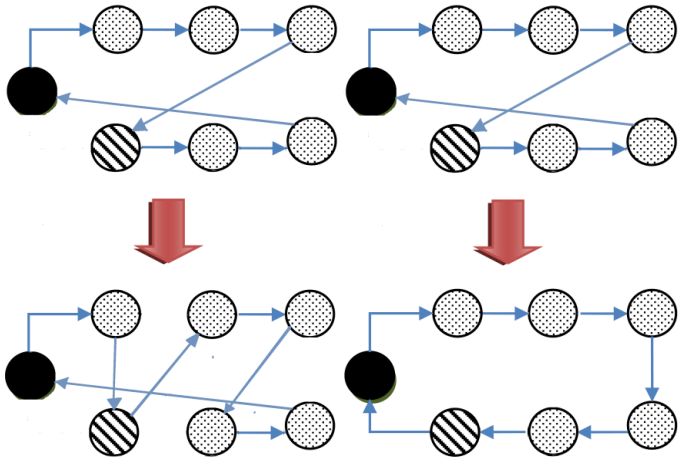

Fig. 3. Mutations (1) (right) and (2) (left)

3-3-4- Local search

The vast literature on meta-heuristics tells us that a promising approach to obtaining high-quality solutions is to couple a local search algorithm with a mechanism to generate initial solutions. ACS's definition includes the possibility of using local search as daemon actions. Daemon actions are used to implement centralized actions which cannot be performed by single ants. An example of daemon actions is the activation of a local optimization procedure.During the implementation phase, the ACS is combined with 2-opt local search procedure to obtain more improvement in the algorithm's performance. The 2-opt heuristic tries to improve a single route by replacing its two non-adjacent edges by two other edges. It should be noted that there are several routes for connecting nodes in order to produce the tour again. However, only a state that satisfies the problem's constraints is acceptable. Therefore, 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.

3-3-4- Pheromone updating rule

In order to improve future solutions, the pheromone trails of the ants must be updated to reflect the ant's performance and the quality of the solutions found. The pheromone updating formula was meant 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. This updating is a key element to the adaptive learning technique of ACS and helps to ensure the improvement of subsequent solutions. Unlike the AS, the pheromone of all edges belonging to the route chosen by ants is not updated in the proposed algorithm. In other words, the pheromone updating of ACSGA includes only global updating rules. What distinguishes the proposed algorithm from the other algorithms is the fact that only arcs belonging to the best known solution and the best solution in current iteration are updated. In other words, when *n* solutions for the problem are producedafter local search algorithms in the current iteration, the pheromone level on the edges of thebest known solution found up to now and the current best solution are updated by formula (8).This rule is intended to provide a greater amount of pheromone on the paths of the two best solutions, thus intensifying the search around these solutions.

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \rho(1/C_b) \qquad if\{edge(i,j) \in T_b\} \tag{8}$$

Where

C_b : The cost of the best known tour or the current best solution.

T_b : The best know tour or the current best solution.

ρ : A parameter in the range [0, 1] that regulates the reduction of pheromone on the edges.

A pseudo code of the proposed algorithm for the TSP is presented in Figure4.

```
S:= none;                          // S is population of solutions
//
s* is the random solution;   // s* is the best solution found
//
f* is value of s*;
initialize pheromone trails;
repeat // main cycle //
    for i := 1 to n do
    begin
        Construct a solution si by using formula 7; S=S
        ∪ si
    end;
        apply Genetic Algorithm to S and generate S*;
        S:=none;
        for i:=1 to length( S* ) do
        begin
            if f(si*) < f* then
            begin
                f* := f(si*); s* = si*;
            end // save the best so far solution //
            apply 2-opt local search to s*
        end
        global update pheromone trails
until no improvement finding for 20 iterations
show s* and f*
```

Fig.4. Paradigm of proposed algorithm

After the global updating, the system evaluates the fitness value of the best known solution. If the algorithm has not improved the best solution for 20 iterations, the termination condition is met and the algorithm is stopped. Otherwise, go to state transition rule step.

4- Experiments and computational results

The proposed algorithm is coded in C and implemented on a Pentium 4, 3 GHZ (512 MB RAM) PC with windows XP. Because the MACSGA is a metaheuristic algorithm, the solutions produced by the proposed algorithm were dependent on the seed used to generate the sequence of pseudo-random numbers and on the different values of the search parameters of the algorithm. The ranges of seven parameters were set in the table 1. In this table, all of the parameter value have been determined on the first instance i.e. Eil51 by the numerical experiments. The parameter setting procedure is necessary to reach the best balance among the quality of the solutions obtained and the required computational attempt. The algorithm with each parameter combination for this instance was tested10 times. Then to determine the value of parameters several alternative values for each parameter were tested while all the others were

held constant, and the ones that were selected gave the best computational results concerning both the quality of the solution and the computational time needed to achieve this solution. The results confirm that our parameter setting worked well. It is also possible that better solutions could exist.

Table 1. Parameter setting for MACSGA

Parameters	Candidate Value	The best value
Power of ant's visiblity measure on arc (i,j)	1, 2, 3, 4, 5	2
Power value of pheromone releaseed on arc (i,j)	1, 2, 3, 4, 5	3
The constant coefficient of evaporation of pheromone	0.05, 0.1, 0.15, 0.2, 0.25	0.15
Power value of savings algorithm on arc (i,j)	1, 2, 3, 4, 5	3
The size of the population	n	n
The rate of mutation	0.1, 0.15, 0.2, 0.25, 0.3	0.2
The number of iterations after which the metaheuristic algorithm terminate if fails to reach a new best solution	5, 10, 15, 20, 25	20

Table 2 shows the results of proposed algorithm for the 22TSP benchmark instances. The selected test problems are Euclidean TSP instances where the sizes of the problems range from 24 to 400. The first and second columns in Table 3 specify number of instance and its name. Furthermore, size of these instances is shown in the third column. Columns 4-6 show the best value result of MACSGA(BVR), the worst value result of MACSGA(WVR), the average value of MACSGA(AV) over the ten runs for each problem. Finally, the best known solutions (BKS) of these instancesin the literature are presented in the last column.

Table 2. Results of MACSGAfor the TSP

	Instance	n	BVR	WVR	AV	BKS
1	GR24	24	1272	1272	1272	1272
2	Bayg29	29	1610	1610	1610	1610
3	Pr76	76	108363	109541	108553	108159
4	Rat99	99	1222	1308	1262	1211
5	Rd100	100	7995	8152	8056	7910
6	Pr107	107	44385	45124	44821	44303
7	Pr124	124	59124	60741	59854	59030
8	Bier127	127	118282	120541	119124	118282
9	Ch130	130	6139	6299	6195	6110
10	Pr136	136	97132	98006	97645	96772
11	Pr144	144	58746	59745	58889	58537
12	Ch150	150	6559	6698	6614	6528
13	Pr152	152	73682	74923	74236	73682
14	Rat195	195	2339	2485	2417	2323
15	D198	198	15958	16798	16331	15780
16	Ts225	225	127895	129541	128293	126643
17	Pr226	226	81021	83879	81598	80369
18	Gil262	262	2393	2578	2488	2378
19	Pr264	264	49842	52548	51158	49135
20	A280	280	2606	2895	2797	2579
21	Pr299	299	48298	50364	49739	48191
22	Rd400	400	15529	16984	16658	15281

The relative deviations calculated by formula (9), where value of the algorithm denotes the cost of the optimum found by proposed algorithm, and BKS is the cost of the optimal solution. A simple criterion to measure the efficiency and the quality of an algorithm is to compute the relative average of percentage deviation of its solution from the BKS on specific benchmark instances. Figure 5 shows the percentage deviation of the best, worst and average solutions of the algorithm. It is concluded that the proposed method obtains high quality solutions from the BKS. As this figure indicates, maximum relative error and average relative error for 26 test problems of the BVR are 2.17% and 0.63%, respectively. In more detail, the BVR version ofMACSGAhas found the best known solutions for 5 examples including 1, 2, 10, 16 and 19. However, in other instances, the BVR finds nearly the BKS, i.e. the gap is about as high as 2%.

$$\frac{\text{Value of the algorithm} - \text{Value of the BKS}}{\text{Value of the BKS}} \times 100 \tag{9}$$

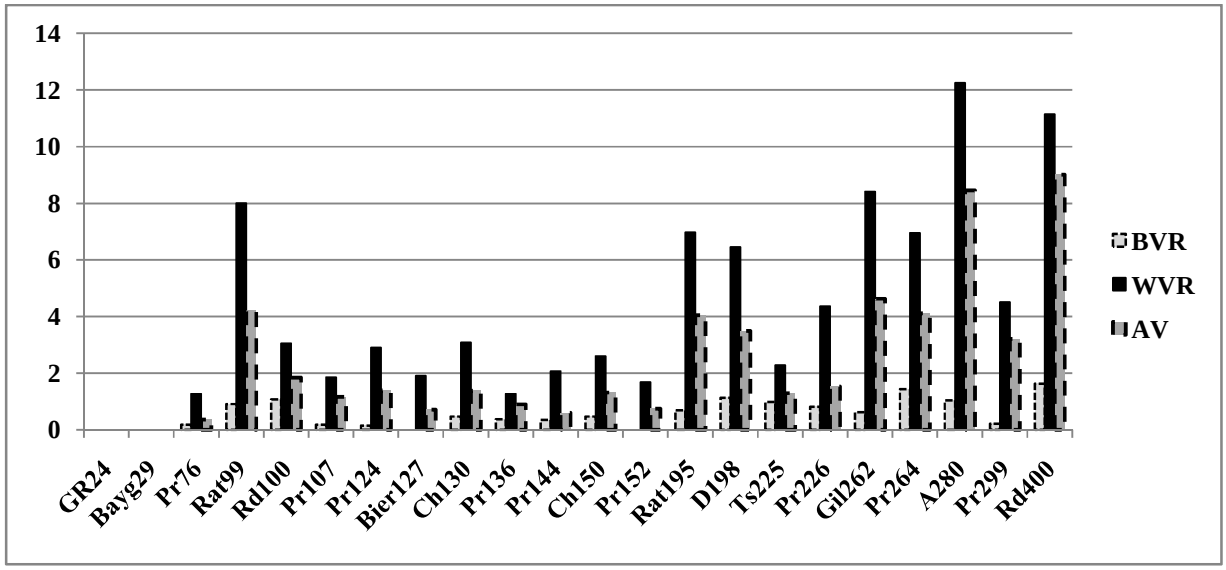

Fig.5. Comparison Gap of the best, worst and average solutions of the proposed algorithm

A computational experiment has been conducted in Table 3 to compare the performance of the proposed algorithm with some of the best techniques designed including GA, ACS, PSO (Zhong, Zhang and Chen, 2007) and Bee Colony Optimization (BCO) (Wong, Low and Chong, 2008) for TSP. We executed the algorithm on some of the well-known problem instances from one dataset. The sets of data used for the experiment are TSP instances available on the TSP library (http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/). The selected test problems are Euclidean TSP instances where the sizes of the problems range from 24 to 200. The first, second and third columns in Table 3 specify the number, name of instance and its size referenced. Moreover, the fourth, fifth and sixth, seventh and eighth columns show the 5 meta-heuristic algorithms. Additionally, in order to recognize the performance of the method, the best known solutions (BKS) in the literature and also on the web, are presented in the last column.

The results show that the MACSGA has the ability to escape from local optimum and find the best solutions for all of the instances.The results of this comparison show that the proposed algorithm gains equal solutions to the GA in GR24 and Bayg29, and it gains better solutions than the GA in other problems from Gr48 to KroB100.Furthermore, the results indicate that although the ACS gives an equal solution to the proposed algorithm for 5 instances, this algorithm cannot gain optimal solutions for others and yields worsesolutions than the proposed algorithm. The computational experiments also show that in

general the proposed algorithm gives better results compared to other two algorithms including PSO and BCO algorithms in terms of the solution's quality. As a result, the proposed algorithm yields better solutions than the GA, PSO, ACS, and BCO.

Table 3. Comparison of algorithms for standard problems of TSP

	Instance	n	GA	ACS	PSO	BCO	MACSGA	BKS
1	GR24	24	1272	1272	-	-	1272	1272
2	Bayg29	29	1610	1610	-	-	1610	1610
3	GR48	48	5037	5046	-	-	5046	5046
4	ATT48	48	10643	10628	-	10661	10628	10628
5	Eil51	51	429	427	427	428	426	426
6	Berlin52	52	7548	7542	7542	-	7542	7542
7	ST70	70	688	679	-	-	675	675
8	Eil76	76	549	542	540	539	538	538
9	KroA100	100	21540	21309	21296	21763	21282	21282
10	KroB100	100	22431	22183	-	22637	22141	22141
11	KroC100	100	22993	20787	-	20853	20749	20749
12	KroD100	100	21591	21341	-	21643	21294	21294
13	KroE100	100	22198	22109	-	22450	22068	22068
14	Eil101	101	643	636	-	635	629	629
15	Lin105	105	14703	14534	-	15288	14379	14379
16	Bier127	127	-	-	-	-	118282	118282
17	KroA150	150	27054	26749	-	27858	26524	26524
18	KroB150	150	26659	26431	-	26535	26130	26130
19	KroA200	200	30276	29762	29563	29961	29368	29368
20	KroB200	200	31980	29653	-	30350	29437	29437

In addition, four of the best solutions searched by the proposed method in Table 1 are presented in Figure 6. In this figure, the horizontal and vertical axes show the two-dimensional coordinates x and y.It should be noted that in the all examples presented, the proposed algorithm has been able to find the optimum solution.

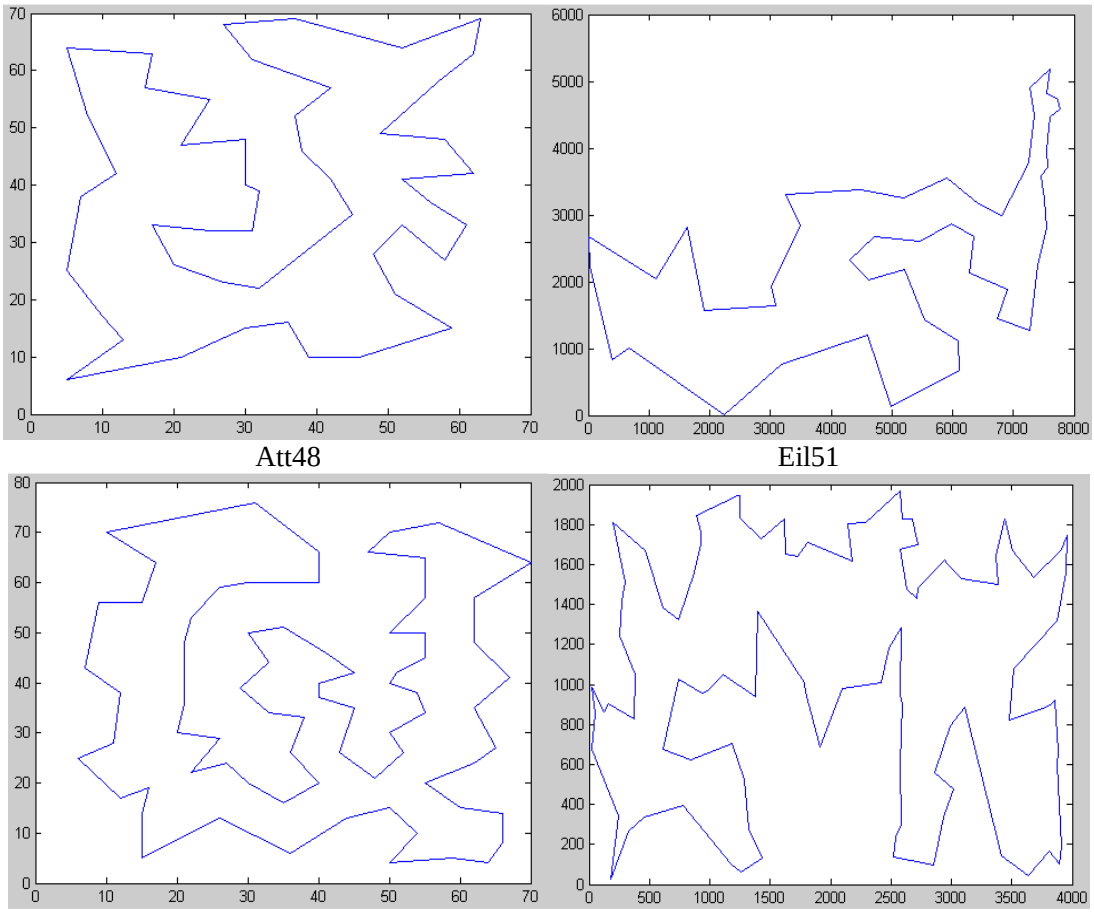

Fig.6. Best routes found by the proposed algorithm for four instances

5- Conclusions

In this paper, a hybrid algorithm combiningMACS, GA and 2-opt local search was proposed for solving the TSP. One of the main contributions of this paper was to show that the modified ant colony system can be used in hybrid synthesis with other metaheuristics for the solution of the TSP with remarkable results. The algorithm was applied on a set of benchmark instances from TSPLIB and gave very satisfactory results. Using this proposed algorithm for other versions of the TSPand also applying this method in other combinational optimization problems including the vehicle routing problem, School bus routing problem and the sequencing of jobs are suggested for future research.

6- Acknowledgement

The authors would like to acknowledge the Payame Noor university for the financial support of this work.

References

Ahmadvand, M., YousefiKhoshbakht, M., &MahmoodiDarani, N. (2012).Solving the Traveling Salesman Problem by an Efficient Hybrid Metaheuristic Algorithm.*Journal of Advances in Computer Research, 3(3),* 75-84.

Anantathanavit, M., &Munlin, M. (2015).Using K-means Radius Particle Swarm Optimization for the Travelling Salesman Problem. IETE Technical Review, 1-9.

Balachandar, S. R., & Kannan, K. (2007).Randomized gravitational emulation search algorithm for symmetrictraveling salesmanproblem. Applied Mathematics and Computation, 192(2), 413-421.

Battarra, M., & Pessoa, A. A. (2014). Subramanian, A., Uchoa, E., Exact algorithms for the traveling salesman problem with draft limits. *European Journal of Operational Research, 235(1),* 115-128.

Dorigo, M., Maniezzo, V., &Colorni, A. (1996).Ant System: Optimization by a colony of cooperating agents.*IEEE Transactions on Systems, Man, and Cybernetics—Part B, 26(1),* 29-41.

Glover, F. (1986).Future paths for integer programming and links to artificial intelligence.*Computer Operation Research, 13,* 533.549.

Hernandez-Perez, H., Rodríguez-Martín, I., & Salazar-Gonzalez, J. J. (2009).Ahybrid GRASP/VND heuristic for the one-commodity pickup-and-deliverytravelingsalesman problem.*Computers & Operations Research, 36(5),* 1639-1645.

Karapetyan, D., &Gutin, G. (2011).Lin-Kernighan Heuristic Adaptations for the GeneralizedTravelingSalesmanProblem.*European Journal of Operational Research, 208(3),* 221-232.

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

Marinakis, Y., &Marinaki, M. (2010).AHybrid Multi-Swarm Particle Swarm Optimization algorithm for the ProbabilisticTravelingSalesmanProblem.*Computers & Operations Research, 37(3),* 432-442.

Odili, J. B., &MohmadKahar, M. N. (2016).Solving the Traveling Salesman's Problem Using the African Buffalo Optimization. Computational Intelligence and Neuroscience.

Qinghong, WU.,&Jihui, Z. (1999). ACO with the characteristic of Mutation, *Computer research and development,* 240-245.

Sedighpour, M., YousefiKhoshbakht, M., &MahmoodiDarani, N. (2011). An Effective Genetic Algorithm for Solving the Multiple Traveling Salesman Problem. *Journal of optimization in Industrial Engineering, 4(8),* 73-79.

Tsai, C. F.,&Tsai.C.W. (2002).A new approach for solving large traveling salesman problem using evolution ant rules, in Neural Networks, IJCNN 2002, Proc. of the 2002 Int'l Joint Conf., Honolulu, *IEEE Press,* 540-1545.

Wong, L. P., Low, M. Y. H., & Chong, C. S. (2008). A bee colony optimization algorithm for traveling salesman problem. Modeling & Simulation, 2008.AICMS 08.*Second Asia International Conference*, 818–823.

Xiao, M., &Nagamochi, H. (2016).An Improved Exact Algorithm for TSP in Graphs of Maximum Degree 4. *Theory of Computing Systems, 58(2),* 241-272.

YousefiKhoshbakht, M., Didehvar, F., & Rahmati, F. (2013). Modification of the Ant Colony Optimization for Solving the Multiple Traveling Salesman Problem, *Romanian Journal of information science and technology, 16(1),* 65–80.

Yousefikhoshbakht, M., Didehvar, F., & Rahmati, F. (2014).Solving the heterogeneous fixed fleet open vehicle routing problem by a combined metaheuristic algorithm. *International Journal of Production Research, 52(9),* 2565-2575.

YousefiKhoshbakht, M., &Sedighpour, M. (2012).A New Imperialist Competitive Algorithm to Solve the Traveling Salesman Problem.*International Journal of Computer Mathematics, 90(7),* 1495-1505.

YousefiKhoshbakht, M., Sedighpour, M., &Mahmoodabadi, E. (2011).A Modified Elite ACO Based Avoiding Premature Convergence for Traveling Salesman Problem. *Journal of Industrial Engineering International, 7(15),* 68-75.

Zhong, W., Zhang, J., & Chen, W. (2007). A novel discrete particle swarm optimization to solve traveling salesman problem. *Evolutionary Computation,* 3283–3287.