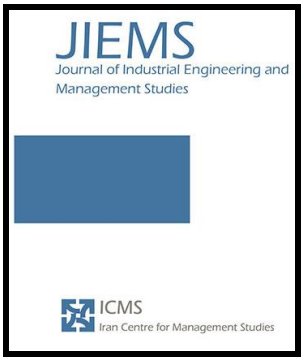
JIEMS

Journal of Industrial Engineering and Management Studies

Vol. 2, No. 2, pp. 13-25

www.jiems.icms.ac.ir



A reactive bone route algorithm for solving the traveling salesman problem

N. Mahmoodi Darani[1], A. Dolatnejad[2], M. Yousefikhoshbakht[3,*]

Abstract

The traveling salesman problem (TSP) is a well-known optimization problem in graph theory, as well as in operations research that has nowadays received much attention because of its practical applications in industrial and service problems. In this problem, a salesman starts to move from an arbitrary place called depot and after visits all of the nodes, finally comes back to the depot. The objective is to minimize the total distance traveled by the salesman. Because this problem is a non-deterministic polynomial (NP-hard) problem in nature, it requires a non-polynomial time complexity at runtime to produce a solution. Therefore, a reactive bone route algorithm called RBRA is used for solving the TSP in which several local search algorithms as an improved procedure are applied. This process avoids the premature convergence and makes better solutions. Computational results on several standard instances of TSP show the efficiency of the proposed algorithm compared to other meta-heuristic algorithms.

Keywords: Reactive Bone Route Algorithm, Traveling Salesman Problem, local search algorithms, NP-hard Problems.

Received: July 2015-28

Revised: August 2015-25

Accepted: December 2015 -10

1. Introduction

The traveling salesman problem (TSP) is one of the most studied problems in combinatorial optimization problems which is concerned with searching for the minimum Hamiltonian cycle in a network of nodes with some additional constraints (Chen, et al., 2012). In literature, this minimum Hamiltonian cycle means the closed walk that traverses every vertex once and only once in a graph traversing the minimum path in terms of the length of the edges. This closed walk starts from a fixed vertex, to which it also returns after the walk. Their importance relies upon the fact that they are difficult to be solved but are intuitively used for modeling several real world problems (Yousefikhoshbakht, et al., 2014).

* Corresponding Author.
[1] Young Researchers & Elite Club, Robatkarim Branch, Islamic Azad University, Robatkarim, Iran.
[2] Young Researchers & Elite Club, Tehran North Branch, Islamic Azad University, Tehran, Iran.
[3] Young Researchers & Elite Club, Hamedan Branch, Islamic Azad University, Hamedan, Iran.

The first instance of the TSP was from Euler in 1759 whose problem was to move a knight to every position on a chess board exactly once. The TSP first gained fame in a book written by German salesman BF Voigt in 1832 on how to be a successful traveling salesman (Michalewicz, 1994). He mentions the TSP, though not by that name, by suggesting that to cover as many locations as possible without visiting any location twice is the most important aspect of the scheduling of a tour. The origins of the TSP in mathematics are not really known all we know for certain is that it happened around 1931.

In practice, the basic TSP is 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. It should be noted that the main goal in all TSP problems is to obtain the minimal transportation cost. In another view, there are several reasons for choosing the TSP as the problem to examine the efficiency of new algorithm:

1) It is an important NP-hard optimization problem (Ahmadvand, et al., 2012) like the n-queens problem (Masehian, et al., 2014), k-sat (Jaafar and Samsudin, 2013) and maximum propositional satisfiability problem (Bachir Menai and Batouche, 2005) that arises in several applications including the computer wiring, sequencing job, designing hardware devices and radio electronic devices, communications, architecture of computational networks, etc.

2) It is a problem to which new algorithms are easily applied (Barketau and Pesch, 2015).

3) It is easily understandable, since the algorithm behavior is not obscured by too many technicalities (Yang, et al., 2015).

4) It is a standard test bed for new algorithmic ideas. A good performance of the TSP is often taken as a proof of their usefulness (Xiao and Nagamochi, 2016).

The TSP belongs to NP-hard problems. 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. These kinds of problems are regularly solved by heuristic and metaheuristic techniques. Therefore, an efficient adaptive memory based algorithm equipped with diversification and intensification mechanisms is proposed in this paper in which a reactive bone route algorithm is combined with some local search algorithms. In this work, a new solution from a component of the other solutions is produced while using new diversification and intensification mechanisms. Some test problems of TSPLIB are considered and results of RBRA are compared to the several metaheuristic algorithms. The proposed algorithm in comparison with these algorithms can provide better solutions.

In the following parts of this paper, background and literature review is presented in Section 2. In Section 3, the proposed idea is explained in details. In Section 4, the RBRA is compared with some of the metaheuristic algorithms on standard TSP problems. Finally in Section 5, the conclusions are presented.

2. Background and literature review

The TSP is considered one of the most well-known combinatorial optimization tasks and researchers have paid so much attention on the TSP for many years, while there are many problems extended from it, like the vehicle routing problem and the multiple traveling salesmen problem and so on (Figure 1). Therefore, different approaches for solving the TSP have been explored during the several decades ago. These approaches range from the use of exact optimization methods for solving small-size problems with relatively simple constraints to the use of heuristic and meta-heuristic algorithms that provide near-optimal solutions for medium and large-size problems. Exact approaches for solving the TSP can guarantee optimality based on different techniques and use some

algorithms that generate both a lower and an upper bound on the true minimum value of the problem instance. If the upper and lower bound coincide, a proof of optimality is achieved. There have been many papers proposing exact algorithms for solving the TSP. These algorithms are based on Lagrangean relaxation (Yadlapalli, et al., 2009), branch-and-cut method (Cordeau, et al., 2010), branch and bound (Carpaneto and Toth, 1980), etc.

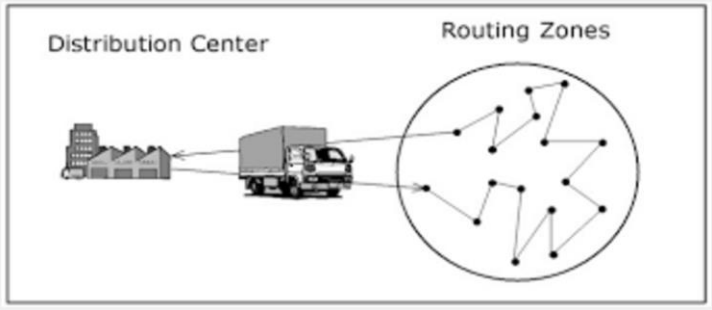


Figure 1. Relation between TSP and VRP

Although optimal solutions can be obtained using exact methods, the computational time required to solve adequately large problem instances is still prohibitive. For this reason, the focus of most researchers is given to the design of heuristic and metaheuristic approaches capable of producing high quality near optimum solutions with reasonable computational time. The proposed heuristics can be separated into three categories: tour construction heuristics, tour improvement heuristics and composite heuristics. The composite heuristics serve both purposes of the first two kinds of heuristics. Typical tour construction heuristics are fragment-growing methods whereas tour-growing heuristics are cross-products of certain selection rules (e.g. farthest, nearest, and random) and expansion rules (e.g. addition and insertion). The sweep algorithm (Gillett and Miller, 1974), partitioning approach (Karp, 1977) and saving algorithm (Clarke and Wright, 1964) are some important tour construction heuristics. On the other hand, an improvement heuristic starts with an initial solution and then provides a scheme for iteratively obtaining an improved solution until reaching stopping criteria. These algorithms for the TSP are based on simple route modifications and may operate together. For the tour improvement heuristics, variant employments of local search (Bianchi, et al., 2005), 3-opt heuristics (Lin, 1965) and Lin-Kernighan (Karapetyan and Gutin, 2011) are probably the most significant methods because they are the most frequently adopted improvement tools by tour construction heuristics and some composite heuristics.

As the TSP is NP-complete, great effort has been devoted to metaheuristics that produce a good, if not optimal, tour. Johnson and McGeoch in 2002 conclude that, for practical instances, these algorithms can provide remarkably good results in reasonable amounts of time. Before this term was widely adopted, metaheuristics were often called *modern heuristics* (Reeves, 1998). 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 gravitational emulation search (Balachandar and Kannan, 2007), neural network (Thiago, et al., 2009), ant colony optimization (ACO) (Yousefikhoshbakht, et al., 2013), african buffalo optimization (Odili and Mohmad Kahar, 2016), recurrent neural network (Tarkov, 2015), imperialist competitive algorithm (Yousefikhoshbakht and Sedighpour, 2012) and particle Swarm optimization (Anantathanavit and Munlin, 2015).

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. Since hybrid algorithms such as ant colony optimization (ACO) and beam algorithm (López-Ibáñez, et al., 2010), genetic algorithm (GA) with a local search (Créput and Koukam,

2009), ACO with Sweep algorithm (Yousefikhoshbakht and Sedighpour, 2013), threshold accepting and edge recombination (Liu, 2007), particle swarm optimization and local search (Shi, et al., 2007), variable neighborhood descent search and GRASP (Hernandez-Perez, et al., 2009) have greater ability for finding an optimal solution, they have been considered to solve complex problems. For example, (Wang, 2010) presented a hybrid algorithm in which GA, ACO and a new strategy called GSA was proposed aiming at the key link in the algorithm. This algorithm converts the genetic solution from GA into information pheromone to distribute in ACO. Furthermore, GSA takes a new matrix which is formed by the combination of the former 90% of individual from genetic solution and 10% of individual by random generation as the basis of the transformation of pheromone value. The best combination of genetic operators in GA was also discussed. Besides, (Weber, 2006) proposed a distributed algorithm in which ant colonies and genetic algorithms work independently of each other and only communicate when better solutions are discovered. Ant colonies are used to explore the solution space, while genetic algorithms are used to improve the convergence rate of the search.

3. Proposed algorithm

An important research topic in the research area of computational intelligence is swarm intelligence which consists of a population of artificial agents mimicking the animals' behavior in the real world. Each agent follows some simple rules and interacts with the others to share information to lead the behavior to convergence. It should be noted that the proposed algorithm is a method producing a new solution out of components of routes called the bones of previous solutions. This component of used routes is sequences of nodes. In other words, the fact behind the bone route is to extract bone nodes with predefined size and frequency of the adaptive memory (AM). Then the bones will be used to construct new solutions. The size and frequency specified by the algorithm-designer restrict the number of nodes in a bone (bone-size) and the minimum number of stored routes in the AM that must include a bone (bone-freq-min). In our approach, we have made two main modification respects to the Bone Route method:

1- Value of bone-size systematically changes during run time.

2- In order to extract the bones from AM, they must satisfy an additional criterion that specifies the maximum number of stored routes in the AM that must include a bone (bone-freq-max).

The rationale behind changing mentioned parameters is that the value of the bone-freq-max and bone-size express the degree of similarity among the new constructed solution and the previously stored solutions in the AM. Therefore, the new solution is more similar to other solutions in AM, whenever their value is high. In the proposed RBRA, *n* different diversified initial solutions are generated randomly. The goal of building multiple initial solutions is to spread the search in order to explore different regions of the solution space of the problem (diversification strategy). After producing n solutions, the AM is Constructed using improved solutions of previous step in which their routes are sorted by increasing costs of relative solutions. In each iteration, the *s* randomly detects bone-like sequences of a predetermined number of nodes using the nearest neighborhood heuristic. Then bone sequences are used to construct a new solution. It is noted that selected bones must not contain common nodes between them. In order to avoid this, the bones are selected that belong to the high quality solution. Furthermore, if those solutions have the same cost, that one is choosing with the highest frequency. Then, new solutions are generated by combining the extracted bones.

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. So,

three different neighborhood operators are considered in the proposed algorithm including insert, swap and 2-opt algorithms in this step (Figure 2). In insert move a node from its position is changed to another position. In the swap move, two nodes from the route are changed. In 2-opt move, two non-adjacent edges are replaced by two other edges. These algorithms lead to improve the new constructed solution generated in the previous step and set as the best solution if it is better than the previous elite solution. It should be noted that there are several routes for connecting nodes in order to produce the tour again, but a state that satisfies the problem's constraints is acceptable. As a result, the new 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.

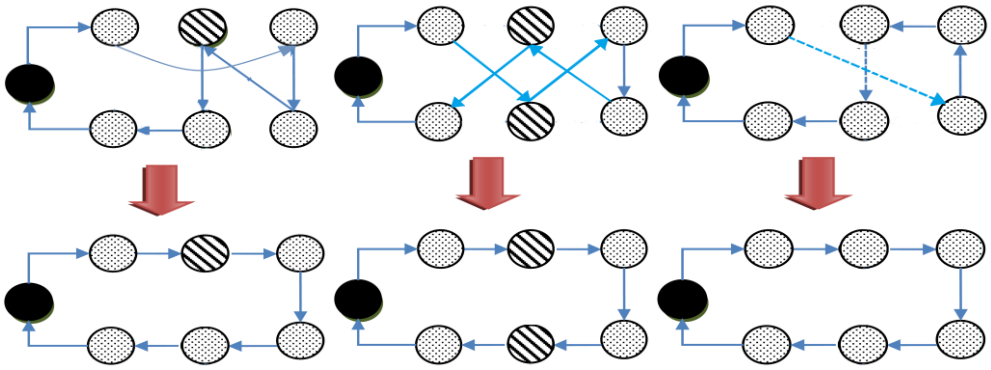


Figure 2. insert (left), swap (middle) and 2-opt exchanges (right)

If AM is not full, the AM is updated and the new improved solution is added, Otherwise AM is updated by inserting the routes of new improved solution and removing routes that belong to the worst solution when the new improved solution is better than the low quality solution in AM. If the best solution of algorithm in current iteration has been improved, the search process is intensified in neighborhood of current solutions by applying these local search algorithms again. Then replace the new solution in this procedure instead of the best solution in AM if it has higher quality. This technique leads to increase the convergence of the algorithm to the best solution.

To effectively implement a metaheuristic, one of the most important elements is how to employ the search memory to get a compromise between the diversification and intensification in the search space of the problems. The intensification forces the search to check the neighborhood of some good solutions. Such procedure is a kind of utilization and learning from the accumulated experience. Nevertheless, the diversification is to explore the unexplored regions in the search space. In this step, If the meta-heuristic has not updated the best solution for a prespecified number of consecutive iterations, we must drive the search towards a part of the solution space that has not explored yet (diversification policy). So, we decrease the similarity among the solutions in AM with decreasing the value of bone-size in a number of consecutive iterations. After the diversification policy, the search process is intensified by increasing the value of bone-size for a number of consecutive iterations. This reactive behavior of the adaptive memory based metaheuristic provides a diversified solution to explore the solution space more precisely. Figure 3 shows the main steps of the proposed algorithm.

1: Generate n initial solutions randomly and determine the best solution.
2: Construct the Adaptive Memory (AM) and sort the routes by increasing the costs of their relative solutions.
Repeat
3: Extract the promising bone sequences of nodes according to some selection criteria.
4: Extract the selected bones from the AM, according to the values of the bone-size and bone-freq-min.
5: Generate n new solutions using the extracted bones.
6: Improve the quality of the new constructed solution generated in Step 6 by local search algorithms and obtain the best current solution.
7: update AM.
8: if the best solution until now is improved, apply local search algorithms on that solution and then update AM (If necessary)
9: update parameter
Until the best solution has not been changed for 20 iterations;

Figure 3: Outline of the RBRA

4. Experiments and computational results

The RBRA was implemented on two sets of symmetric and Euclidean TSP datasets from 24 to 1655 cities from TSPLIB95 (Reinelt, 2012). The word ‘‘symmetric’’ means that the travel cost from city A to city B is the same as the travel cost from city B to city A. The first experiment was concerned with the comparison of the performance of RBRA in small TSP instances with the results obtained from several algorithms involving GR24, Bayg29, GR48, ATT48, Eil51, Berlin52, St70, Eil76, KroA100, KroB100, KroC100, KroD100, KroE100, Eil101, Lin105, KroA150, KroB150, KroA200, and KroB200 in Table 2. The second set of experiments was concerned with testing RBRA’s performance with another recently published study on KroE100, Pr152, Gil262, Rd400, Rat575, Rat783, Pr1002, D1291, Rl1323 and Dl1655. The algorithm was implemented in C and implemented on a Pentium 4, 3 GHZ (2 GB RAM) PC with windows XP. Like every meta-heuristic algorithm, the quality solutions produced by the proposed algorithm have been dependent on the seed used to generate the sequence of pseudo-random numbers and on the different values of the parameters. Therefore, a number of different alternative values were tested and the ones selected are those that gave the best computational results concerning both the quality of the solution. Although the results confirm that our parameters setting worked well, it is also possible that the better solutions may exist. Thus, the selected parameters are given in Table 1. All of the parameter values have been determined on the Eil51 by the numerical experiments. Furthermore, the proposed algorithm stops after no improvements are found for 20 iterations. To reveal the variety of the RBRA's performance from one run to another, 10 runs are carried out for each instance with different random numbers.

Table 1. Parameter values

Parameter	Candidate	Value
AMsize	3,5,7,9	7
Bonefreq_min	1,2,3,4,5	2
Bonefreq_max	3,5,7,9	7
Number of consecutive iterations. Diversification policy is done	2,4,6,8	4
Number of consecutive iterations. Intensification policy is done	2,4,6,8	4
Stop condition	10,15,20,25	20

In the first set of problems shown in table 2, the algorithm was tested on a set of 19 Euclidean sample problems with sizes ranging from 24 to 200 nodes. The sets of data used for the experiment are TSP instances available on the TSPLIB (Gutin and Punnen, 2002). In Table 2, the first column shows the name of the instance and the second column specifies the instance's size referenced. Moreover, the third, fourth, fifth and sixth columns show the four meta-heuristic algorithms including genetic algorithm (GA), ant colony system (ACS), particle swarm optimization (PSO) (Zhong, et al., 2007) and bee colony optimization (BCO) (Wong, et al., 2008). Finally, the seventh column presents the best solution of the proposed algorithm through 10 independent runs and the best solutions published in the literature and also on the web (BKS) are presented in eighth column.

Table 2. Comparison of algorithms for standard problems of the TSP

Instance	N	GA	ACS	PSO	BCO	RBRA	Optimum
GR24	24	1272	1272	-	-	1272	1272
Bayg29	29	1610	1610	-	-	1610	1610
GR48	48	5047	5046	-	-	5046	5046
ATT48	48	10643	10628	-	10661	10628	10628
Eil51	51	429	427	427	428	427	426
Berlin52	52	7548	7542	7542	-	7542	7542
ST70	70	688	679	-	-	675	675
Eil76	76	549	542	540	539	538	538
KroA100	100	21540	21309	21296	21763	21282	21282
KroB100	100	22431	22183	-	22637	22183	22141
KroC100	100	22993	20787	-	20853	20815	20749
KroD100	100	21591	21341	-	21643	21341	21294
KroE100	100	22198	22109	-	22450	22068	22068
Eil101	101	643	636	-	635	631	629
Lin105	105	14703	14534	-	15288	14524	14379
KroA150	150	27054	26749	-	27858	26749	26524
KroB150	150	26659	26431	-	26535	26202	26130
KroA200	200	30276	29762	29563	29961	29368	29368
KroB200	200	31980	29653	-	30350	29586	29437

Besides, Figure 4 shows the gap value of the RBRA, where the gap is defined as the percentage of deviation from the best known solution in the literature. The gap is equal to $100[c(s^{**}) - c(s^{*})]/c(s^{*})$, where s^{**} is the best solution found by the algorithm for a given instance, and s^{*} is the overall best known solution for the same instance on the web. A zero gap indicates that the best known solution is found by the algorithm. As can be seen from Figure 4, the RBRA finds the optimal solution for ten out of nineteen problems that are published in the literature. For instances Lin105, KroA150 and KroB200, the gap is relatively as high as 0.4. However, in other instances, the proposed algorithm finds nearly the best known solution, i.e., the gap is below 0.3, and overall, the average difference is 0.20.

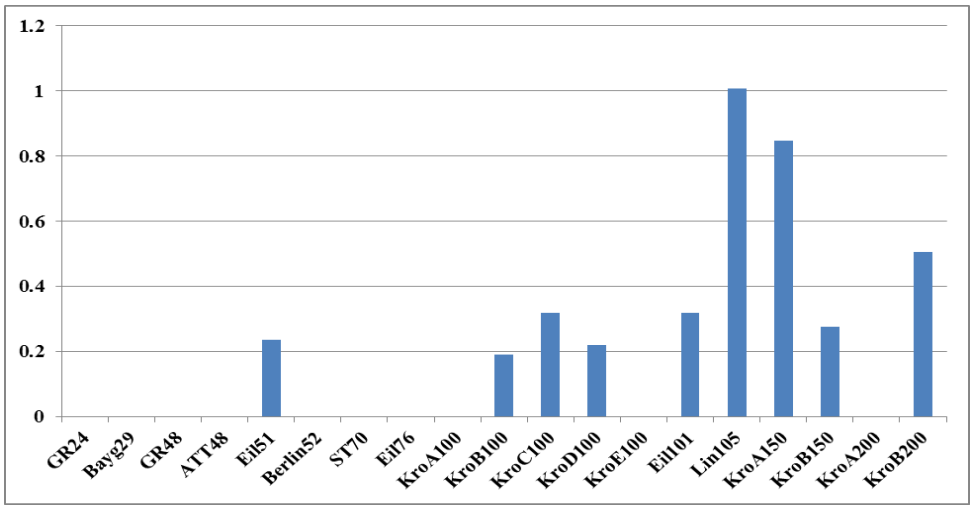
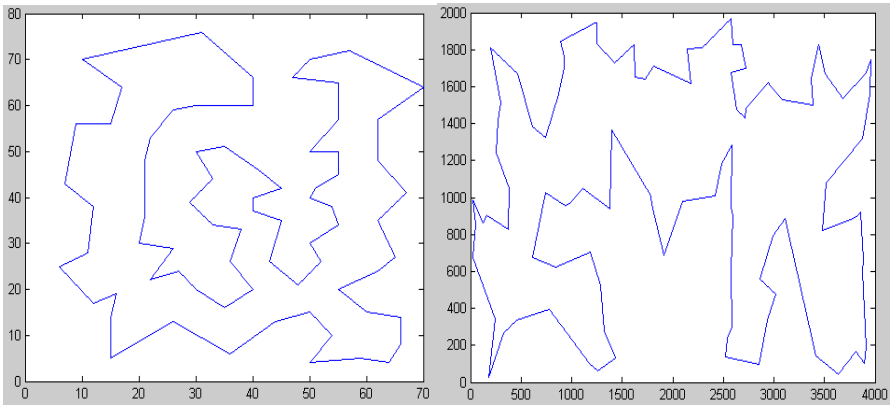


Figure 4. The gap value of the RBRA

The results also show that the RBRA has the ability to escape from local optimum and find the best solutions for most 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 KroB200. Furthermore, the results indicate that although the ACS gives an equal solution to the proposed algorithm for 9 instances, this algorithm cannot gain optimal solutions for others and yields worse solutions than the proposed RBRA algorithm. Besides, 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. The performance Comparison of results shows that the proposed algorithm method clearly yields better solutions than the BCO. Moreover, Computational results of the RBRA and PSO shows that these algorithms have a close competition and the proposed algorithm gives better 4 solutions than PSO. In other words, the performance of the proposed algorithm is better in reaching the sub-optimal solution than the PSO. As a result, the proposed algorithm yields better solutions than the GA, PSO, ACS, and BCO.

In addition, in order to demonstrate the efficiency of the RBRA, two of the solutions found for the examples in table 2 are presented in Figure 5. It should be noted that in two examples presented this figure, the RBRA has been able to find the best solution ever found.



Eil76 KroA100

Figure 5. Some of the Solutions to the TSP Found by RBRA

In table 3, 11 large-scale problems of the TSP are considered in which the number of customers ranges in size from 100 to 1655. Each problem exhibits a geometric symmetry which allows us to visually estimate a solution. In all of these instances, the vertices are taken to be points located in the Euclidean plane and the cost of an edge is then taken to be equal to the Euclidean distance between its end-vertices computed with real numbers. In this table, some of the characteristics of these problems are described. The first column includes the instance name and the second column shows the best know solution for each instance. The objective of the computational experiments is to compare the performance of the proposed algorithm with several famous metaheuristic algorithms. For achieving this goal, seven different metaheuristic approaches given in the literature for the TSP such as ABO (Odili, Mohmad Kahar, 2016), PSO, ACO and HPSACO (Jia, 2015), ANN (Cochrane, Beasley, 2003), SNN (Masutti, Castro, (2009) and GSAP (Chen, Chien, 2011) are considered in columns 3 to 9. Finally, results of the proposed RBRA is shown in the last column. In this table, three sub-columns which include the best gained solution, average solutions and Gap of the best obtained solution compared with BKS are allocated to each algorithm. To measure the efficiency and the quality of an algorithm, a simple criterion is to compute the number of optimal solutions found in specific benchmark instances by algorithm. As it can be seen from this table, the proposed algorithm RBRA finds the optimal solution for 4 out of 11 problem instances and these solutions have been published in the literature. Moreover the PSO, ACO, HPSACO, ANN, SNN, ABO and GSAP have been able to find 0,0,0,0,0,1 and 1 optimal solutions from among these instances. These results indicate that the proposed algorithm is a competitive approach compared to mentioned algorithms and the results are much better than the ones found by these algorithms. As it is shown in (Tarantilis *et al.* 2008), direct comparisons of the required computational times cannot be conducted as they closely depend on various factors such as the processing power of the computers, the programming languages, the coding abilities of the programmers, the compilers and the running processes on the computers.

A simple criterion to measure the efficiency and the quality of an algorithm is to compute the Gap of its solution from the BKS on specific benchmark instances. Figure 4 shows the mean solutions Gap of our algorithm and the ones for seven other metaheuristic algorithms. From this figure, we conclude that the proposed method has the best deviation from the BKS. In more detail, the best algorithm is RBRA which has almost found the best known solutions for all 3 examples including KroE100, Pr152 and Gil2621291 and is competitive with other algorithms. However, in other instances except Rl1323, Fl1400 and D1655, the proposed algorithm finds nearly the BKS, i.e. the gap is about as high as 2. The performance Comparison of results shows that the proposed HACO clearly yields better solutions than the other algorithms.

Table 3. Comparison between results of the RBRA with other algorithms

Instance	Opt	ABO			PSO			ACO			HPSACO			ANN			SNN			GSAP			RBRA		
		Best	Ave	Gap	Best	Ave	Gap	Best	Ave	Gap	Best	Ave	Gap	Best	Ave	Gap	Best	Ave	Gap	Best	Ave	Gap	Best	Ave	Gap
KroE100	22,068	-	-	-	-	-	-	-	-	-	-	-	-	22403.43	22632.94	1.52	22394.61	22714.59	1.48	22068	22182.75	0.00	22068	22097	0
Pr152	73682	73730	73990	0.07	75361	75405	2.28	74689	74936	1.37	74165	74654	0.66	-	-	-	-	-	-	-	-	-	73682	73713	0
Gil262	2378	2378	2386	0	2513	2486	5.68	2463	2495	3.57	2413	2468	1.47	-	-	-	-	-	-	-	-	-	2378	2378	0
Rd400	15281	15301	15304	0.14	16964	17024	11.01	16581	16834	8.51	16067	16513	5.14	-	-	-	-	-	-	-	-	-	15291	15392	0.07
Rat575	6773	-	-	-	-	-	-	-	-	-	-	-	-	7104.2	7192.926	4.89	7047.307	7115.714	4.05	6890.85	6934.197	1.74	6811	6899	0.56
Rat783	8806	-	-	-	-	-	-	-	-	-	-	-	-	9304.42	9384.554	5.66	9246.3	9344.047	5.00	8988.284	9078.986	2.07	8894	8987	1.00
Pr1002	259045	259132	261608	0.03	278923	279755	7.67	269758	271043	4.14	267998	269789	3.46	-	-	-	-	-	-	-	-	-	259167	261231	0.05
D1291	50801	50839	50839	0.07	53912	54104	6.12	52942	53249	4.21	52868	52951	4.07	-	-	-	-	-	-	-	-	-	50801	51064	0
Rl1323	270	-	-	-	-	-	-	-	-	-	-	-	-	300.483	301.995	11.29	300.537	305.1	11.31	277.425	279.963	2.75	275	278	1.85
Fl1400	20,127	-	-	-	-	-	-	-	-	-	-	-	-	20553.69	20984.41	2.12	20851.57	21109.2	3.60	20593.95	21348.71	2.32	20452	21152	1.61
D1655	62,128	-	-	-	-	-	-	-	-	-	-	-	-	67781.65	68396.72	9.10	70919.11	72111.97	14.15	64153.37	65619.59	3.26	63654	65812	2.46

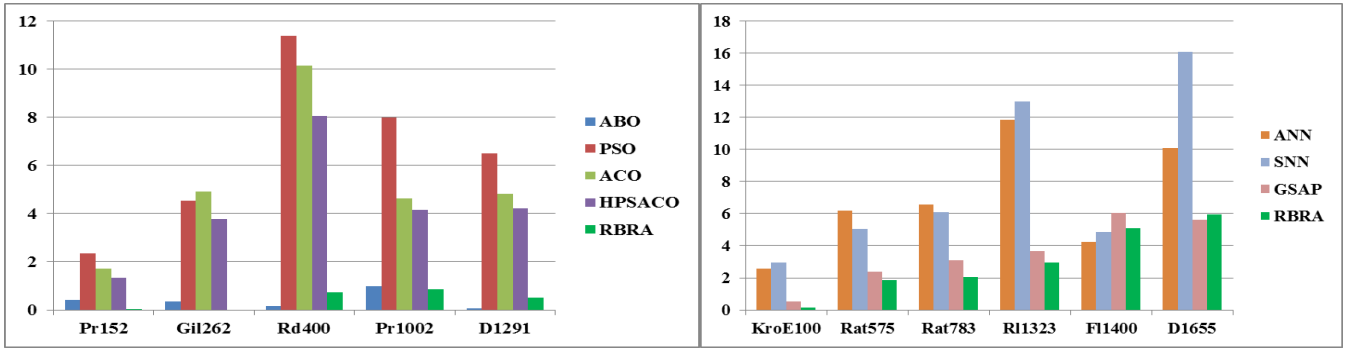


Figure 6. Comparison between mean solutions gap of the algorithms

5. Acknowledgment

The authors would like to acknowledge the Young Researchers And Elite club, Robatkarim Branch, Islamic Azad University, Robatkarim, Iran for the financial support of this work.

6. Conclusions

In this paper, a reactive bone route algorithm which uses several local search algorithms for improvement procedure was proposed for solving the TSP as an example application problem. Experiments are implemented to evaluate the algorithm's performance on some test instances of TSPLIB. Computational results demonstrate that our algorithm is effective for solving TSP. As shown, in most cases of instances, the best results can be obtained by our algorithm and in cases, the Gap between the BKS reported and the solution found by the proposed algorithm for most of the instances is very satisfied. Using this proposed algorithm for other versions of the TSP and 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.

References

Ahmadvand, M., YousefiKhoshbakht, M. and Mahmoodi Darani, 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. and Munlin, M. 2015, "Using K-means Radius Particle Swarm Optimization for the Travelling Salesman Problem", *IETE Technical Review*, 1-9.

Bachir Menai, M. E. and Batouche M. 2005, "Solving the maximum satisfiability problem using an evolutionary local search", *The International Arab Journal of Information Technology*, 2(2), 154-161.

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

Barketau, M. and Pesch, E. 2015, "An approximation algorithm for a special case of the asymmetric travelling salesman problem", *International Journal of Production Research*, DOI: 10.1080/00207543.2015.1113327.

Bianchi L., Knowles, J. and 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.

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

Chen, S. M. and Chien, C. Y. 2011, "Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques", *Expert Systems with Applications*, 38(12), 14439-14450.

Chen, X., Tan, Z., Yang, G. and Cheng, W. 2012, "A hybrid algorithm to solve traveling salesman problem, In Advances in Electronic Engineering", *Communication and Management*, 1, 99-105. Springer Berlin Heidelberg.

Clarke, G. and Wright, J.W., 1964, 'Scheduling of vehicles from a central depot to a number of delivery points', *Operations Research*, 12, 568-581.

Cochrane, E. M. and Beasley, J. E., 2003, "The co-adaptive neural network approach to the Euclidean traveling salesman problem", *Neural Networks*, 16(10), 1499–1525.

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

Créput, J. C, and Koukam, A., 2009, "A memetic neural network for the Euclidean traveling salesman problem", *Neurocomputing*, 72(4-6), 1250-1264.

Dorigo, M., Maniezzo, V. and 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.

Garey, M.R. and Johnson, D.S., 1979, "Computers and intractability: A guide to the theory of NP-completeness", *San Francisco: W.H. Freeman.*
Gillett, B.E. and Miller, L.R., 1974, "A heuristic algorithm for the vehicle dispatch problem", *Operations Research*, 22, 340-349.
Gutin, G. and Punnen, A., 2002, "The Traveling Salesman Problem and its Variations", *Kluwer Academic Publishers*, Dordrecht.
Hernandez-Perez, H., Rodríguez-Martín, I. and Salazar-Gonzalez, J. J., 2009, "A hybrid GRASP/VND heuristic for the one-commodity pickup-and-delivery traveling salesman problem", *Computers & Operations Research*, 36(5), 1639-1645.
Jaafar, A. and Samsudin, A., 2013, "An Improved Version of the Visual Digital Signature Scheme", *The International Arab Journal of Information Technology*, 10(6), 20-32.
Jia, H., 2015, "A novel hybrid optimization algorithm and its application in solving complex problem", *International Journal of Hybrid Information Technology*, 8(1), 1–10.
Johnson D.S. and McGeoch L.A., 2002, "Experimental analysis of heuristics for the STSP, in G. Gutin and A.P. Punnen (eds.)", *The Traveling Salesman Problem and Its Variations*, Kluwer Academic Publishers, The Netherlands, 369–443.
Karapetyan D. and Gutin, G., 2011, "Lin-Kernighan Heuristic Adaptations for the Generalized Traveling Salesman Problem", *European Journal of Operational Research*, 208 (3), 221–232.
Karp, R. M., 1997, "Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane", *Mathematics of Operations Research*, 2, 209-224.
Lin S., 1965, "Computer solutions of the traveling salesman problem", *Bell System Technical Journal*, 44, 2245-2269.
Liu Y. H., 2007, "A hybrid scatter search for the probabilistic traveling salesman problem", *Computers & Operations Research*, 34(10), 2949-2963.
López-Ibáñez M. and Blum C., 2010, "Beam-ACO for the traveling salesman problem with time windows", *Computers & Operations Research*, 37(9), 1570-1583.
Masehian, E., Akbaripour H. and Mohabbati-Kalejahi N. 2014, "Solving the n-Queens Problem using a Tuned Hybrid Imperialist Competitive Algorithm", *The International Arab Journal of Information Technology*, 11(6).
Masutti, T. A. S. and 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.
Masutti, T. A. S. and Castro, L. N., 2009, "Neuroimmune approach to solve routing problems", *Neurocomputing*, 72, 2189–2197.
Michalewicz, Z., 1994, "Genetic Algorithms + Data Structures = Evolution Programs". *Springer-Verlag*, 2nd edition.
Odili, J. B. and Mohmad Kahar, M. N., 2016, "Solving the Traveling Salesman's Problem Using the African Buffalo Optimization". *Computational Intelligence and Neuroscience*, 2016
Reeves C. R., 1993, editor. "Modern Heuristic Techniques for Combinatorial Problems", *Blackwell Scientific Publishing*, Oxford, England.
Reinelt, G., 2012, "Tsplib95", 1995, *http://comopt.ifi.uni-heidelberg. de/software/TSPLIB95*.
Shi X. H., Liang Y. C., Lee H. P., Lu C. and Wang, Q.X., 2007, "Particle swarm optimization-based algorithms for TSP and generalized TSP", *Information Processing Letters*, 103(5), 169-176.
Tarantilis, C.D., Zachariadis, E. and Kiranoudis, C., 2008, "A guided Tabu search for the heterogeneous vehicle routing problem", *Journal of the Operational Research Society*, 59, 1659-1673.
Tarkov, M. S., 2015, "Solving the traveling salesman problem using a recurrent neural network", *Numerical Analysis and Applications*, 8(3), 275-283.
Thiago A.S., Masutti L. and Castro N., 2009, "A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem", *Information Sciences*, 179(10), 1454-1468.
Xiao, M., and Nagamochi, H., 2016, "An Improved Exact Algorithm for TSP in Graphs of Maximum Degree 4", *Theory of Computing Systems*, 58(2), 241-272.

Yadlapalli S., alik W.A., Darbha S. and Pachter M., 2009, "A Lagrangian-based algorithm for a Multiple Depot", Multiple Traveling Salesmen Problem, *Nonlinear Analysis: Real World Applications*, 10(4), 1990-1999.

Yang, J., Ding, R., Zhang, Y., Cong, M., Wang, F. and Tang, G., 2015, "An improved ant colony optimization (I-ACO) method for the quasi travelling salesman problem (Quasi-TSP)", *International Journal of Geographical Information Science*, 29(9), 1534-1551.

Yousefikhoshbakht M., Didehvar F. and 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. and Sedighpour M., 2013, "New Imperialist Competitive Algorithm to solve the travelling salesman problem", *International Journal of Computer Mathematics*, 90 (7), 1495-1505.

Yousefikhoshbakht, M. and Sedighpour, M., 2012, "A Combination of Sweep Algorithm and Elite Ant Colony Optimization for Solving the Multiple Traveling Salesman Problem", *Proceedings of the Romanian academy A*, 13(4), 295-302.

Wang C., 2010, "A hybrid algorithm based on genetic algorithm and ant colony optimization for Traveling Salesman Problems", *Information Science and Engineering 2nd International Conference on Date of Conference*, 4257 – 4260.

Weber B., 2006, "Distributed Hybrid Metaheuristics for Optimization", *Work paper*, 1-12.

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

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