Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2018) 26: 605 – 617

doi:10.3906/elk-1705-61

An elitist approach for solving the traveling salesman problem using an animal migration optimization algorithm

Ezgi ÜLKER*
Department of Computer Engineering, Faculty of Engineering, European University of Lefke, Lefke, Northern Cyprus

Received: 05.05.2017 • **Accepted/Published Online:** 14.11.2017 • **Final Version:** 26.01.2018

Abstract: This paper presents an improved version of the animal migration optimization (AMO) algorithm for solving the traveling salesman problem (TSP), which is classified as a combinatorial NP-hard problem. AMO is one of the recent metaheuristic algorithms inspired by the migration behavior of animals and has been efficiently applied to a variety of optimization problems. The algorithm is improved by reconstructing the neighborhood topology of each animal during the migration. This modified algorithm is called the elitist animal migration optimization (ELAMO) algorithm, since elitism is introduced as a way in which the positions of the leaders are considered for the neighborhood scheme. To observe the performance of ELAMO, it is compared with AMO and some efficient algorithms. The experimental results showed that the ELAMO algorithm has improved the solution quality of AMO and has produced effective or even competitive values for the selected TSP data sets.

Key words: Animal migration, nature-inspired metaheuristics, optimization, traveling salesman problem

1. Introduction

Solving complex optimization problems is a challenging task for many researchers. The various complex optimization problems are classified as multimodal and multiobjective, and finding their optimal solutions requires a considerable amount of time. The traveling salesman problem (TSP) is categorized as a complex NP-hard problem due to the increasing complexity of the number of cities visited in the field of optimization [1]. The problem consists of a number of cities and a salesman who wishes to visit these cities only once and at the end of the journey should return to the city from which he began. In addition, the problem aims to minimize the total cost of this travel. Although this problem is regarded as quite complex, it has been an intriguing area of study for researchers. Since the problem's first introduction in 1949 by Robinson [1], new techniques have been proposed to tackle the TSP.

The TSP has been handled by various metaheuristic algorithms because of their efficiency and easy adaptation to such a complex problem. These algorithms include simulated annealing, greedy search and genetic algorithms [2–4], ant colony optimization and swarm intelligence [5–7], discrete cuckoo search [8], and artificial bee colony optimization [9]. However, as accepted in the "no free lunch theorem", there is no metaheuristic algorithm that gives the best performance for all kinds of optimization problems [10]. Therefore, improvements on metaheuristics continue to have great importance and researchers work especially to solve complex problems. This paper concentrates on a recently developed metaheuristic, called the animal migration optimization (AMO)

*Correspondence: eulker@eul.edu.tr

algorithm, by Li et al. in 2014 [11]. Since the AMO algorithm is efficient and one of the most recently introduced techniques to solve optimization problems, researchers have presented improved versions to achieve better solution quality. One of these techniques is known as the improved AMO (IAMO) algorithm, which is specifically proposed for clustering analysis [12]. It has been constructed by restricting the boundary of the living area of animals and has been compared with several well-known algorithms including PSO, CPSO, ABC, CABC, and the original algorithm, AMO. It is observed that IAMO outperforms all the analyzed algorithms for certain clustering problems. Another study researched constrained engineering optimization problems [13]. This improved algorithm is called the fast convergence AMO algorithm and it is based on reducing the search space of animals dynamically. It has been verified in solving complex constrained engineering problems and it is seen that it has better performance than the algorithms ABC, CS, and BA, and the original algorithm, AMO.

Due to the promising solution quality of AMO-based algorithms, the introduction of an improved version of the AMO algorithm is the main aim of this study. The improved algorithm is called elitist animal migration optimization (ELAMO), taking its name from the tendency of animals to follow their leaders in a herd. Although ELAMO can be applicable to all optimization problems, in this paper it is specifically adapted to the TSP. The standard version of AMO is based on the migration behavior of an animal in the herd and it is assumed that each animal follows its neighbors during migration. On the other hand, in ELAMO, each animal only follows those leaders that have the best positions during migration. This approach results in high solution quality with respect to the performance of AMO. Furthermore, ELAMO has the advantage of elitism, according to the comparison results for certain other metaheuristics that were especially adapted to solve the TSP.

The rest of the paper is organized as follows. Section 2 briefly describes standard AMO and the main improvements on animals during migration in biological and computational aspects. Section 3 gives information about the symmetric TSP and Section 4 introduces the adaptation of ELAMO for solving the TSP. Section 5 presents the comparative experimental results of ELAMO and other algorithms for various TSP data sets and discusses the results. Finally, Section 6 gives the concluding remarks for the paper.

2. Animal migration optimization algorithm

2.1. Standard AMO algorithm

The AMO algorithm was introduced by Li et al. [11] and is derived from the migration behavior of animals to discover better life areas. Animals that belong to a herd can migrate long distances due to climate changes or lack of food in their current habitat. All the animals, including the leader of the herd, should follow three generalized rules during the process of migration: i) an animal should move according to its neighbors' positions; ii) an animal's position should be close to its neighbors' positions; and iii) an animal should retain a distance from its neighbors to avoid collisions.

The AMO algorithm is implemented in two fundamental steps:

- Animal migration step: the animals change their directions according to the positions of their close neighbors, using the following formula:

$$X_{i,G+1} = X_{i,G} + \delta(X_{neighbor,G} - X_{i,G}), \tag{1}$$

where δ is a random number produced by Gaussian distribution, G is the generation counter, and $X_{neighbor,G}$ is the current neighbor of the animal $X_{i,G}$.

A neighborhood structure is necessary for each animal in a herd. The ring topology is used to construct the possible neighbors of an animal i. The number of possible neighbors is fixed to 5, as recommended in the studies of AMO [11,14]. Figure 1 illustrates the neighborhood scheme of an animal i.

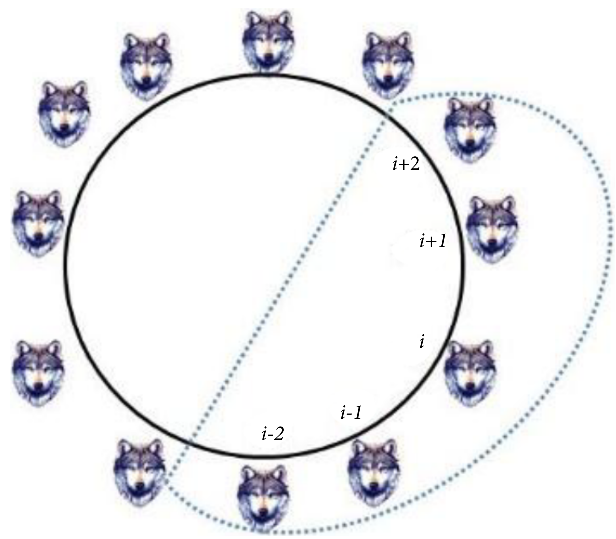


Figure 1. Neighborhood topology of AMO.

- Population updating step: animals come together by joining herds or they naturally join them at birth. The same number of animals join the herd as the number of animals that leave, according to probability Pa, using the following formula:

$$X_{i,G+1} = X_{r1,G} + rand\left(X_{best,G} - X_{i,G}\right) + rand\left(X_{r2,G} - X_{i,G}\right) \tag{2}$$

where $r_1,\ r_2, i \in [\ 1\ ...\ NP]$, $r_1 \neq r_2 \neq i$, $rand$ is a random number between 0 and 1, and X_{best} is the best position of an animal obtained so far.

In the AMO algorithm, each animal can trigger the migration process by constructing its neighborhood scheme and can update its position by considering the positions of its nearby neighbors. The main steps of AMO are given in the form of a flowchart in Figure 2.

2.2. Elitist AMO algorithm

In the animal migration algorithm, animals take their close neighbors' positions as a reference to explore better life conditions. A significant improvement in AMO is achieved by changing the existing neighborhood topology. In ELAMO, animals follow their leaders instead of their close neighbors. Each animal constructs its neighborhood scheme by considering the positions of alpha and beta animals in the herd. Three types of casts appear in the hierarchical structure of a herd. The first is the leader of the herd, known as alpha, and it is responsible for the existence of the current population. The alpha animal decides the migration process for finding a new life area or protecting the herd from other threats. The second type of animal is called beta, and it is the second animal in charge after the alpha animal. When the alpha leaves the herd for hunting, beta animals are responsible for the others. There can be more than one beta animal, and when the alpha dies or becomes old, beta animals compete against each other to become the new alpha of the herd. The remaining animals that obey the rules determined by the alpha and beta animals are called omega. The ELAMO algorithm mainly

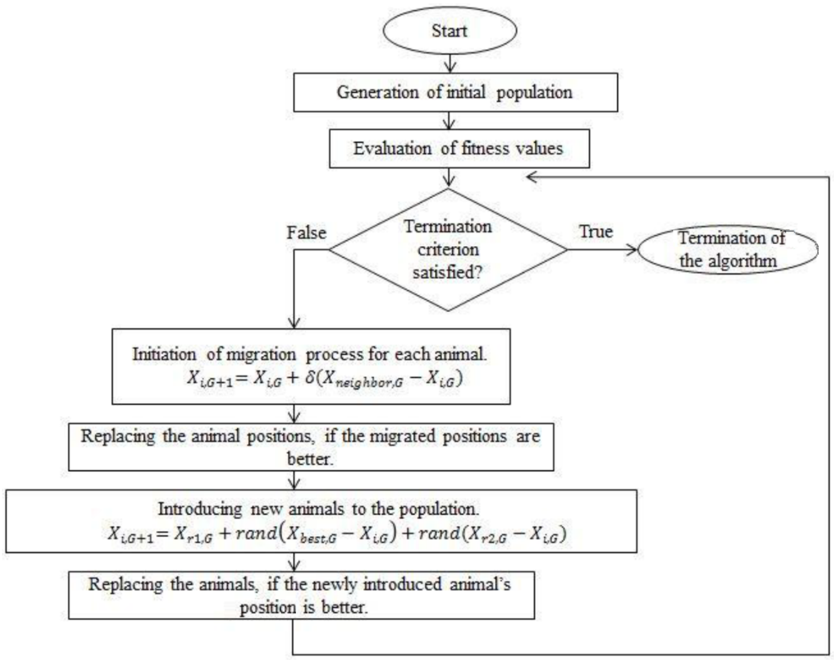


Figure 2. Main steps of standard AMO in form of a flowchart.

relies on this hierarchical structure of animals and their basic instincts for migration. Considering the three main rules proposed by AMO, the ELAMO implements the following steps: i) an animal should move according to the positions of the alpha and beta animals; ii) an animal's position should be close to the alpha and beta positions; and iii) an animal should retain a distance between its alpha and betas to avoid collisions.

The neighborhood topology of ELAMO is constructed only among the alpha and beta animals. These animals have better positions than the other animals in the population. It is assumed that the alpha has the best position and the number of alphas is fixed to 1, whereas the positions of beta animals are ranked after the alpha according to their fitness values and their number is set to 5. The neighborhood selection is illustrated in Figure 3 for an animal i. The processes of migration and population update are adopted from AMO and implemented in ELAMO after significant improvements.

- Animal migration process: animals move from one region to another by following their leaders. The position of an animal i is updated by the positions of the alpha and betas using the following formula:

$$X_{i,G+1} = X_{i,G} + \delta(X_{neighborLeader,G} - X_{i,G}), \tag{3}$$

  where δ is a random number produced by Gaussian distribution, G is the generation counter, and $X_{neighborLeader,G}$ is the leader's position at the current generation, which is randomly selected from the neighborhood structure of animal X_i.

- Population updating process: animals can be expelled as a result of going against the rules of the herd or due to death. A new alpha is selected among the beta animals when the alpha of the population dies or becomes old. The beta animals compete against each other with respect to their positions. The winner is accepted as the new alpha and the loser or losers may be expelled. When a beta animal is selected as the new alpha, the previous position of the beta is occupied by an omega animal according to its position.

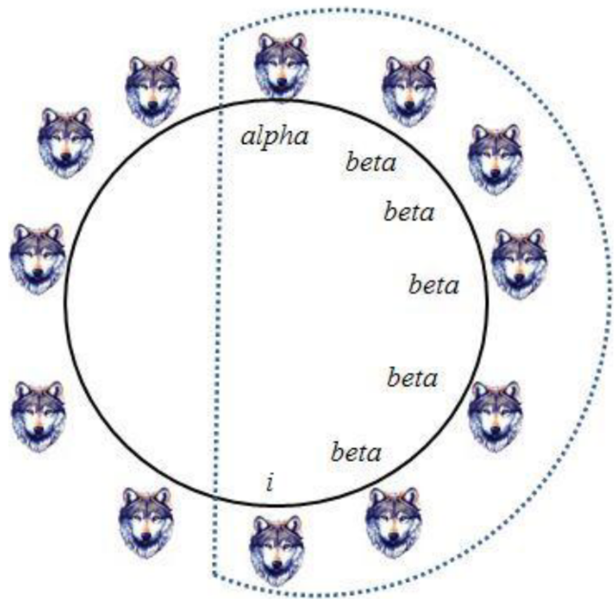


Figure 3. Neighborhood topology of ELAMO.

The positions of the alpha and beta animals are updated in each generation according to their fitness ranking. These updated positions are used for the other animals to update their positions, as shown in the following equation, by using probability Pa:

$$X_{i,G+1} = X_{betaRand_a,G} + rand\,(X_{alpha,G} - X_{i,G}) + rand(X_{betaRand_b,G} - X_{i,G}), \tag{4}$$

where $X_{betaRand}$ is an animal selected randomly among beta animals, X_{alpha} is the position of the alpha, $rand$ is a random number in between 0 and 1, and $a \neq b$.

Detailed steps of ELAMO are given in Figure 4 in the form of a flowchart, emphasizing the improvements in bold. One of the main benefits of using the ELAMO algorithm is its high performance in tracing the positions of group leaders rather than following close neighbors. However, it is worth mentioning that, in ELAMO, the desired characteristics of an individual may be lost by eliminating it in the early stages of optimization and this may affect the diversification and intensification balance of the algorithm.

3. Traveling salesman problem

The TSP consists of N number of cities, $C = c_1, c_2, c_3 \ldots c_N$, and a distance matrix that defines the distances between each pair of cities, $D = d(c_i, c_j)_{N\times N}$. The aim is to find the minimum tour length under the condition that a salesman visits each city exactly once and returns to the city from where he started to travel [15]. The tour length is defined as a cyclic permutation (π) of the cities visited:

$$f(\pi, C) = \sum_{i=1}^{N-1} d(c\pi_{(i)}, c\pi_{(i+1)}) + d(c\pi_{(N)}, c\pi_{(1)}), \tag{5}$$

where $i = 1 \ldots N$ and $c\pi_{(i)}$ is the city visited in step i.

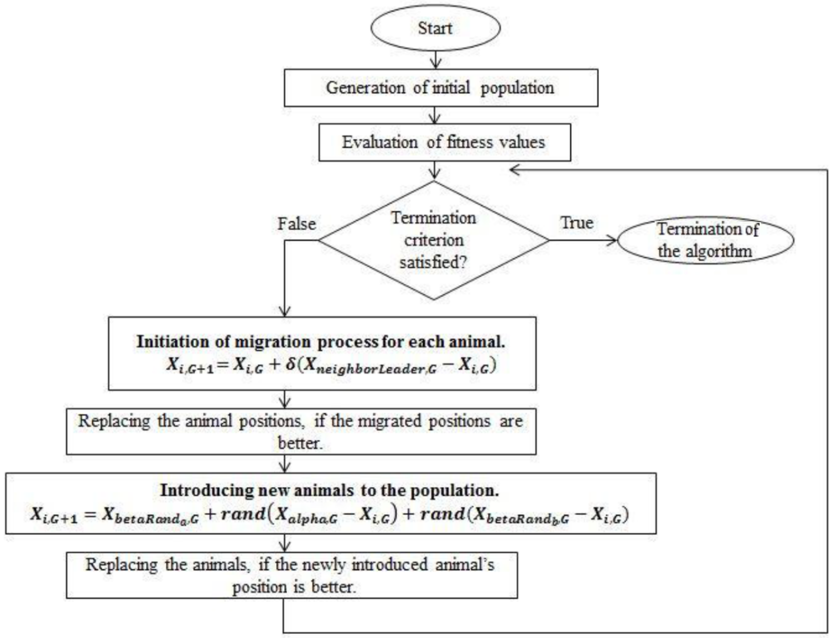


Figure 4. Main steps of ELAMO in form of a flowchart.

This paper considers the symmetric TSP, which satisfies the rule of distance $d(c_i, c_j) = d(c_j, c_i)$, where $1 \leq i, j \leq N$. A weighted graph can be a representation of a TSP with vertices for the cities, edges for the connections between the cities, and the weight of an edge for the connections distance. A TSP tour can be expressed as a Hamiltonian cycle and it is aimed to find the shortest length of the cycle.

4. ELAMO for solving the TSP

It is known that the TSP is a combinatorial NP-hard problem and ELAMO is developed to solve continuous problems. The adaptation of the main parameters used in ELAMO for the TSP is necessary to transform from a continuous space to a combinatorial space. Therefore, adaptations of animals, herd, objective function, and migration process to positions, displacements, and distances are given as follows.

4.1. Animals

In ELAMO, three types of animals are classified in a hierarchical structure. Alpha: there is only one and it is assumed that it has the best position so far. Beta: there are a few, their number is set to 5, and it is assumed that they have the second best position after the alpha. Omega: there are many and it is assumed that all the animals follow the rules of alpha and beta animals. It can be said that all the animals are potential solutions to solving the TSP by representing the coordinate values of the cities as weighted graphs.

4.2. Herd

In ELAMO, the following assumptions are practiced: i) the number of animals is fixed; ii) when an alpha's solution quality is not good enough to solve a problem, a new alpha is selected among beta animals. The position of the selected beta is filled by an omega animal whose fitness value is relatively better than the other omega animals. In adaptation to the TSP, a herd with animals represents a TSP tour with a weighted graph of Hamiltonian cycle.

4.3. Objective function

An alpha leads to better solutions, and it is supposed that the other animals following the alpha is directly related to the solution quality. In the TSP, the objective function is the sum of the Hamiltonian cycle lengths, and an optimal solution refers to the shortest Hamiltonian cycle.

4.4. Migration process

During the migration process, which is triggered by the alpha, the animals update their positions by changing their coordinates. Since the coordinate values of animals are representations of the coordinate values of cities in the TSP, the solution strategy can be figured out by moving animals from one location to another. In the adaptation of the TSP, the migration process of ELAMO is managed by changing the visiting order of cities. It is worth noting that the coordinates of the cities are fixed, whereas the visiting order of the cities is not.

4.5. Experimental results

In order to verify the performance of elitist AMO and show its efficiency over standard AMO, 15 TSP benchmark sets are chosen from the TSPLIB Library according to their complexity [16]. First a detailed comparison between standard and elitist AMO is performed in Table 1, and then ELAMO is compared with PSO and the GA for large-scale and synthetic TSP data sets [15,17] in Table 2. Additionally, the performance of ELAMO is analyzed with methods that are specifically designed to solve the TSP, namely genetic simulated annealing ant colony system with particle swarm optimization techniques (GSA-ACS-PSOT) [3] and a self-organizing neural network (SONN) [18], presented in Table 3. In all tables, the better results are indicated in bold. In Figure 5, the solution quality of ELAMO is illustrated against the solution quality of ant colony optimization (ACO) [6]. The results achieved by the algorithms used in the comparison are achieved as they were in the original works [3,6,15,17,18]. It is necessary to specify that in similar studies [3,18], the authors accessed the results of the previously proposed algorithms for solving TSP data sets and compared them with their own algorithms. For the compared algorithms, GSA-ACS-PSOT and SONN, the maximum number of cycles is set to 1000 and the obtained results involve 30 runs. In the comparison of ELAMO with PSO and the GA, the maximum number of iterations is set to 10,000 and the number of populations is selected as 150. The parameter settings used in all experimental studies for AMO and ELAMO are given in Table 4. The standard and elitist versions of AMO algorithms are simulated with 4 GB RAM on an Intel Core-i5 processor using C++ language. In order to carry out a relevant performance analysis among the algorithms, the known optimum (length value for each instance, which is taken from TSPLIB), the best value (shortest length obtained in 30 independent runs), the worst value (longest length obtained in 30 independent runs), time (s) (averaged computational time to obtain the length value for each instance), PDav (%) (percentage deviation of average solution from the best known solution length of 30 trials), and PDbest (%) (percentage deviation of the best found solution from the best known solution length of 30 trials) are provided. The formulations of PDav (%) and PDbest (%) are given as follows:

$$\mathrm{PDav}\,(\%) = \frac{\text{average solution-best known solution}}{\text{best known solution}} \times 100 \tag{6}$$

$$\mathrm{PDbest}\,(\%) = \frac{\text{best found solution-best known solution}}{\text{best known solution}} \times 100 \tag{7}$$

Table 1. A comparative analysis of AMO and ELAMO for TSP data sets.

TSP instances		AMO						ELAMO					
Data set [16]	opt	best	avg	worst	PDav (%)	PDbest (%)	Time (s)	best	avg	worst	PDav (%)	PDbest (%)	Time (s)
eil51	426	426	439.60	460	3.19	0	12.49	426	**426**	426	**0**	0	9.81
berlin52	7542	7542	7717.40	8096	2.32	0	11.64	7542	**7542.20**	7545	**0.002**	0	8.76
eil76	538	552	568.13	612	5.6	2.60	19.52	**538**	**538**	538	**0**	**0**	14.61
kroA100	21282	21598	22615.07	24432	6.26	1.48	64.88	**21282**	**21411.87**	21612	**0.61**	**0**	58.41
kroB100	22141	22936	24184.33	26104	9.22	3.59	67.43	**22141**	**22143.80**	22162	**0.01**	**0**	59.92
bier127	118282	122767	134757.40	161437	13.92	3.79	72.11	**118282**	**118336.70**	119101	**0.04**	**0**	60.74
ch130	6110	6297	6797.73	7612	11.25	3.06	72.49	**6110**	**6115.46**	6187	**0.08**	**0**	62.31
ch150	6528	7012	7576.73	8143	16.06	7.41	81.12	**6528**	**6532.40**	6594	**0.06**	**0**	67.90
kroA200	29368	31543	33805.13	35982	15.10	7.40	85.32	**29379**	**29445.87**	29881	**0.26**	**0.03**	71.88
lin318	42029	45674	54545.67	71832	29.78	8.67	112.61	**42110**	**42218.07**	42457	**0.44**	**0.19**	109.14
rat575	6773	7012	7316.73	7854	8.02	3.52	349.06	**6898**	**7058.53**	7543	**4.21**	**1.84**	285.29
rat783	8806	9159	9299.26	9643	5.60	4.00	397.19	**8995**	**9128.93**	9412	**3.66**	**2.14**	346.12
rl1323	270199	281754	294042.90	321758	8.82	4.27	1286.74	**279841**	**293151.90**	317958	**8.49**	**3.56**	1128.87
fl1400	20127	21897	22718.93	24952	12.87	8.79	1346.63	**20976**	**22421.93**	23653	**11.40**	**4.21**	1201.03
d1655	62128	67841	71475.67	79723	15.04	9.19	2376.48	**65153**	**70385.33**	74120	**13.29**	**4.86**	1549.78

Table 2. Comparison of the experimental results of ELAMO with PSO and GA for large-scale problems and synthetic data.

TSP instances		PSO [15]		GA [15]		ELAMO	
Data set [15,17]	opt	avg	PDav (%)	avg	PDav (%)	avg	PDav (%)
XQF131	564	584	3.59	576	2.13	**573.60**	**1.70**
XQG237	1019	1070	5.19	1068	4.84	**1057.60**	**3.78**
BCL380	1621	1774	9.44	**1748**	**7.89**	1768.70	9.11
PBM436	1443	1634	10.33	1574	9.10	**1518.60**	**5.23**
Att532	27686	30363	9.67	29718	7.34	**29296.10**	**5.81**
C20	62575	63276	1.12	63188	0.98	**62820**	**0.39**
C30	62716	63625	1.45	63356	1.02	**63294.10**	**0.92**
C40	62768	64212	2.30	**63753**	**1.57**	64348.40	2.51
F32	84180	85535	1.61	85392	1.44	**85046.40**	**1.02**
F41	68168	69995	2.68	**69702**	**2.25**	69733.50	2.29
S21	60000	60786	1.31	60648	1.08	**60386.60**	**0.64**

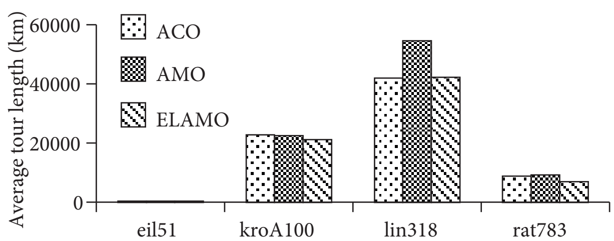


Figure 5. Average tour length comparison of ACO, AMO, and ELAMO.

As seen in Table 1, ELAMO performs better than standard AMO for the 15 analyzed test functions. A total of 53.3% of the values of PDbest (%) are 0.0, which shows that the best solution obtained for these data sets as well as 60% of the values of PDav (%) are less than 0.5%, which implies that almost all values of 30 runs are the same as the known optimum solution. Regarding the average length and computational time, it can be expressed that ELAMO gives better results than AMO. However, it should be noted that when larger TSP instances are analyzed, both algorithms produce competitive results even if they do not obtain the best routes for them. Additionally, a statistical analysis is performed for AMO and ELAMO using the Wilcoxon ranked sign test. The z value turned out to be –3.4077, which indicated that AMO and ELAMO caused significantly different results. When their average time (s) values are compared, it is found that ELAMO has performed approximately 5%–20% faster than AMO. Furthermore, ELAMO has performed quite faster than AMO (approximately 13 min) for the largest data set, d1651. The main improvement is achieved in considering only alpha and beta animals. It is thought that this may affect the speed of optimization when such a large problem is considered. When the AMO and ELAMO algorithms are considered according to their algorithmic complexity, it is seen that both have the same complexity of O (n^2), because the number of nested loops appearing in the algorithms is the same.

According to the results obtained from Table 2, the GA performs better than PSO for all the analyzed TSP data sets. In general, it can be said that ELAMO produces more successful results than the GA. However, the performance of the GA is slightly better than that of ELAMO for only three functions. A similar observation

Table 3. Comparison of the experimental results of ELAMO with GSA-ACS-PSOT and SONN for TSP data sets.

TSP instances		GSA-ACS-PSOT [3]			SONN [18]			ELAMO		
data set [16]	opt	best	avg	PDav (%)	best	avg	PDav (%)	best	avg	PDav (%)
eil51	426	427	427.27	0.3	427	437.47	2.69	**426**	**426**	**0**
berlin52	7542	**7542**	**7542.00**	**0**	7542	7932.50	5.18	7542	7542.20	0.002
eil76	538	538	540.20	0.41	541	556.33	3.41	**538**	**538**	**0**
kroA100	21282	**21282**	**21370.47**	**0.42**	21333	21522.73	1.13	21282	21411.87	0.61
kroB100	22141	22141	22282.87	0.64	22343	22661.47	2.35	**22141**	**22143.80**	**0.01**
bier127	118282	118282	119421.83	0.96	118970	120886.33	2.20	**118282**	**118336.7**	**0.04**
ch130	6110	6141	6205.63	1.57	6145	6282.40	2.82	**6110**	**6115.467**	**0.08**
ch150	6528	6528	6563.70	0.55	6602	6738.37	3.22	**6528**	**6532.40**	**0.06**
kroA200	29368	29383	29738.73	1.26	29600	30190.27	2.80	**29379**	**29445.87**	**0.26**
lin318	42029	42487	43002.09	2.32	42834	43696.87	3.97	**42110**	**42218.07**	**0.44**
rat575	6773	**6891**	**6933.87**	**2.38**	7047	7115.67	5.06	6898	7058.533	4.21
rat783	8806	**8988**	**9079.23**	**3.10**	9246	9343.77	6.11	8995	9128.933	3.66
rl1323	270199	**277642**	**280181.47**	**3.69**	300770	305314.33	13.00	279841	293151.9	8.49
fl1400	20127	**20593**	21349.63	6.07	20851	**21110.00**	**4.88**	20976	22421.93	11.40
d1655	62128	**64151**	**65621.13**	**5.62**	70918	72113.17	16.07	65153	70385.33	13.29

Table 4. Parameter settings used in the experiments.

Parameters	AMO	ELAMO
Population size	30	30
Max. number of iterations	1000	1000
Number of neighbors	5	6 (1 alpha, 5 beta)
rand	Produced randomly between 0 and 1	Produced randomly between 0 and 1
δ	Produced randomly by using Gaussian distribution	Produced randomly by using Gaussian distribution
Pa	Applied according to the quality of fitness	Applied according to the quality of fitness

can be made for the comparison between ACO, AMO, and ELAMO in Figure 5. For the analyzed TSP data sets, ELAMO gives better tour lengths than AMO and ACO. However, when AMO and ACO are compared, it is observed that the solution quality of ACO is better than that of AMO. These results can be an indication that the applied modifications have improved the solution quality of AMO. In order to observe the efficiency of selecting different neighborhood operators of ELAMO, Figure 6 is plotted. The number of beta animals is selected as 3, 4, 5, and a random value between 3 and 5. For all the analyzed TSP data sets, the optimum values or the values that are closest to the optimum tour lengths are obtained with the selection of beta as 5. Other best values are obtained when the beta value is selected as 4, random [3–5], and 3, respectively. Figures 7 and 8 are plotted to see the convergence rates of ELAMO with respect to the number of iterations and to observe the best routes found for the eil76 and kroA100 data sets. The efficiency of ELAMO can be seen clearly when the number of generations increases. In the comparison of ELAMO with GSA-ACS-PSOT and SONN from Table 3, it can be argued that the results obtained with ELAMO are of good quality and generally better or even more competitive than the results provided by the other algorithms for the selected TSP instances. Moreover, the obtained results imply that the solution quality of ELAMO is better than that of SONN for all the test problems except fl1400. However, in the comparison of GSA-ACS-PSOT and ELAMO, the average and PDav (%) values of ELAMO were not as good as the values of GSA-ACS-PSOT, especially for the larger data sets, but they were still considerable.

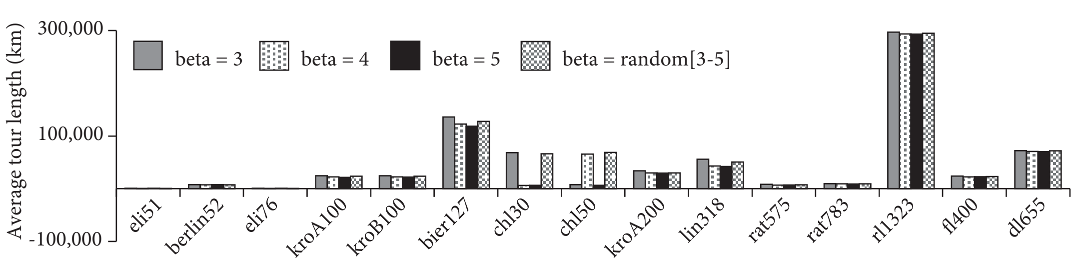
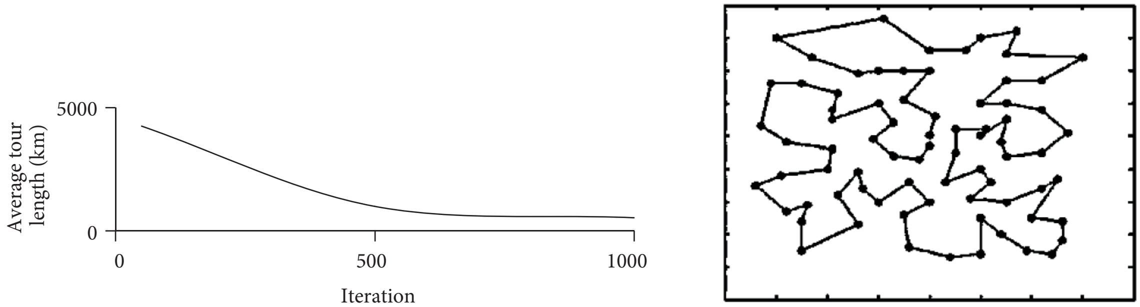


Figure 6. Effects of different neighborhood selection on average tour length in ELAMO.

5. Conclusion

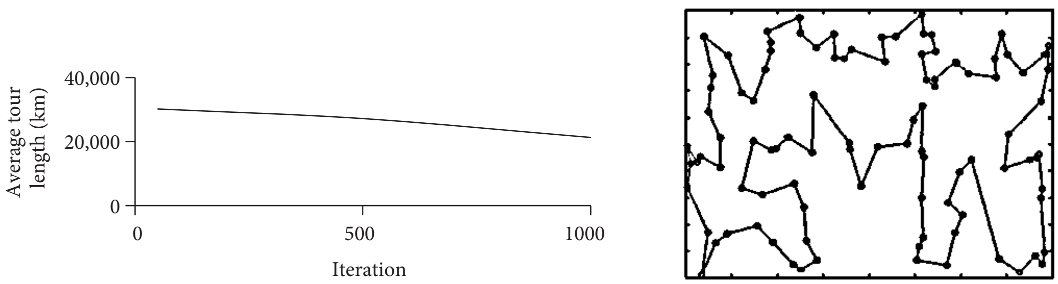
Considering the satisfactory performance of AMO in solving complex optimization problems, an improved algorithm is developed by reconstructing the neighborhood. The new algorithm, ELAMO, is specifically adapted



(a) Convergence graph of eil79.

(b) The best route for eil76.

Figure 7. Convergence graph of average tour length and the best route found by ELAMO for eil76: a) convergence graph of eil76, b) best route for eil76.



(a) Convergence graph of kroA100.

(b) The best route for kroA100.

Figure 8. A convergence graph of average tour length and the best route found by ELAMO for kroA100: a) convergence graph of kroA100, b) best route for kroA100.

to solve various TSPs. The experimental results show the success of the ELAMO algorithm compared to standard AMO. In the comparison of ELAMO to several algorithms, namely GA, PSO, GSA-ACS-PSOT, SONN, and ACO, it is clearly seen that ELAMO has improved the solution quality of AMO and has produced better or even more competitive values than the compared algorithms. The advanced performance of ELAMO is due to the selection of group leaders and the tendency of animals in a herd to migrate according to their leaders' positions. As a consequence of this movement, better solutions are explored by the animals.

On the other hand, the algorithm's performance is strongly dependent on the leaders' positions. The main improvement of AMO creates relative independency of animals, in such a way that they do not rely on their close neighbors but rather only on their leaders.

Further studies should investigate the diversification and intensification characteristics of ELAMO to reduce dependency on leaders' positions. According to the promising results attained in this study, the ELAMO algorithm is highly recommended for solving complex problems from different areas.

References

[1] Robinson J. On the Hamiltonian Game: A Traveling Salesman Problem. No. RAND/RM-303. Arlington, VA, USA: US Department of Defense, 1949.

[2] Geng X, Chen Z, Yang W, Shi D, Zhao K. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. Appl Soft Comput 2011; 11: 3680-3689.

[3] Chen SM, Chien CY. Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. Expert Syst Appl 2011; 38: 14439-14450.

[4] Albayrak M, Allahverdi N. Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. Expert Syst Appl 2011; 38: 1313-1320.

[5] Peker M, Şen B, Kumru PY. An efficient solving of the traveling salesman problem: the ant colony system having parameters optimized by the Taguchi method. Turk J Elec Eng & Comp Sci 2013; 21: 2015-2036.

[6] Dorigo M, Gambardella LM. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE T Evol Comput 1997; 1: 53-66.

[7] Gündüz M, Kıran MS, Özceylan E. A hierarchic approach based on swarm intelligence to solve the traveling salesman problem. Turk J Elec Eng & Comp Sci 2015; 23: 103-117.

[8] Ouaarab A, Ahiod B, Yang XS. Discrete cuckoo search algorithm for the travelling salesman problem. Neural Comput Appl 2014; 24: 1659-1669.

[9] Karaboga D, Gorkemli B. A combinatorial artificial bee colony algorithm for traveling salesman problem. In: IEEE 2011 International Symposium on Innovations in Intelligent Systems and Applications, 15–18 June 2011; İstanbul, Turkey. New York, NY, USA: IEEE. pp. 50-53.

[10] Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE T Evol Comput 1997; 1: 67-82.

[11] Li X, Zhang J, Yin M. Animal migration optimization: an optimization algorithm inspired by animal migration behavior. Neural Comput Appl 2014; 24: 1867-1877.

[12] Ma M, Luo Q, Zhou Y, Chen X, Li L. An improved animal migration optimization algorithm for clustering analysis. Discrete Dyn Nat Soc 2015; 2015: 194792.

[13] Zhou Y, Luo Q, Ma M, Qiao S, Bao Z. An animal migration optimization algorithm for constrained engineering optimization problems. J Comput Theor Nanosci 2016; 13: 539-546.

[14] Ülker ED, Ülker S. Antenna design using animal migration optimization algorithm. IET J Eng 2016; 1: D23J.

[15] Çunkaş M, Özsağlam M.Y. A comparative study on particle swarm optimization and genetic algorithms for traveling salesman problems. Cybern Syst 2009; 40: 490-507.

[16] Reinelt G. TSPLIB—A traveling salesman problem library. ORSA J Comput 1991; 3: 376-384.

[17] Özcan E, Erentürk M. A brief review of memetic algorithms for solving Euclidean 2D traveling salesrep problem. In: 2004 Turkish Symposium on Artificial Intelligence and Neural Networks; 10–11 June 2004; İzmir, Turkey. pp. 99-108.

[18] Masutti TAS, Castro LN. A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem. Inf Sci 2009; 179: 1454-1468.