

Finding the best tour for travelling salesman problem using artificial ecosystem optimization

Quyên Thi Nguyen, Minh-Phung Bui

Faculty of Information Technology, Van Lang University, Ho Chi Minh City, Vietnam

Article Info

Article history:

Received Aug 25, 2020

Revised Apr 14, 2021

Accepted Apr 26, 2021

Keywords:

2-opt algorithm
Artificial ecosystem optimization
Organisms
Shortest tour
TSP

ABSTRACT

This paper presents a new method based on the artificial ecosystem optimization (AEO) algorithm for finding the shortest tour of the travelling salesman problem (TSP). Wherein, AEO is a newly developed algorithm based on the idea of the energy flow of living organisms in the ecosystem consisting of production, consumption, and decomposition mechanisms. In order to improve the efficiency of the AEO for the TSP problem, the 2-opt movement technique is equipped to enhance the quality of the solutions created by the AEO. The effectiveness of AEO for the TSP problem has been verified on four TSP instances consisting of the 14, 30, 48 and 52 cities. Based on the calculated results and the compared results with the previous methods, the proposed AEO method is one of the effective approaches for solving the TSP problem.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Quyên Thi Nguyen

Faculty of Information Technology

Van Lang University

45 Nguyen Khac Nhu street, Co Giang ward, District 1, Ho Chi Minh city, Vietnam

Email: quyen.nt@vlu.edu.vn

NOMENCLATURE

N_{city}	Number of cities
N	Number of tours
down, up	Lower and upper limits of each city
$\theta_j, \theta_1, \theta_2, \theta_3$	Random number in [0, 1]
θ_4	Random number in [1, 2].
k, k_{max}	Number of current iterations and maximum number of iterations
$(x_i, y_i), (x_{i+1}, y_{i+1})$	Coordinates of the i^{th} and j^{th} cities
$L(tour)$	Length of the tour
$tour_{best}$	Best tour
L_{best}	Shortest tour

1. INTRODUCTION

The travelling salesman problem (TSP) problem is one of the famous optimization problems that aims to find the shortest tour among cities. The result of the TSP problem is the path that has the shortest length. Wherein, the best tour includes the starting and ending cities and passing cities, but all of them appear only once in the tour. The idea of the TSP problem has been applied in many fields such as paper cutting, computer wiring, routing, scheduling, social networking [1], [2]. However, the TSP problem is a complex problem with the number of feasible solutions up to $n!$ for the problem with n cities. Therefore, it is required to have effective solutions for finding the optimal solution for the problem.

From the first time proposed in 1970 [3], the TSP problem has been solved by difference approaches consisting of the exacting methods such as branch-and-cut [4], branch-and-bound [5], Lagrangian [6] and the methods relied on the metaheuristic algorithms such as genetic algorithm (GA) [7]-[9], particle swarm optimization (PSO) [10]-[12], ant colony optimization (ACO) [13]-[16] cuckoo search (CS) [17], [18]. Generally, each group of approaches has certain advantages and limitations. For the exacting methods, the optimal solution can be obtained but it usually takes long running time while using the optimization algorithms, the near optimal solution can be gained with the relatively short computation time [19]. In addition, a point that deserves attention when using the optimization algorithms is the influence of control parameters on the obtained results. For each population-based optimization algorithm, the main control parameters to be set are the population size and the maximum number of iterations. These two parameters are usually chosen based on the size of the problem.

However, many algorithms need to adjust additional parameters other than the above aforementioned parameters such as GA, PSO and CS. In GA, the rates of selection and mutation needs to be set or in PSO the proportional coefficients related to the velocity of the individual must also be chosen, while the rate of mutation in CS has also to determine before executing to find the optimal solution for the TSP problem. Furthermore, choosing inappropriate values of these parameters can affect the calculation results. Therefore, when using these methods for the specific optimization problems, examining the effect of these parameters is a requirement that cannot be almost ignored. In this aspect, to eliminate the effects of additional parameters, using the algorithms with no additional control parameters is one of the solutions that deserves attention. In addition, an algorithm is not able to have the highly performance for all of problems. It can work well for a given problem, but it may get a bad result for another ones. Therefore, adding new approaches to difficult optimization problems like TSP should also be welcome.

This paper shows a new method for solving the TSP problem based on artificial ecosystem optimization (AEO). The AEO is inspired from the idea of the flow of energy in living organisms including production, consumption, and decomposition [20]. In comparisons with other algorithms, AEO has no additional control parameters and its performance compared to other methods has been demonstrated over the benchmark functions [20]. In order to improve the efficient of AEO for the TSP problem, the 2-opt local search technique has been equipped to adjust the created solutions. The proposed AEO method is used to find the shortest path of the instances including 14, 30, 48 and 52 cities. The obtained results are compared with the published studies. The novelty and the contribution of the paper can be summarized as:

- AEO is the first proposed for solving the TSP problem.
- The 2-opt local search technique has been added for AEO to improve quality of the candidate solutions.
- The effectiveness of AEO has validated on the difference instances consisting of 14, 30, 48 and 52 cities.
- The proposed AEO method is effective for the TSP problem compared with some previous methods in the literature.

The paper is organized as follows: The introduction is discussed in this section. The TSP problem is presented in section 2. The application of AEO for the TSP problem is shown in section 3. Section 4 presents the experiment results of AEO for the instances. Finally, the conclusion is demonstrated.

2. TRAVELLING SALESMAN PROBLEM

Given N_{city} cities and the coordinates of each city, the tour length among cities $tour = [c_1, c_2, \dots, c_{N_{city}}]$ is determined as (1):

$$L(tour) = \sum_{i=1}^{N_{city}-1} d(c_i, c_{i+1}) + d(c_n, c_1) \quad (1)$$

where, $d(c_i, c_{i+1})$ is the Euler distance between the i^{th} and $(i + 1)^{\text{th}}$ cities that is calculated as (2):

$$d(c_i, c_{i+1}) = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \quad (2)$$

3. APPLICATION AEO FOR FINDING THE SHORTEST TOUR OF THE TSP

3.1. Solution description

In order to find the shortest tour for the TSP problem, each individual in the population is treated as a tour and each variable of the individual represents for a city. However, since each city can only be visited once or in other words, each city can only appear once in the candidate solution. Therefore, if each city is represented directly by the value of a variable, the random initialization and solution generation processes will not guarantee to create feasible variables. Therefore, in this study, each tour will be encoded as (3):

$$tour_i = [c_j]; j = 1, 2, \dots, N_{city}; i = 1, 2, \dots, N \quad (3)$$

where, $tour_i$ is the i^{th} tour in the population with N candidate tours. c_j is the real value of the j^{th} control variable in the range of $[dow, up]$. Noted that, the limit of all variables of the $tour_i$ is the same.

To determine the corresponding city of the variable c_j , the variables of the $tour_i$ are sorted based on their value. At that time, their order in the array that is sorted in ascending order represent the cities. For example the $tour_i$ consisting 6 cities has the values of $tour_i = [1.5, 6.3, 4.6, 5.7, 8.9, 4.0]$, the real tour of the $tour_i$ is determined by $real_tour_i = [1, 6, 3, 4, 2, 5]$. In which, the 1st city is assigned for the first variable because it has the smallest value in the $tour_i$. The second variable of the $tour_i$ has the highest value so its corresponding value in the $real_tour_i$ is 6. This rule is applied for the rest of the variables.

3.2. Initialization

In order to find the shortest tour, the population of tours of AEO is randomly created as (4):

$$tour_{i,j} = dow + \theta_j \cdot [up - dow]; j = 1, 2, \dots, N_{city}; i = 1, 2, \dots, N \quad (4)$$

where, $tour_{i,j}$ is the variable that represents for the j^{th} city in the i^{th} tour. Based on the $tour_i$, the value of the $real_tour_i$ is determined as (5):

$$[tour_{i,sort}, real_tour_i] = sort(tour_i) \quad (5)$$

where, $sort$ is a function that sorts the array in ascending order. The equation (5) shows that the index of the variables in the tour is the corresponding city while $tour_{i,sort}$ is the value of the variable in $tour_i$ sorted in ascending order.

3.3. Move in the search space of the TSP

Since the coordinates of the cities are fixed, the search space of the control variables in the TSP problem is the order of the cities. In this paper, to create high quality solutions, AEO's solution generation process is combined with 2-opt technique [21]. The main idea of this technique is to cut the two edges of the tour and reconnect the two parts of the tour. If the length of the two lines connecting the two parts of the tour is less than the length of the two edges removed, the new tour will be selected to replace the current one. The detailed implementation steps of the 2-opt technique for creating a new solution are described by using pseudo-code.

Algorithm 1: The 2-opt procedure pseudo code

```

1: Input: The current tour
2: Set  $dx_{min} = 0$ 
3: For  $i = 1$  to  $N_{city}$  do
4:   Set  $n_1 = i$  and  $m_1 = i - 1$ 
5:   If  $m_1 = 0$  then
6:     Set  $m_1 = N_{city}$ 
7:   End if
8:   For  $j = i + 2$  to  $N_{city}$  do
9:     Set  $n_2 = j$  and  $m_2 = n_2 - 1$ 
10:    Determine  $dx = d(m_1, m_2) + d(m_1, m_2) - d(m_1, n_1) - d(m_2, n_2)$ 
11:    If  $dx < dx_{min}$  then
12:      Set  $dx_{min} = dx$ 
13:      Save  $m_1, n_1, m_2$  and  $n_2$  as the positions to swap the current tour
14:    End if
15:  End for  $j$ 
16: End for  $i$ 
17: Generate the new tour by swapping the current tour  $tour(n_1:m_2) = reverse(tour(m_2:n_1))$ 
18: Output: The new tour

```

After performing the 2-opt procedure for adjusting the $real_tour_i$, the candidate tour $tour_i$ is updated as (6):

$$tour_i = tour_{i,sort}(real_tour_i) \quad (6)$$

Each tour created and updated using the 2-opt technique is evaluated by the fitness function using the corresponding $real_tour_i$ and (1) to calculate the length of the tour. Then the best so far tour $tour_{best}$ with the shortest length L_{best} is determined.

3.4. Generation of new tours

In the process of finding the best tour, AEO has two times of generating new population of tours. Wherein, in the first new population creation, the first new tour is generated as (7):

$$tour_{1,new} = \left[1 - \left(1 - k/k_{max}\right)\right] \cdot \theta_1 \cdot tour_{best} + \left(1 - k/k_{max}\right) \cdot tour_r \quad (7)$$

where, $tour_{1,new}$ is the first tour in the new population. $tour_r$ is a randomly tour created by using (4).

The new tours from 2nd to Nth are produced by using one of three ways as described in (8), (9) or (10). The probability that each equation used to create a new solution is equal to 1/3.

$$tour_{i,new} = tour_i + \lambda_1 \cdot (tour_i - tour_1) \quad (8)$$

$$tour_{i,new} = tour_i + \lambda_1 \cdot (tour_i - tour_j) \quad (9)$$

$$tour_{i,new} = tour_i + \lambda_1 \cdot \theta_2 \cdot (tour_i - tour_1) + (1 - \theta_2) \cdot (tour_i - tour_j) \quad (10)$$

where, $tour_j$ is a random tour selected randomly in the current population. The coefficient λ_1 is calculated as (11):

$$\lambda_1 = \frac{1}{2} \cdot \frac{N(0,1)_1}{N(0,1)_2} \quad (11)$$

where $N(0,1)$ is the standard distribution factor that has the average of 0 and the standard deviation of 1.

Then the new population of tours is validated by the fitness function (1). If the new tours are better than the corresponding ones, they are replaced for the current ones. The result is that the current population is updated. Whereas in the first new population creation, tours are produced using different techniques, at the second generation, all new tours are created by the same mechanism as (12):

$$tour_{i,new} = tour_{best} + 3 \cdot \lambda_2 \cdot (\beta_1 \cdot tour_{best} - \beta_2 \cdot tour_i); \quad (12)$$

where, $\lambda_2 \sim N(0,1)$. β_1 and β_2 are determined by (13):

$$\begin{cases} \beta_1 = \theta_3 \cdot \theta_4 - 1 \\ \beta_2 = 2 \cdot \theta_3 - 1 \end{cases} \quad (13)$$

Then the new population of tours is validated by the fitness function (1) and the current population is updated one more time. Furthermore, the best so far tour is also updated at the end of this stage.

3.5. The steps of AEO for finding the best tour

The process of generating new populations, adjusting relied on the 2-opt procedure and updating the current population are implemented until the maximum number of iterations reaches. The details of steps for finding the best tour are presented below.

Algorithm 2: The AEO pseudo code for TSP

```

1: Input: population size N, number of cities  $N_{city}$ , maximum number of iterations  $k_{max}$ 
2: For i = 1 to  $N_{city}$  do
3:   Produce the ith current tour by using (4)
4:   Determine the corresponding real tour by using (5)
5:   Adjust the real tour by using the 2-opt procedure in Algorithm 1
6:   Adjust the ith current tour by using (6)
7:   Calculate the fitness value of the ith current tour by using (1)
8: End for
9: Set the current iteration k = 0
10: While k <  $k_{max}$  do
11:   Produce the first new tour by using (7)
12:   Adjust the first new tour by using (5), the 2-opt procedure and (6), respectively
13:   For i = 2 to  $N_{city}$  do
14:     If rand < 1/3 then
15:       Produce the ith tour by using (8)
16:     Else if 1/3 < rand < 2/3 then
17:       Produce the ith tour by using (9)

```

```

18:         Else
19:             Produce the ith tour by using (10)
20:         End if
21:         Adjust the ith new tour by using (5), the 2-opt procedure and (6), respectively
22:         Calculate the fitness value of the ith new tour by using (1)
23:     End for
24:     Update the current tours by comparing the fitness value of the current tours and
the corresponding new tours in the population
25:     For i = 1 to Ncity do
26:         Produce the ith tour by using (12)
27:         Adjust the ith new tour by using (5), the 2-opt procedure and (6), respectively
28:         Calculate the fitness value of the ith new tour by using (1)
29:     End for
30:     Update the current tours by comparing the fitness value of the current tours and the
corresponding new tours in the population
31:     Update the best tour by comparing the fitness value of the best so far tour and
the best one in the current population
32: End while
33: Output: The best so far tour

```

4. SIMULATION RESULTS AND DISCUSSION

In order to evaluate the efficiency of the proposed method, the AEO method for finding the best tour is coded in MATLAB 2016a and run on the personal computer with CPU core i5, 2.4 GHz and 8 GB RAM. The instances consisting of Burma14, Oliver30, Att48 and Berlin52 referenced from TSPLIB [3] and [22] are used to find the shortest tour. The population size and maximum number of iterations of AEO for Burma14 and Oliver30 instances are set to {10, 50} and {20, 200}, respectively meanwhile these parameters are selected to {40, 300} for the Att48 and Berlin52 instances. The upper and lower values of each variable for all instances are set to [-1000, 1000]. The obtained results by AEO are compared to the previous methods in the literature.

The calculation results of the instances are presented in Table 1, the shortest tour length obtained by AEO for Burma14, Oliver30, Att48 and Berlin52 are 30.8785, 423.7406, 33522 and 7526 respectively, corresponding to the shortest tour depicted in Figure 1. Convergence characteristics of AEO in 50 independent runs for the instances described in Figure 2 shows that the mean convergence curves are very close to the minimum convergence characteristics. This shows that AEO is an approach worth considering in finding the best tour to the TSP problem.

Table 1. The best tours obtained by AEO for the instances

Problem	The best tour	L_{min}
Burma14	6, 5, 4, 3, 14, 2, 1, 10, 9, 11, 8, 13, 7, 12	30.8785
Oliver30	9, 8, 7, 6, 5, 4, 3, 1, 2, 30, 29, 28, 27, 26, 24, 25, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10	423.7406
Att48	8, 1, 9, 40, 15, 12, 11, 13, 25, 14, 23, 3, 22, 16, 41, 34, 29, 2, 26, 4, 35, 45, 10, 24, 42, 5, 48, 39, 32, 21, 47, 20, 33, 46, 36, 30, 43, 17, 27, 19, 37, 6, 28, 7, 18, 44, 31, 38	33522
Berlin52	34, 35, 36, 39, 40, 37, 38, 48, 24, 5, 15, 6, 4, 25, 12, 28, 27, 26, 47, 14, 13, 52, 11, 51, 33, 43, 10, 9, 8, 41, 19, 45, 32, 49, 1, 22, 31, 18, 3, 17, 21, 42, 7, 2, 30, 23, 20, 50, 29, 16, 46, 44	7526

The comparison results among AEO with the previous methods in terms of maximum (L_{max}), minimum (L_{min}), mean (L_{mean}) tour lengths and its standard deviation (STD) in 50 runs are presented in Table 2. For the Burma14 instance, the best tour obtained by AEO is identical to that of PSO [11] but the mean fitness value in 50 runs of AEO is lower than that of PSO [11]. For the Oliver30 instance, the best solution gained by AEO is similar to that of PSO [11], DTSA [23] and artificial bee colony (ABC) [23] and better than that of ABC-RS [24]. In comparison with DTSA [23], the mean fitness value of AEO is 3.57 higher but this value is respectively 0.1531, 30.48 and 45.79 lower than that of PSO [11], ABC [23] and random swap ABC (ABC-RS) [24]. Also from the table, the best tour for the Att48 instance obtained by AEO is identical to list-based simulated annealing (LBSA) [25] and lower than that of discrete invasive weed optimization (DIWO) [26], simulated annealing based symbiotic organisms search (SOS-SA) [27] and hybrid among GA, PSO and ACO (GA-PSO-ACO) [28]. For the Berlin52 instance, the shortest tour gained by AEO is 16 lower than that of DIWO [26], LBSA [25], dynamic flying ant colony optimization (DFACO) [13], DTSA [23] and discrete symbiotic organisms search (DSOS) [29]. These results show the high efficiency of the proposed AEO method for solving the TSP problem.

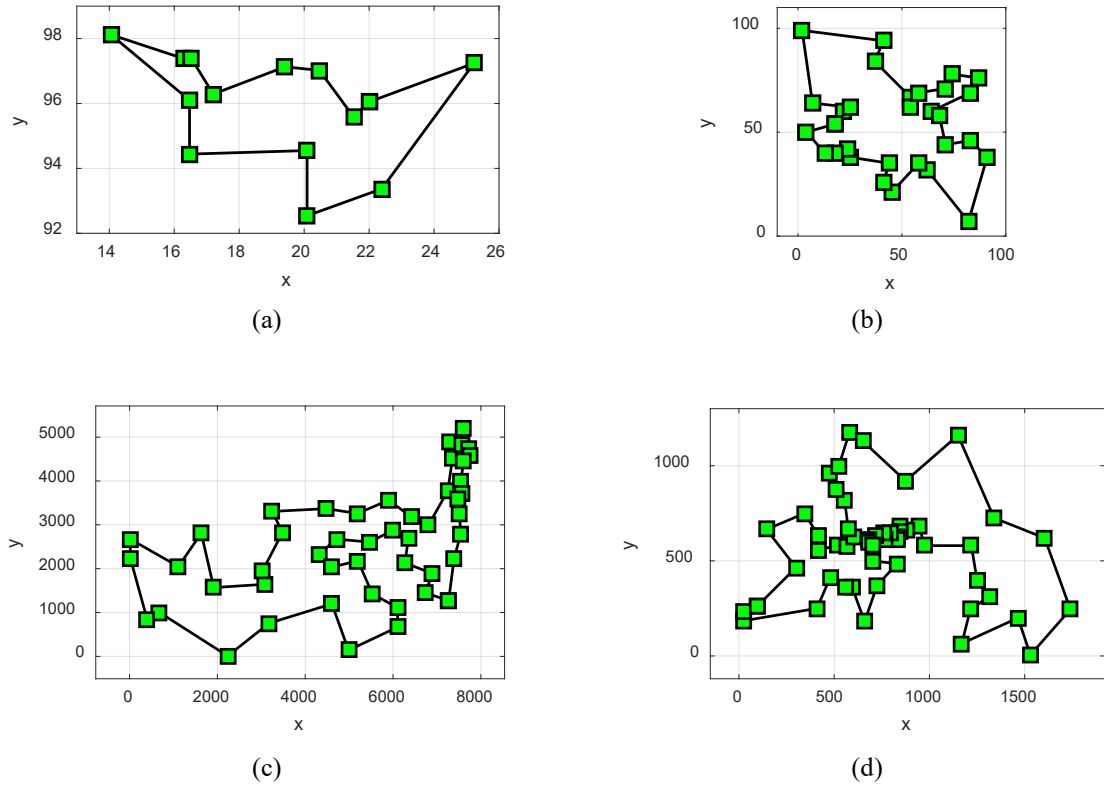


Figure 1. The best tours obtained by AEO for four instances: (a) Burma14, (b) Oliver30, (c) Att48, (d) Berlin52

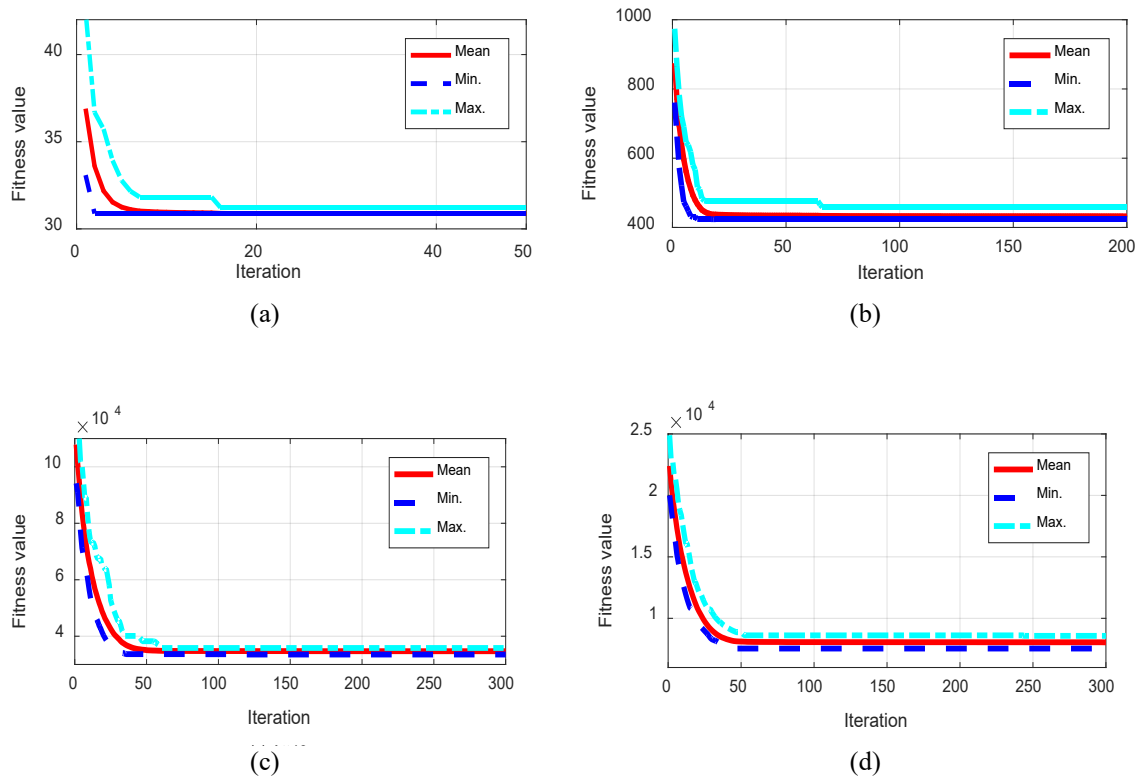


Figure 2. The convergence curves of the proposed method for four instances, (a) Burma14, (b) Oliver30, (c) Att48, (d) Berlin52

Table 2. The comparisons of AEO with other methods for the instances

Problem	Method	L_{max}	L_{min}	L_{mean}	STD	Time (s)
Burma14	AEO	31.2269	30.8785	30.8855	0.0492	0.5659
	PSO [11]	31.022	30.8785	30.9245	-	-
Oliver30	AEO	459.4105	423.7406	432.07	9.8856	7.2756
	PSO [11]	457.2354	423.7406	432.2231	-	-
	DTSA [23]	-	423.74	428.50	4.21	-
	ABC [23]	-	423.74	462.55	12.47	-
Att48	ABC-RS [24]	510.94	439.83	477.86	-	-
	AEO	35836	33522	34693.46	503	29.0044
	DIWO [26]	-	33523	-	-	-
	SOS-SA [27]	-	33523	33539.68	-	-
	LBSA [25]	-	33522	33536.6	-	-
	GA-PSO-ACO [28]	-	33524	33662	-	-
Berlin52	AEO	8488	7526	8064.94	241.6042	30.6872
	DIWO [26]	-	7542	-	-	-
	LBSA [25]	-	7542	7542	-	-
	DFACO [13]	7542	7542	7542	0	-
	DTSA [23]	-	7542	7761.6	62.8594	-
	DSOS [29]	7542	7542	7542	0	-

5. CONCLUSION

This paper presents the application of the AEO algorithm to the TSP problem. In which, to improve the search efficiency of AEO in finding the shortest tour for the instances, the 2-opt local search algorithm has been added for AEO to adjust solutions obtained from AEO. The effectiveness of AEO is verified on the instances consisting of Burma14, Oliver30, Att48 and Berlin52. Results obtained by the AEO are compared with the previous methods in the literature. The results show that, AEO has the better performance than PSO, ABC, ABC-RS, DIWO, SOS-SA, GA-PSO-ACO, LBSA, DFACO, DTSA and DSOS. Thus, AEO is a method that deserves attention in finding the shortest tour length to the TSP problem. For the future work, the AEO can be applied to larger-scale instances and real world applications.

REFERENCES

- [1] G. Campuzano, C. Obreque, and M. M. Aguayo, "Accelerating the Miller–Tucker–Zemlin model for the asymmetric traveling salesman problem," *Expert Systems with Applications*, vol. 148, 2020, Art. no. 113229, doi: 10.1016/j.eswa.2020.113229.
- [2] Y. Hu, Y. Yao, and W. S. Lee, "A reinforcement learning approach for optimizing multiple traveling salesman problems over graphs," *Knowledge-Based Systems*, vol. 204, 2020, Art. no. 106244, doi: 10.1016/j.knsys.2020.106244.
- [3] I. M. Oliver, Dj. Smith, and J. R. Holland, "Study of permutation crossover operators on the traveling salesman problem," in *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987, at the Massachusetts Institute of Technology, Cambridge, MA. Hillsdale, NJ: L. Erlbaum Associates, 1987.*
- [4] Y. Yuan, D. Cattaruzza, M. Ogier, and F. Semet, "A branch-and-cut algorithm for the generalized traveling salesman problem with time windows," *European Journal of Operational Research*, vol. 286, no. 3, pp. 849–866, 2020, doi: 10.1016/j.ejor.2020.04.024.
- [5] R. Salman, F. Ekstedt, and P. Damaschke, "Branch-and-bound for the Precedence Constrained Generalized Traveling Salesman Problem," *Operations Research Letters*, vol. 48, no. 2, pp. 163–166, 2020, doi: 10.1016/j.orl.2020.01.009.
- [6] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 2, pp. 231–247, 1992, doi: 10.1016/0377-2217(92)90138-Y.
- [7] X. Dong and Y. Cai, "A novel genetic algorithm for large scale colored balanced traveling salesman problem," *Future Generation Computer Systems*, vol. 95, pp. 727–742, 2019, doi: 10.1016/j.future.2018.12.065.
- [8] Y. Deng, Y. Liu, and D. Zhou, "An Improved Genetic Algorithm with Initial Population Strategy for Symmetric TSP," *Mathematical Problems in Engineering*, vol. 2015, 2015, doi: 10.1155/2015/212794.
- [9] J. Xu, L. Pei, and R. Z. Zhu, "Application of a genetic algorithm with random crossover and dynamic mutation on the travelling salesman problem," *Procedia Computer Science*, vol. 131, pp. 937–945, 2018, doi: 10.1016/j.procs.2018.04.230.
- [10] X. L. Xu, X. Cheng, Z. C. Yang, X. H. Yang, and W. L. Wang, "Improved particle swarm optimization for Traveling Salesman Problem," *Proceedings-27th European Conference on Modelling and Simulation, ECMS 2013*, 2013, pp. 857–862, doi: 10.7148/2013-0857.
- [11] Z. Yuan, L. Yang, Y. Wu, L. Liao, and G. Li, "Chaotic particle swarm optimization algorithm for traveling salesman problem," *Proceedings of the IEEE International Conference on Automation and Logistics, ICAL 2007*, 2007, pp. 1121–1124, doi: 10.1109/ICAL.2007.4338736.

- [12] M. Hoffmann, M. Mühlenthaler, S. Helwig, and R. Wanka, "Discrete particle swarm optimization for TSP: theoretical results and experimental evaluations," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6943 LNAI, pp. 416–427, 2011, doi: 10.1007/978-3-642-23857-4_40.
- [13] F. Dahan, K. El Hindi, H. Mathkour, and H. Alsalmán, "Dynamic flying ant colony optimization (DFACO) for solving the traveling salesman problem," *Sensors*, vol. 19, no. 8, 2019, doi: 10.3390/s19081837.
- [14] Ş. Gülcü, M. Mahi, Ö. K. Baykan, and H. Kodaz, "A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem," *Soft Computing*, vol. 22, no. 5, pp. 1669–1685, 2018, doi: 10.1007/s00500-016-2432-3.
- [15] S. Ebadinezhad, "DEACO: Adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem," *Engineering Applications of Artificial Intelligence*, vol. 92, 2020, Art. no. 103649, doi: 10.1016/j.engappai.2020.103649.
- [16] S. Bouzbita, A. El Afia, and R. Faizi, "The behaviour of ACS-TSP algorithm when adapting both pheromone parameters using fuzzy logic controller," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 5436–5444, 2020, doi: 10.11591/IJECE.V10I5.PP5436-5444.
- [17] A. Ouaraab, B. Ahiod, and X. S. Yang, "Discrete cuckoo search algorithm for the travelling salesman problem," *Neural Computing and Applications*, vol. 24, no. 7–8, pp. 1659–1669, 2014, doi: 10.1007/s00521-013-1402-2.
- [18] X. Ouyang, Y. Zhou, Q. Luo, and H. Chen, "A novel discrete cuckoo search algorithm for spherical traveling salesman problem," *Applied Mathematics and Information Sciences*, vol. 7, no. 2, pp. 777–784, 2013, doi: 10.12785/amis/070248.
- [19] E. Liao and C. Liu, "A hierarchical algorithm based on density peaks clustering and ant colony optimization for traveling salesman problem," *IEEE Access*, vol. 6, pp. 38921–38933, 2018, doi: 10.1109/ACCESS.2018.2853129.
- [20] W. Zhao, L. Wang, and Z. Zhang, "Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9383–9425, Jul. 2020, doi: 10.1007/s00521-019-04452-x.
- [21] G. A. Croes, "A Method for Solving Traveling-Salesman Problems," *Operations Research*, vol. 6, no. 6, pp. 791–812, 1958, doi: 10.1287/opre.6.6.791.
- [22] G. Reinelt, "TSPLIB. A traveling salesman problem library," *ORSA journal on computing*, vol. 3, no. 4, pp. 376–384, 1991, doi: 10.1287/ijoc.3.4.376.
- [23] A. C. Cinar, S. Korkmaz, and M. S. Kiran, "A discrete tree-seed algorithm for solving symmetric traveling salesman problem," *Engineering Science and Technology, an International Journal*, 2019, doi: 10.1016/j.jestech.2019.11.005.
- [24] M. S. Kiran, H. Işcan, and M. Gündüz, "The analysis of discrete artificial bee colony algorithm with neighborhood operator on traveling salesman problem," *Neural Computing and Applications*, vol. 23, no. 1, pp. 9–21, 2013, doi: 10.1007/s00521-011-0794-0.
- [25] S. H. Zhan, J. Lin, Z. J. Zhang, and Y. W. Zhong, "List-Based Simulated Annealing Algorithm for Traveling Salesman Problem," *Computational Intelligence and Neuroscience*, vol. 2016, 2016, doi: 10.1155/2016/1712630.
- [26] Y. Zhou, Q. Luo, H. Chen, A. He, and J. Wu, "A discrete invasive weed optimization algorithm for solving traveling salesman problem," *Neurocomputing*, vol. 151, no. P3, pp. 1227–1236, 2015, doi: 10.1016/j.neucom.2014.01.078.
- [27] A. E. S. Ezugwu, A. O. Adewumi, and M. E. Frîncu, "Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem," *Expert Systems with Applications*, vol. 77, pp. 189–210, 2017, doi: 10.1016/j.eswa.2017.01.053.
- [28] W. Deng, R. Chen, B. He, Y. Liu, L. Yin, and J. Guo, "A novel two-stage hybrid swarm intelligence optimization algorithm and application," *Soft Computing*, vol. 16, no. 10, pp. 1707–1722, 2012, doi: 10.1007/s00500-012-0855-z.
- [29] A. E. S. Ezugwu and A. O. Adewumi, "Discrete symbiotic organisms search algorithm for travelling salesman problem," *Expert Systems with Applications*, vol. 87, pp. 70–78, 2017, doi: 10.1016/j.eswa.2017.06.007.