



## Performance of Equilibrium Optimizer for the Traveling Salesman Problem

Quyên Thi Nguyen<sup>1\*</sup>      Minh-Phung Bui<sup>1</sup>

<sup>1</sup>*Faculty of Information Technology, Van Lang University, Ho Chi Minh City, Viet Nam*

\* Corresponding author's Email: [quyen.nt@vlu.edu.vn](mailto:quyen.nt@vlu.edu.vn)

---

**Abstract:** This paper presents a new method based on Equilibrium Optimizer (EO) algorithm that is inspired from the mass balance of a control volume for traveling salesman problem (TSP). For enhancing the efficiency of EO, the 2-opt movement algorithm is used to update the solution generated by EO. The efficiency of the proposed EO for the TSP problem has been compared with Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) on different instances consisting of the 14-city, 30-city, 48-city and 52-city. The calculated results show that for the large scale instances such as 48-city and 52-city, EO has found the better tour than PSO. In comparison with GA, EO has ability finding the best tour with the smaller mean and standard deviation. The comparisons with previous methods in literature have also demonstrated that EO has ability to search the better tour than other methods. Thus, the proposed EO can be a potential method for the TSP problem.

**Keywords:** Equilibrium optimizer, Genetic algorithm, Particle swarm optimization, Traveling salesman problem, 2-opt.

---

### 1. Introduction

The purpose of the traveling salesman problem (TSP) is to search the minimum length salesman's tour under the constraint that all given cities are visited only once by the salesman. The idea of TSP problem has been applied in different fields such as routing, computer wiring, paper cutting, social networking and scheduling [1-4]. TSP is a NP-complete problem with discrete control variables [1]. For  $n$  given cities, the number of possible tours is very large that may reach to  $n!$ . Thus, many researchers have focused their attention on finding effective solving methods to the TSP problem.

In order to solve TSP problem, there are two main method groups including exacting methods and metaheuristic algorithms. Typical methods in the first group must include methods such as branch-and-bound [5], branch-and-cut [6], lagrangian [7] and pseudo-polynomial time exact algorithm [8]. The common feature of these methods is that the description of the problem is quite complex. In addition, for large-scale problems, they can take a

long time to find the optimal results [9]. In order to overcome these disadvantages, more and more researchers have focused on applying metaheuristic algorithms for TSP problem. By using this group method, the description of the TSP problem is quite simple. In addition, the metaheuristic methods have been proven to be applicable to many large-scale problems [9]. Typical methods in this group include well-known methods such as Particle Swarm Optimization (PSO) [10-12], Genetic Algorithm (GA) [13-15], and Ant Colony Optimization (ACO) [16-19]. In addition, in recent years, many developed new algorithms have been successfully applied to TSP problems such as Cuckoo Search (CS) [20, 21], Discrete Spider Monkey Optimization (DSMO) [22], Harmony Search Algorithm (HSA) [23], Tree-Seed algorithm (TSA) [24], Artificial Bee Colony (ABC) [25], etc. When using optimization algorithms for the optimization problems in general and the TSP problem in particular, an issue that needs to be examined is the appropriateness of the algorithms for that problem because an algorithm can work well for the given problem, but it may work bad for another

problem [26, 27], Therefore, the study of applying new methods for TSP problem is also a problem worth considering.

For solving the TSP problem using the method-based on metaheuristic algorithms, it can be shown that the methods relied on evolutionary and swarm intelligence classes are used more commonly than other classes. GA is one of the most popular ones in the evolutionary algorithms. GA uses crossover mechanism to generate new solutions based on information of parental pairs that are selected naturally. While crossover maintains diversity or exploration of the search space, GA's mutation creates individuals that have partially different from their parents to exploit the search space. For the methods based on swarm intelligence, PSO is one of the most famous algorithms. In PSO, each particle fly in the search space for finding new solutions. This process relies on information on itself position and the location of the current best particle of the entire population. Equilibrium Optimizer (EO) is developed based on idea of the mass balance of a control volume [28]. Unlike GA and PSO, EO can be classified into a physics-based class. However, EO's new solution-finding mechanisms also have similar characteristics to PSO and GA. In EO, each particle's concentration is considered as a candidate solution. The particles' concentration is updated relied on three parts consisting of the equilibrium pool, the difference of concentration between the current particle and the equilibrium state and the generation rate. The equilibrium pool contains the best so far particles of the population. This technique is similar to the natural selection mechanism of GA. Meanwhile, the sharing of the equilibrium state information, which is also one of the best individuals of the population, with the other ones in the second component is similar to the PSO that relies on the current best one. In EO, this technique helps to explorer the search space. Furthermore, the generation rate makes a small adjustment to the new solutions that helps EO enhance the exploitation of the search space. This technique is considered to be similar to GA's mutation mechanism [28]. In [28], the performance of EO has demonstrated over 58 mathematical functions and three engineering problems. However, its effect on the TSP problem is still a question worthy of attention. Thus, this paper presents application of EO for solving the TSP problem. In order to fit with the TSP problem, each solution vector represents each candidate tour. The index vector that describes the arrangement in ascending order of the elements of a given solution vector is considered as the corresponding tour with the solution vector. In addition, the 2-opt algorithm has

been used to improve the generated solutions of EO. The effectiveness of EO is validated on different instances consisting of the 14-city, 30-city, 48-city and 52-city. Along with comparisons to other previous methods in literature, PSO and GA are also implemented for comparing with EO. The main contributions of this work can be listed as follows:

- (i) EO is the first presented for finding the optimal solution for the TSP problem.
- (ii) The 2-opt algorithm has been combined with EO to enhance the quality of the solutions.
- (iii) The efficiency of EO method has evaluated on the instances including the 14-city, 30-city, 48-city and 52-city.
- (iv) The efficiency of EO method has been compared with PSO and GA.
- (v) The proposed EO method has the better performance than PSO and GA as well as some other methods in the literature.

The organization of the paper is listed as follows: This section shows the introduction part. The TSP problem is defined in section 2. The application of EO for the TSP problem is demonstrated in section 3. Section 4 shows the numerical results and the conclusion part is presented in section 5.

## 2. TSP problem

The TSP problem requires to find the shortest possible tour, wherein each city is visited exactly once and go back to the origin city. The distances between two cities ( $l(c_i, c_{i+1})$ ) is determined as follows:

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

Where,  $i = 1, 2, \dots, n_c$  with  $n_c$  is the number of cities.  $(x_i, y_i)$  and  $(x_{i+1}, y_{i+1})$  are the coordinate of the  $i$ th and  $(i+1)$ th cities in the two-dimensional space.

Then, objective function of the TSP problem is defined as follows:

$$\min f = \sum_{i=1}^{n_c} l(c_i, c_{i+1}) + l(c_{n_c}, c_1) \quad (2)$$

Where,  $f$  is the tour length.  $l(c_{n_c}, c_1)$  is the distances between the  $c_{n_c}$ th and the 1st cities.

## 3. Application of EO for TSP problem

### Step 1: Initialization

In order to find the best tour for the TSP problem, each particle's concentration presents for a tour. At the beginning, the population of concentrations is created randomly as follows:

$$c_{i,j} = c_{j,min} + r_1(c_{j,max} - c_{j,min}) \quad (3)$$

Where,  $i = 1, 2, \dots, N$  and  $j = 1, 2, \dots, n_c$ .  $c_{i,j}$  is the  $j$ -th variable of the  $i$ -th concentration.  $c_{j,max}$  and  $c_{j,min}$  are high and low limits of the  $j$ -th variable, respectively.  $r_1$  is a random number in  $[0, 1]$ .  $N$  and  $i$ s number of concentrations in the population.

For solving the TSP problem, the high and low limits of each variable are chosen to 1000 and -1000, respectively. From the  $i$ -th concentration, the corresponding tour for the TSP problem is drawn by the following steps: The variables in the  $i$ -th concentration vector are sorted in ascending order. Then, the index vector that describes the arrangement of the elements of  $i$ -th concentration is considered as the tour corresponding to the  $i$ -th concentration. For example, the  $i$ -th concentration is  $\{3.5, 2.6, 4.9, 9.8\}$ , the corresponding tour will be  $\{2, 1, 3, 4\}$ . This technique ensures that each city is only visited once in each tour.

**Step 2: Improvement of the created tours**

In order to improve the quality of the generated solutions for the TSP problem, the 2-opt technique [29] is used to adjust the solutions that generated by EO. The 2-opt technique is implemented by cutting two edges of the current tour and connect two parts to form a new one as shown in Figure. 1. If two cut edges are longer than two new ones, the new tour will substitute for the current one. The detailed implementation steps of the 2-opt technique for creating a new solution are described by using pseudo-code as shown in Figure. 2.

From the created population of concentrations, the corresponding tours of the concentrations are validated the quality by calculating the tour length for the corresponding tour of each concentration. Then, the candidate concentrations that belong to anequilibrium pool are determined. It is noted that the equilibrium pool is a set of the best solutions of the population. It consists of five individuals, wherein the first four individuals are the best individuals in the

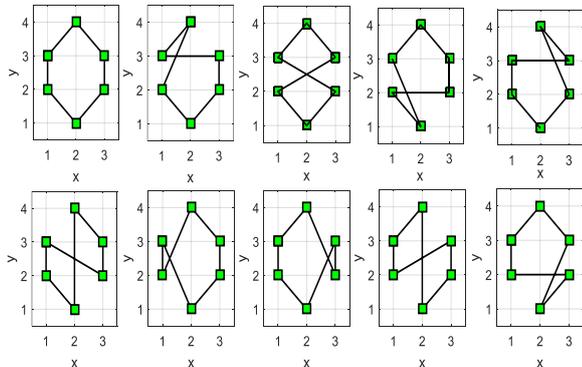


Figure. 1 Movement of 2-opt for generating new solutions

```

1. Input: The tour
2. Set  $lx_{min} = 0$ 
3. Set  $i = 0$ 
4. While  $i < n_c$  do
5.   Set  $i = i + 1$ 
6.   Set  $n_1 = i$  and  $m_1 = i - 1$ 
7.   If  $m_1 = 0$  then
8.     Set  $m_1 = n_c$ 
9.   End if
10.  Set  $j = i + 2$ 
11.  While  $j < n_c$  do
12.    Set  $i = i + 1$ 
13.    Set  $n_2 = j$  and  $m_2 = n_2 - 1$ 
14.    Calculate  $lx = l(m_1, m_2) + l(m_1, m_2) - l(m_1, n_1) - l(m_2, n_2)$ 
15.    If  $lx < lx_{min}$  then
16.      Set  $lx_{min} = lx$ 
17.      Save  $m_1, n_1, m_2$  and  $n_2$  as positions for swapping the tour
18.    End if
19.  End While  $j$ 
20. End While  $i$ 
21. Swap the tour  $tour(n_1:m_2) = reverse(tour(m_2:n_1))$ 
22. Output: The updated tour
    
```

Figure. 2 The 2-opt procedure pseudo code

population and the remaining individual is the mean of the four aforementioned ones.

**Step 3: Generate the new population of concentrations**

Each individual in the EO population is updated based on information of an individual chosen randomly from the equilibrium pool, an exponential term (E) and a generation rate (G) as follows:

$$c_i = c_{eq} + (c_i - c_{eq})E + \frac{G}{\lambda}(1 - E) \quad (4)$$

Where,  $c_{eq}$  is a concentration selected randomly from the equilibrium pool.  $\lambda$  is a random vector in  $[0, 1]$ .

In the above equation, E is the exponential vector that helps EO to poise between exploration and exploitation meanwhile G is the generation rate vector support EO to improve exploitation ability [28]. The E and G vectors are defined as follows:

$$E = a.sign(r - 0.5)(e^{-\lambda.t} - 1) \quad (5)$$

Where,  $a$  is a constant number chosen to 2.  $r$  is a random vector in  $[0, 1]$ .  $sign$  is the sign function that return 1, 0 and -1 if the corresponding argument is greater, equals and less than 0, respectively.  $t$  is a function that depends on the number of current iterations ( $it$ ) and the maximum number of iterations ( $it_{max}$ ) determined as follows:

$$t = \left(1 - \frac{it}{it_{max}}\right)^{\left(\frac{it}{it_{max}}\right)} \quad (6)$$

For G vector, it is determined as follows:

$$G = G_0 E \quad (7)$$

Where,  $G_0$  is an initial value of the decay process that is defined as follows:

$$G_0 = \delta(c_{eq} - \lambda c_i) \quad (8)$$

Where,  $\delta$  is the parameter of controlling the generation. It is determined as follows:

$$\delta = \begin{cases} 0.5rand(0,1); & \text{if } rand(0,1) \geq 0.5 \\ 0 & ; \text{otherwise} \end{cases} \quad (9)$$

As mentioned in the aforementioned part, using the information of the best particles in the equilibrium pool to create a new particle is similar to GA's natural selection mechanism. In addition, the information sharing of the best particles is in the second component of (4) which acts as a PSO, and the third component of (4) acts as a minor regulator to induce the mutation in the new particle like the GA mutation mechanism. This combination will help EO be able to better explore and exploit the search space than GA and PSO.

Step 4: Update the equilibrium pool

The new concentrations are checked and adjusted their boundaries to ensure the permitted range of them as follows:

$$c_{i,j} = \begin{cases} c_{j,max}; & \text{if } c_{i,j} > c_{j,max} \\ c_{j,min}; & \text{if } c_{i,j} < c_{j,min} \\ c_{i,j} & ; \text{otherwise} \end{cases} \quad (10)$$

Then, the new population of concentrations is evaluated the quality by calculating the tour length for the corresponding tour of each concentration. Then, the equilibrium pool is updated as follows:

$$c_i \text{ is assigned to } \begin{cases} c_{eq,1}; & \text{if } f(c_i) < f(c_{eq,1}) \\ c_{eq,2}; & \text{elseif } f(c_{eq,1}) < f(c_i) < f(c_{eq,2}) \\ c_{eq,3}; & \text{elseif } f(c_{eq,2}) < f(c_i) < f(c_{eq,3}) \\ c_{eq,4}; & \text{elseif } f(c_{eq,3}) < f(c_i) < f(c_{eq,4}) \end{cases} \quad (11)$$

Not only the concentrations in the equilibrium pool are updated according to  $c_i$ , but their corresponding fitness function value is also updated according to  $f(c_i)$  when the updated conditions in the above equation are met.

```

Set the parameters consisting of N,  $n_c$  and  $it_{max}$ 
Generate random population of concentrations using (3)
Update the population by executing the 2-opt for each concentration
Evaluate the fitness value of each concentration  $f(c_i)$  using (2)
Determine the four best concentrations and assign to  $c_{eq,1}$ ,  $c_{eq,2}$ ,  $c_{eq,3}$  and  $c_{eq,4}$ 
Calculate the  $c_{eq,5}$  by getting the mean of  $c_{eq,1}$ ,  $c_{eq,2}$ ,  $c_{eq,3}$  and  $c_{eq,4}$ 
Set  $it = 1$ 
While  $it < it_{max}$  do
    Determine  $t$  value using (6)
    For  $i = 1$  to N do
        Create random vector  $r$  and  $\lambda$ 
        Calculate the E vector using (5)
        Determine the  $\delta$  value using (9)
        Calculate the  $G_0$  vector using (8)
        Calculate the G vector using (7)
        Update the new concentration  $c_i$  using (4)
    End for
    For  $i = 1$  to N do
        Check and adjust the boundaries of the new concentration  $c_i$  using (10)
        Update the new concentration  $c_i$  using the 2-opt procedure
        Evaluate the fitness value of the new concentration  $c_i$  using (2)
        Update the  $c_{eq,1}$ ,  $c_{eq,2}$ ,  $c_{eq,3}$ ,  $c_{eq,4}$  using (11)
    End for
    Determine the  $c_{eq,5}$  by getting the mean of  $c_{eq,1}$ ,  $c_{eq,2}$ ,  $c_{eq,3}$  and  $c_{eq,4}$ 
    Increase the current iteration  $it$  by 1.
End while
Output: the optimal solution  $c_{eq,1}$  and its fitness value

```

Figure. 3 The EO procedure pseudo code for the TSP problem

Step 5: Check the stop condition for searching the optimal solution

The process of generating new concentrations and updating the equilibrium pool are executed until the number of current iterations reaches to the maximum number of iterations. The EO pseudo code for the TSP problem is presented in Figure. 3.

#### 4. Numerical results and discussions

The method of finding the best tour based on EO is developed on the MATLAB 2016 and executed on the computer with 8GB RAM, CPU core i5, 2.4GHz. In order to validate the efficiency of EO, two well-known methods based on PSO [30] and GA [31] are also implemented for comparing with the proposed EO method. Three methods are used to find the best tour for four instances including 14-city, 30-city, 48-city and 52-city [32,33]. For the 14-city and 30-city,

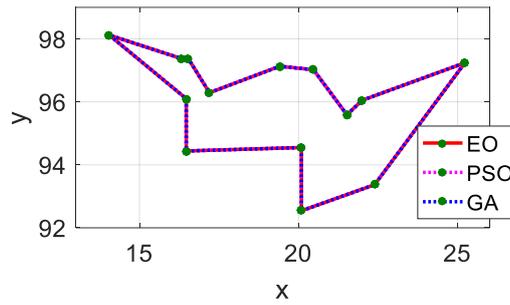


Figure. 4 The best tour for the 14-city problem

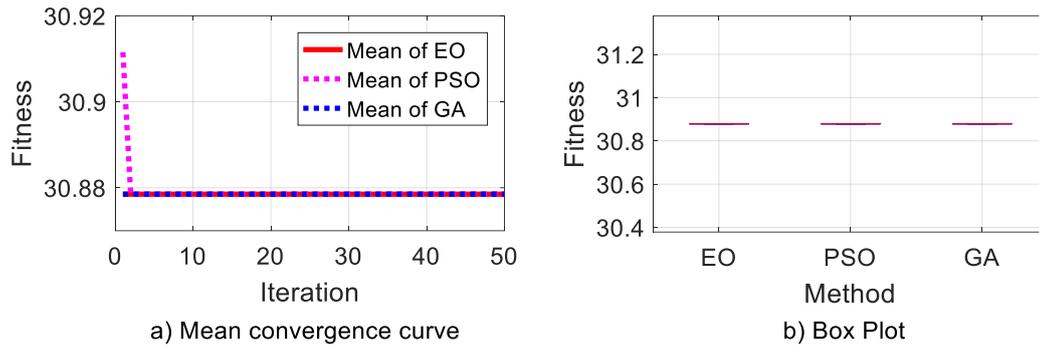


Figure. 5 The performance of EO, PSO and GA for the 14-city problem

48-city and 52-city [32,33]. For the 14-city and 30-city instances, the control parameters consisting of population size and maximum number of iterations for all methods are set to {10, 50}. For the 48-city and 52-city instances, they are set to {40, 500}. The extra control parameters of GA consisting of nature selection and mutation rates are set to default values of 0.5 and 0.2, respectively [31]. The constants C1 and C2 of PSO are also set to default values 2 [30].

The comparison results among EO, PSO and GA in 30 independent runs for the instances are shown in Table 1. and Figures. 4 to 11. The results show that for the 14-city case, all of three methods have

determined the shortest tour as shown in Figure. 4. In each run, all three methods have found the optimal solution corresponding to the mean value of the fitness function equal to the maximum one and the standard deviation (STD) is equal to zero. Convergence characteristics of the three methods and the box plot of three methods for the 14-city case obtained in Figure. 5 show that the efficiency of EO, PSO and GA for the 14-city instance is similar.

For the 30-city instance, in 30 independent runs, all three methods have also determined the best solution as shown in Figure. 6. However, the static results in Table 1. and Figure. 7 show that the effect of EO and

Table 1. The comparison results among EO, PSO and GA for the cities

Cities	Method	Max	Min	Mean	STD	Run time (s)	Best known tour
14	EO	30.8785	30.8785	30.8785	0	0.39948	30.8785
	PSO	30.8785	30.8785	30.8785	0	0.6651	-
	GA	30.8785	30.8785	30.8785	0	0.41146	-
30	EO	424.6918	423.7406	423.7723	0.1736	1.6349	423.7406
	PSO	482.749	423.7406	447.2276	20.6853	1.3526	-
	GA	423.7406	423.7406	423.7406	0	1.7036	-
48	EO	34195.8573	33523.7085	33793.6249	199.9462	137.8656	33522
	PSO	53782.8373	34473.2885	45512.1997	6645.0328	124.9417	-
	GA	34242.7578	33523.7085	33834.2566	231.1495	119.3604	-
52	EO	7962.1491	7544.3659	7618.7714	146.1061	165.912	7542
	PSO	12413.1922	7819.17327	10868.9774	1719.56595	140.3786	-
	GA	8043.6809	7544.3659	7644.7747	146.4246	133.1328	-

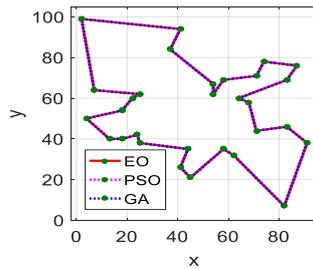


Figure. 6 The best tour for the 30-city problem

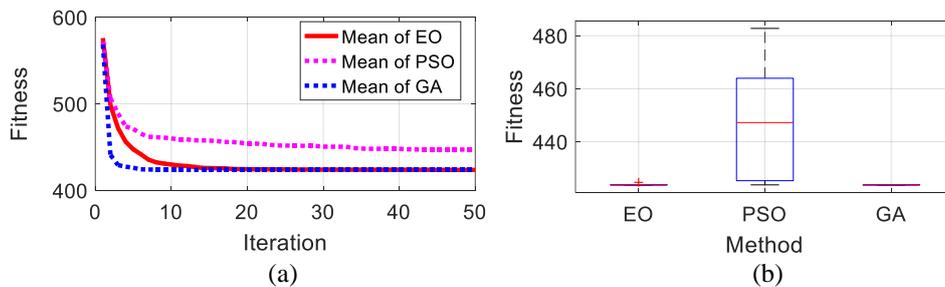


Figure. 7 The performance of EO, PSO and GA for the 30-city problem: (a) mean convergence curve and (b) box plot

GA is similar whereas the efficiency of EO superiors to that of PSO. Specifically, the mean and STD values of EO are 23.4553 and 20.5117 lower than that of PSO, respectively. For the 30-city instance, in 30 independent runs, all three methods have also determined the best solution as shown in Figure. 6. However, the static results in Tables 1. and Figure. 7 show that the effect of EO and GA is similar whereas the efficiency of EO superiors to that of PSO. Specifically, the mean and STD values of EO are 23.4553 and 20.5117 lower than that of PSO, respectively.

For the 48-city case, both EO and GA have determined the same tour as shown in Figure. 8 whereas PSO has found only the route that is 949.58 longer than that of EO and GA. In addition, the mean and STD of PSO are also much higher than that of EO and GA. Convergence characteristics and the box plot of three methods in Figure. 9 show that the efficiency of PSO is inferior to that of EO and GA. The comparison result between EO and GA shows that the efficiency of EO is slightly better than GA.

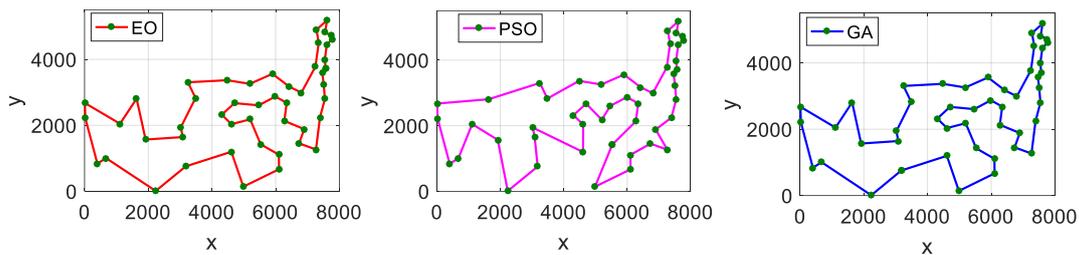


Figure. 8 The best tour of EO, PSO and GA for the 48-city problem

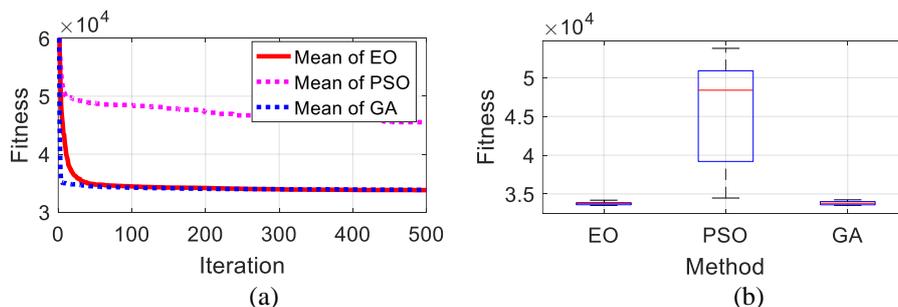


Figure. 9 The performance of EO, PSO and GA for the 48-city problem: (a) mean convergence curve and (b) box plot Table 2. The comparisons of AEO with other methods for the instances

Cities	Method	$f_{min}$	$f_{mean}$	STD
14	EO	30.8785	30.8785	0
	DSMO [22]	30.8785	-	-
	CPSO [11]	30.8785	30.9245	-
30	EO	423.7406	423.7723	0.1736
	CPSO [11]	423.7406	432.2231	-
	ABC [24]	423.74	462.55	12.47
	DTSA [24]	423.74	428.50	4.21
	ABC-RS [34]	439.83	477.86	-
48	EO	33523.7085	33793.6249	199.9462
	CPSO [11]	33534	34556	-
	GA-PSO-ACO [35]	33524	33662	-
	SOS-SA [36]	33523	33539.68	-
	DIWO [37]	33523	-	-
52	EO	7544.3659	7618.7714	146.1061
	SA [24]	8186.40	8983.80	380.10
	ACO [24]	8240.40	8777.60	267.11
	STA [24]	7544.40	8247.20	273.45
	DTSA [24]	7542	7761.6	62.8594
	GA-PSO-ACO [35]	7544.37	7544.37	-
	SOS-SA [36]	7540	7541	-

The better performance of EO is represented by the smaller mean and STD values of EO compared to GA. This demonstrates a good effect of EO for problems with a large number of cities. For the 52-city instance, the results in Table 1., Figure. 10 and Figure. 11 show that the EO efficiency is the best compared to PSO and GA. Specifically, in 30 runs, although both EO and GA determined the best solution, the mean value and STD of EO are smaller than that of GA whereas PSO have not been able to determine the optimal solution for problem. This result shows that EO is effective tool for the TSP problem.

Table 2. shows the compared results of EO with other methods in literature. For the 14-city instance, the result gained by EO is identical to that of DSMO [22] and Chaotic Particle Swarm Optimization (CPSO) [11] but the mean value of EO is smaller than that of CPSO [11]. For the 30-city case, the best tour

obtained by EO is better than that of Random Swap Artificial Bee Colony (ABC-RS) [34] and similar to that of CPSO [11], ABC [24], Discrete Tree-Seed Algorithm (DTSA) [24].

In addition, the mean fitness value gained by EO is the lowest in the compared methods. For the 48-city problem, the best tour gained by EO is identical to that of Simulated Annealing-Symbiotic Organisms Search (SOS-SA) [36], Discrete Invasive Weed Optimization (DIWO) [37] and better than that of CPSO [11] and hybrid GA-PSO-ACO [35]. For the 52-city case, the performance of EO is worse than DTSA [24] and SOS-SA [36] in term of the obtained best solution. However, the obtained solution of EO is similar to that of State Transition Algorithm (STA) [24] and GA-PSO-ACO [35] and better than that of Simulated Annealing (SA) [24], ACO [24]. These results show the high potential of the EO method for finding the best solution to the TSP problem.

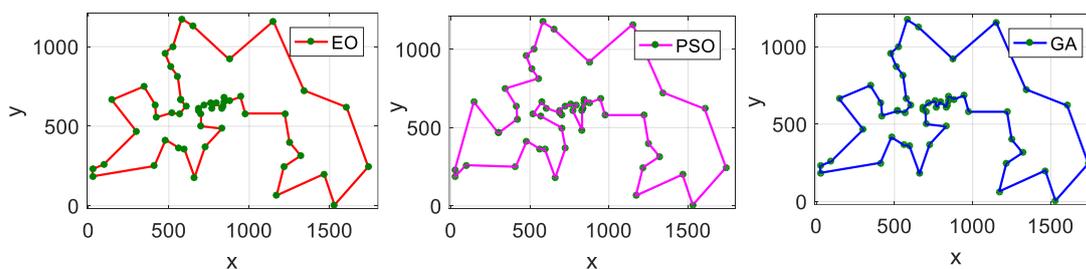


Figure. 10 The best tour of EO, PSO and GA for the 52-city problem

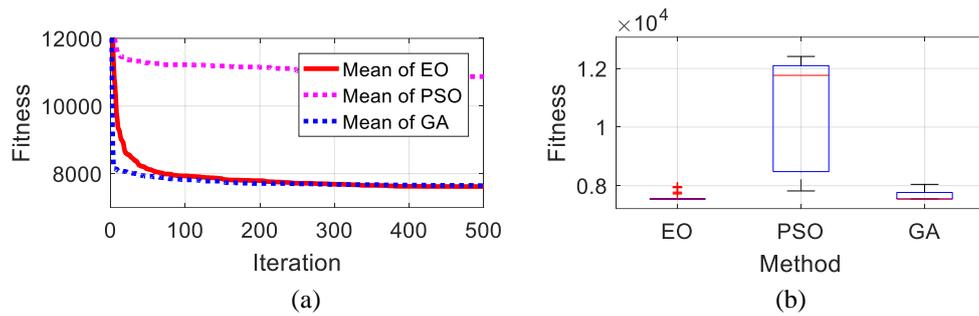


Figure. 11 The performance of EO, PSO and GA for the 52-city problem: (a) mean convergence curve and (b) box plot

## 5. Conclusion

This paper shows the application of EO for solving the TSP problem. In order to find the best solution for the TSP problem, the index vector that describes the arrangement of the variables of solution vector is considered as the corresponding tour. In addition, the generated new solutions are adjusted by the 2-opt procedure to enhance the quality of the solutions. The performance of EO is validated on four instances consisting of the 14-city, 30-city, 48-city and 52-city. In addition, PSO and GA are also implemented for comparing with the proposed EO method. The calculated results show that EO outperforms to PSO and has the better performance than GA in the large-scale instances. For the 48-city and 52-city problems, EO has found the tours which are 949.58 and 274.807 respectively shorter than those of PSO. In addition, the mean and STD values obtained by EO are {11718.6, 6445.09} for the 48-city and {3250.21, 1573.46} for the 52-city also lower than those of PSO. For GA, although both of GA and EO have found the best tour, the mean and STD values gained by EO is {40.6317, 31.2033} for the 48-city and {26.0033, 0.3185} for the 52-city lower than those of GA. In addition, EO has also obtained the better results than many previous methods in literature. Thus, EO is one of potential methods for solving the TSP problem. For future studies, EO can be implemented to find the optimal solution for the larger-scale TSP problem or real applications.

## Conflicts of Interest

The authors declare no conflict of interest.

## Author Contributions

Conceptualization, Q.T.N; methodology, Q.T.N; software, Q.T.N; validation, Q.T.N and M.P.B; formal analysis, Q.T.N and M.P.B; investigation, M.P.B; resources, M.P.B; data curation, Q.T.N and

M.P.B; writing—original draft preparation, Q.T.N; writing—review and editing, Q.T.N and M.P.B; visualization, Q.T.N and M.P.B;

## References

- [1] C. H. Papadimitriou, "The Euclidean travelling salesman problem is NP-complete", *Theoretical Computer Science*, Vol. 4, No. 3, pp. 237-244, 1977.
- [2] 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, pp. 113229, 2020.
- [3] 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, pp. 106244, 2020.
- [4] R. De Santis, R. Montanari, G. Vignali, and E. Bottani, "An adapted ant colony optimization algorithm for the minimization of the travel distance of pickers in manual warehouses", *European Journal of Operational Research*, Vol. 267, No. 1, pp. 120-137, 2018.
- [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.
- [6] 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.
- [7] 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.
- [8] L. Xu, Z. Xu, and D. Xu, "Exact and approximation algorithms for the min-max k-traveling salesmen problem on a tree", *European Journal of Operational Research*, Vol.

- 227, No. 2, pp. 284-292, 2013.
- [9] C. Jiang, Z. Wan, and Z. Peng, "A new efficient hybrid algorithm for large scale multiple traveling salesman problems", *Expert Systems with Applications*, Vol. 139, pp. 112867, 2020.
- [10] X. L. Xu, X. Cheng, Z. C. Yang, X. H. Yang, and W. L. Wang, "Particle swarm optimization for Traveling Salesman Problem", In: *Proc. of the 2003 International Conference on Machine Learning and Cybernetics (IEEE cat. no. 03ex693)*, Xi'an, China, Vol. 3, pp. 1583-1585, 2003.
- [11] Z. Yuan, L. Yang, Y. Wu, L. Liao, and G. Li, "Chaotic particle swarm optimization algorithm for traveling salesman problem", In: *Proc. of the IEEE International Conference on Automation and Logistics*, Jinan, China, pp. 1121-1124, 2007.
- [12] M. Hoffmann, M. Mühlenthaler, S. Helwig, and R. Wanka, "Discrete particle swarm optimization for TSP: theoretical results and experimental evaluations", In: *Proc. of International Conference on Adaptive and Intelligent Systems*, Springer, Berlin, Heidelberg, pp. 416-427, 2011.
- [13] 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.
- [14] Y. Deng, Y. Liu, and D. Zhou, "An Improved Genetic Algorithm with Initial Population Strategy for Symmetric TSP", *Mathematical Problems in Engineering*, Vol. 2015, pp. 212794, 2015.
- [15] 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.
- [16] F. Dahan, K. El Hindi, H. Mathkour, and H. Alsalman, "Dynamic flying ant colony optimization (DFACO) for solving the traveling salesman problem", *Sensors*, Vol. 19, No. 8, pp.1837, 2019.
- [17] 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.
- [18] S. Ebadinezhad, "DEACO: Adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem", *Engineering Applications of Artificial Intelligence*, Vol. 92, pp. 103649, 2020.
- [19] S. Bouzbita, A. El 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*, Vol. 10, No. 5, pp. 5436-5444, 2020.
- [20] A. Ouaarab, 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.
- [21] 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.
- [22] M. A. H. Akhand, S. I. Ayon, S. A. Shahriyar, N. Siddique, and H. Adeli, "Discrete Spider Monkey Optimization for Travelling Salesman Problem", *Applied Soft Computing*, Vol. 86, pp. 105887, 2020.
- [23] U. Boryczka and K. Szwarc, "The Harmony Search algorithm with additional improvement of harmony memory for Asymmetric Traveling Salesman Problem", *Expert Systems with Applications*, Vol. 122, pp. 43-53, 2019.
- [24] 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*, Vol. 23, No. 4, pp. 879-890, 2020.
- [25] V. Pandiri, and A. Singh, "An artificial bee colony algorithm with variable degree of perturbation for the generalized covering traveling salesman problem", *Applied Soft Computing*, Vol. 78, pp. 481-495, 2019.
- [26] D. H. Wolpert, and W. G. Macready, "No free lunch theorems for optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 67-82, 1997.
- [27] T. T. Nguyen, N. A. Nguyen, and T. L. Duong, "A novel method based on coyote algorithm for simultaneous network reconfiguration and distribution generation placement", *Ain Shams Engineering Journal*, Vol. 12, No. 1, pp. 665-676, 2020.
- [28] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm", *Knowledge-Based Systems*, Vol. 191, pp.105190, 2020.
- [29] G. A. Croes, "A Method for Solving Traveling-Salesman Problems", *Operations Research*, Vol. 6, No. 6, pp. 791-812, 1958.
- [30] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", In: *Proc. of IEEE International*

- Conference on Neural Networks*, Perth, WA, Australia, Vol. 4, pp. 1942-1948, 1995.
- [31] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, Second Edition. John Wiley & Sons, Inc., Hoboken, New Jersey, 2004.
- [32] I. M. Oliver, D.J. Smith, and J. R. Holland, "Study of permutation crossover operators on the traveling salesman problem", In: *Proc. of Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms*, Cambridge, MA, United States, pp. 224-230 1987.
- [33] G. Reinelt, "TSPLIB. A traveling salesman problem library", *ORSA journal on computing*, Vol. 3, No. 4, pp. 376-384, 1991.
- [34] 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.
- [35] 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.
- [36] A. E. S. Ezugwu, A. O. Adewumi, and M. E. Frıncu, "Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem", *Expert Systems with Applications*, Vol. 77, pp. 189-210, 2017.
- [37] 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, pp. 1227-1236, 2015.