Received: May 4, 2022. Revised: June 14, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Novel Method Based on Improved Coyote Optimization Algorithm for Searching the Best Tour for the Travelling Salesman Problem

Quyen Thi Nguyen[1*]

[1]*Faculty of Information Technology, School of Engineering and Technology, Van Lang University, Ho Chi Minh City, Vietnam*
* Corresponding author's Email: quyen.nt@vlu.edu.vn

Abstract: This paper presents a new method relied on improved coyote optimization algorithm (ICOA) for traveling salesman problem (TSP). COA is a recent recently metaheuristic algorithm that is inspired from the social life of coyote. To improve the performance of ICOA, the 2-opt algorithm is applied to adjust created new solutions. In addition, the swapping technique with varying exchange city number is also equipped to enhance the exploration and exploitation ability of ICOA. The efficiency of the ICOA is compared with COA on the 14-, 30-, 48-, 52-, 76- and 100-city instances. The simulated results show that ICOA reaches the better results than COA for the most instances. The average error of ICOA is 0.0057%, 0.2937%, 0.0613%, 5.8431% and 49.3940% lower than that of COA for the 30-, 48-, 52-, 76- and 100-city instances, respectively. Furthermore, the maximum, average and standard deviation objective values as well as the number of convergence generations of ICOA are also lower than those of COA. Moreover, ICOA has also achieved the optimal solution with higher quality compared to the previous approaches in literature. Consequently, ICOA is one of the methods worth considering for the TSP problem.

Keywords: Coyote optimization algorithm, Traveling salesman problem, 2-opt algorithm, Swapping technique, Improved coyote optimization algorithm.

1. Introduction

Finding the shortest tour length with given number of cities for salesman is the goal of the traveling salesman problem (TSP). The obtained tour has to ensure that all cities are visited only once. The initiative of the TSP has been used in many technical problems such as crystallography, wallpaper cutting, computer wiring [1], scheduling, social networks [2] and logistics problem [3]. However, the TSP is a NP-complete problem [4]. Furthermore, the number of possible solutions may reach to n! if there are n cities in each tour. Thus, researching the effective approaches for the TSP is attracted by many researches.

The TSP has been solved by various methods since it was first proposed in 1970 [5]. The used methods can be divided into the exacting and metaheuristic methods. The main property of exacting methods such as branch-and-cut [6], pseudo-polynomial exact algorithm [7] and branch-and-bound [8] is that the problem model is described complexly. Furthermore, they may take a lot of time for the large-scale TSP problems [9]. Thus, the TSP has been mainly solved by metaheuristic methods so far due to their simplicity of problem description and reaching high performance [9]. For the metaheuristic approaches, in addition to the popular approaches that already used such as particle swarm optimization (PSO) [10], genetic algorithm (GA) [11], simulated annealing (SA) [12] and ant colony optimization (ACO) [13], there are many recent methods such as cuckoo search (CS) [14], discrete tree-seed algorithm (DTSA) [15], symbiotic organisms search (SOS) [12], equilibrium optimizer [16] and artificial ecosystem optimization [17]. The issue of concern for using the metaheuristic approaches is the dependence on the control parameters of the algorithm to the obtained results. An inappropriate selection of control parameters may affect negatively to the obtained result quality. When using PSO for the TSP problem, the scaling factor of the velocity increments c_1 and c_2 must be set. In [18], the authors has proved that their different values have affected to the obtained results. By applying GA, mutation and crossover rates are also to be chosen. In [11], their values are set to 0.7 and 0.1. As using ACO for the TSP problem, the relative pheromone influence indexes α, β and γ are also to be set before executing the algorithm [13]. By using DTSA, the search tendency index has to be selected [15]. Similarly, the bad-solution portion of CS [14] or the initial temperature and cooling rate of SA [12] have to be set as applying CS and SA for the TSP problem. In this respect, some algorithms with no or few special control parameters such as SOS [12], EO [16] and AEO [17] will help users easily apply to the TSP problem. Therefore, choosing algorithms without special control parameters or researching to eliminate the influence of control parameters when using these algorithms for the TSP problem is a matter of concern. In addition, in recent years in order to improve the efficiency of metaheuristic algorithms for the TSP problem, many studies have focused on improving or combining methods to achieve high quality results for the TSP problem. For example, in [12] the combination of SOS and SA (SOS-SA) has gained the better results than SOS for the TSP problem. In [19], the combination of firefly algorithm and ACO has demonstrated higher performance over ACO for the TSP problem. Similarity, in the combination of GA, PSO, ACO (GA-PSO-ACO) has outperformed to GA and PSO [20]. In [21], the combination of harris hawk optimizer and ACO has gained the higher performance over PSO, GA and ACO for the TSP problem. In [22], the combination of ACO and GA (ACO-GA) has searched the optimal solution of the TSP problem faster than the original algorithms. Although the number of metaheuristic approaches for the TSP problem is abundant, it is clear that there is no quality method for all of problems. Based on the no-free-lunch theorem, an algorithm can achieve high performance for a given problem but it may not be still suitable when applied for other problems [23]. Therefore, researching to add new methods for the TSP problem still need to be encouraged.

Coyote algorithm (COA) is a recent recently metaheuristic algorithm that is inspired from the social life of coyote [24], wherein the social condition of each coyote is represented for a candidate solution of the optimization problem. The COA population is divided into small groups. The characteristic of each group is expressed through the leader and the trending individual of the group. New candidate solutions of each group are generated by interaction with the leader and trending coyotes of the group. Furthermore, the worst coyote in each group is also replaced by a new one in the search space. Moreover, the information exchange among groups also helps COA to avoid trapping to local optimal. The application of COA has been carried out for many problems such as electric network operation optimization [25], control water-pumping system [26] and photovoltaic model parameters extraction [27], etc. However, the efficiency of COA for different problems such as the TSP problem is also a question that needs to be clarified. In this paper, ICOA is adapted to determine the best tour length of the TSP. In which, the 2-opt algorithm is applied to modify the candidate solutions generated by COA. In addition, to improve the ICOA performance, the swapping technique with varying exchange city number is proposed for enhancing the exploration and exploitation of ICOA. The effectiveness of ICOA is compared with COA as well as previous algorithms in literature on different TSPs such as 14-, 30-, 48-, 52-, 76- and 100-city instances. Based on the achieved results, the contributions of this work can be summarized as follows:

(i) ICOA is the first proposed for the TSP problem.

(ii) The 2-opt and swapping techniques with variable exchange city number have been applied to improve ICOA performance for the TSP problem.

(iii) The robustness of ICOA has evaluated on the instances consisting of 14-, 30-, 48-, 52-, 76- and 100-city.

(v) The proposed ICOA has achieved the better results than COA and some previous algorithms in the literature for the TSP problem.

The rest paper is organized as follows: The TSP problem is described in section 2. The COA's application to the TSP problem is described in section 3. ICOA for TSP problem is shown in section 4. Section 5 shows the numerical results for the instances. The conclusion is presented in final section.

2. Travelling salesman problem

The distance between two cities of the tour is defined as follows:

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

Where, (x_i, y_i) and (x_{i+1}, y_{i+1}) are coordinates of the city i and $i+1$.

Then, the length of the whole tour with D cities is calculated as follows:

```
Input: The T_c^g, IT_c^g
Set ΔD_min = 0
For i = 1 to D do
    n_1 = i
    m_1 = i − 1
    If m_1 = 0 then
        m_1 = N_city
    End if
    For j = i + 2 to D do
        n_2 = j
        m_2 = n_2 − 1
        ΔD = d(m_1, m_2) + d(m_1, m_2) −
        d(m_1, n_1) − d(m_2, n_2)
            If ΔD < ΔD_min then
            ΔD_min = ΔD
            Save positions of m_1, n_1, m_2 and n_2
        End if
    End for j
End for i
Swap the current tour to create the new one:
IT_c^g (n_1: m_2) = reverse (IT_c^g (m_2: n_1))
Update the T_c^g
Output: The new tour T_c^g and IT_c^g
```

Figure. 1 The pseudo code of the 2-opt algorithm for adjusting the current tour

$$f(tour) = \sum_{i=1}^{D-1} d(c_i, c_{i+1}) + d(c_D, c_1) \quad (2)$$

Where, $d(c_D, c_1)$ is the distances between the city c_D and the first one.

3. Application COA for TSP

In this section, COA is adjusted for the TSP problem. Details of the COA steps are described as follows:

Step 1: Generate the initial current tours

In COA, the behaviour of each coyote is considered as a candidate solution of the optimization problem. The coyote population is divided into n_g groups, wherein there are n_c coyotes in each group. Thus, to solve the TSP using COA, each candidate tour is presented by the coyote's behaviour. It is created as follows:

$$T_c^g = T_l + rand\ (0,1).(T_h - T_l) \quad (3)$$

Where T_c^g is the solution c in group g with $c = 1,2,\ldots,n_c$ and $g = 1,2,\ldots,n_g$. T_h and T_l are the boundaries of solutions. For TSP, they are selected to [-1000, 1000] for all control variables.

The initialization process will generate real numbers. Thus, to get the real tour corresponding to each candidate solution, the solution T_c^g is sorted in ascending order and the index vector (IT_c^g) showing order of each control variable in the solution is considered as the corresponding tour of the solution T_c^g. For example, the T_c^g has value of [604.6, 414.3, -382.7, -418.1, -504, -652.7, 954.1, 706.3, 657, 644, 692.5, -966.6, 913.1, 344.3], the coresponding tour IT_c^g will be [12, 6, 5, 4, 3, 14, 2, 1, 10, 9, 11, 8, 13, 7].

To enhance the quality of the generated candidate tours, the 2-opt algorithm [28] is used to adjust the current tours. This technique is performed by cutting two edges of the tour and connecting two parts of the tour to create a new one. If the two new edges are shorter than the old ones, the old tour will be replaced by the new one. The pseudo code of the 2-opt algorithm is presented in Fig. 1. Then, the quality of each candidate tour (f_c^g) is calculated by using the fitness value as shown in Eq. (2). The best coyote (T_{best}) with the lowest tour length value (f_{best}) is gained.

Step 2: Generate new candidate tours

In COA, behaviour of coyotes is formed by interacting between the leader coyote behaviour and the tendency behaviour of the group. Thus, for TSP, the new tours are generated as follows:

$$new_{T_c^g} = T_c^g + rand\ (0,1).\left(T_{le}^g - T_{r1}^g\right) + rand\ (0,1).\left(T_{te}^g - T_{r2}^g\right) \quad (4)$$

Where, $new\_T_c^g$ is the new solution c in group g. T_{le}^g is the best tour in the group g. T_{te}^g is the tendency tour of the group g that is defined by the median of fitness values in the group g. T_{r1}^g and T_{r2}^g are random tours in the group g.

The new tours are adjusted by using the 2-opt algorithm and evaluated the quality by the using Eq. (2). Then, the current tours are updated by using the selection technique as follows:

$$T_c^g = \begin{cases} new\_T_c^g\ , if\ new\_f_c^g < f_c^g \\ T_c^g\ , \quad otherwise \end{cases} \quad (5)$$

$$f_c^g = \begin{cases} new\_f_c^g\ , if\ new\_f_c^g < f_c^g \\ fit_c^p\ , \quad otherwise \end{cases} \quad (6)$$

Where, $new\_f_c^g$ is the fitness value of the tour $new\_T_c^g$.

Step 3: Replace the worst tour in each group

In each coyote group, the worst coyote that is poorly adapted to the environment will be died and replaced by a new one (T_{new}^g). Thus, the new tour in each group is generated as follows:

1	2	3	4	5	6	7	8	9	10
swap(3,8)									
1	2	8	4	5	6	7	3	9	10

(a)

1	2	3	4	5	6	7	8	9	10
swap(3,8)									
1	2	8	4	5	6	7	3	9	10
swap(1,4)									
4	2	8	1	5	6	7	3	9	10

(b)

Figure. 2 The example of swapping technique: (a) swapping technique and (b) swapping technique with variable exchange city number

$$x_{new,j}^{g} = \begin{cases} x_{r1,j}^{g}, if\ rand < 1/D \\ x_{r2,j}^{g}, if\ rand < 1/D + 0.5(1-1/D) \\ x_{r,j}^{g}, otherwise \end{cases} \quad (7)$$

Where, $x_{new,j}^{g}$ is the control variable j with $j = 1,2,\dots,D$ of the new tour in the group g. D is number of cities of the TSP.

The new tour is adjust using the 2-opt algorithm and evaluated the quality by the using Eq. (2). Then, the selection technique as shown in Eqs. (5) and (6) is used to replace the worst tour in each group.

Step 4: Exchange of the tours among groups

In COA, the solutions in each group often move to a certain area in the search space. Thus, the information exchanging among the groups helps to avoid trapping to local optimal. This technique is formed based on the metaphor that occasionally coyotes may leave from their group to join another group. The probability of exchanging the solutions between two groups is defined as follows:

$$rand < 0.005{n_C}^2 \quad (8)$$

If Eq. (8) is satisfied, two groups of tours are chosen to exchange a tour as follows:

$$\begin{cases} g(r_{g,1}, r_{T,1}) = g(r_{g,2}, r_{T,2}) \\ g(r_{g,2}, r_{T,2}) = g(r_{g,1}, r_{T,1}) \end{cases} with\ r_{g,1} \neq r_{g,2} \quad (9)$$

Where $r_{g,1}$ and $r_{g,2}$ are random positions of chosen groups. $r_{T,1}$ and $r_{T,2}$ are random positions of chosen coyotes in the corresponding selected groups.

Finally, based on the new tours and their fitness value, the best tour T_{best} is updated again. The steps from 2 to 4 are executed until the current generation (CG) reaches to the maximum value (G_{max}).

4. Improved COA for TSP

In this section, the improved COA is presented to enhance the quality of candidate tours that are generated by COA as described in section 3. To improve the quality of candidate tours generated in Step 2 of COA, the tours will be updated one more time using the swapping technique [29, 15]. For swapping the tour, two random integer numbers from 1 to D are generated. Then, these positions in the tour will be swapped to form a new one. An example of swapping technique is presented in Fig. 2 (a).

It can be seen that the swapping technique only makes minor changes on the tour. It is not efficient to use the same swapping technique in every generation. Therefore, in this work the swapping technique with varying exchange city number is proposed for ICOA. Firstly, the positions for swapping is determined by the below equation:

$$\begin{cases} k_{1,\mathrm{i}} = round(0.5 + (D-1).rand) \\ k_{2,\mathrm{i}} = round(0.5 + (D-1).rand) \end{cases} k_{2,\mathrm{i}} \neq k_{1,\mathrm{i}} \quad (10)$$

Where, $k_{1,\mathrm{i}}$ and $k_{2,\mathrm{i}}$ are the swapping positions of the tour IT_c^g at the swapping ith with $i = 1, 2, \dots, n_{sw}$. n_{sw} is number of cities for swapping.

Then, the new tour is formed by exchanging the variable of between two positions as follows:

$$[IT_c^g(k_{1,\mathrm{i}}), IT_c^g(k_{2,\mathrm{i}})] = swap\left(IT_c^g(k_{1,\mathrm{i}}), [IT_c^g(k_{2,\mathrm{i}})\right) \quad (11)$$

Where, $IT_c^g(k_{1,\mathrm{i}})$ and $IT_c^g(k_{2,\mathrm{i}})$ are control variables k_1th and k_2th of the tour IT_c^g at the swapping ith. Swap is the function of swapping two elements.

As shown in Fig. 2 (b), the new tour is significantly different from the original tour with the number of swapping cities of two. The number of cities for swapping (n_{sw}) will be inversely proportional to the number of generations. In the earlier generations the number of swapped cities will increase to enhance the exploration ability of ICOA while this value will decrease as the number of generations increases to focus on the search space exploitation ability. Its mathematical model is described in as shown in Eq. (12).

$$n_{sw} = round\left(rand.n_{sw,max}.\frac{G_{max}-CG}{G_{max}}\right) \quad (12)$$

Where, $n_{sw,max}$ is the maximum number of cities

Input: The T_c^g, IT_c^g
Determine n_{sw} using (12)
For i = 1 **to** n_{sw} **do**
$k_1 = round(0.5 + (D-1).rand)$
$k_2 = round(0.5 + (D-1).rand)$
While $k_1 = k_2$ **do**
$k_2 = round(0.5 + (D-1).rand)$
End while
$temp = IT_c^g(:, k_1)$
$IT_c^g(:, k_1) = IT_c^g(:, k_2)$
$IT_c^g(:, k_2) = temp$
End for i
Update the T_c^g relied on IT_c^g
Output: The new tour T_c^g and IT_c^g

Figure. 3 The pseudo code of the swapping technique with variable exchange city number

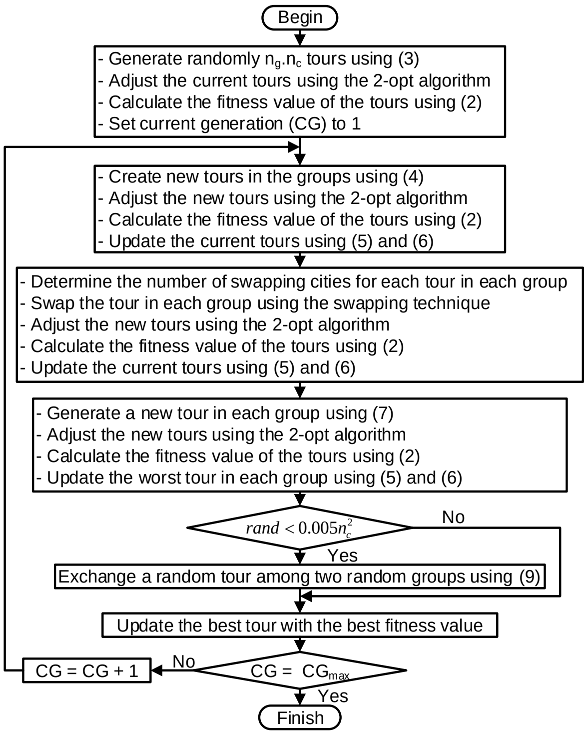

Figure. 4 The flowchart of ICOA for the TSP

for swapping that is limited to ten in this work.

The pseudo code of the swapping technique with input of the solution T_c^g and the index vector IT_c^g is given in Fig. 3. The whole steps of ICOA for the TSP is shown in Fig. 4.

5. Numerical results and discussions

The COA and ICOA for the TSP are implemented in Matlab 2016a and executed on the personal computer with 4GB memory and core i5 processer. The instances are selected from the TSP library consisting 14, 30, 48, 52, 76 and 100 cities for searching the shortest length [5], [30]. For the 14 and 30 cities, the control parameters of n_g, n_c and G_{max} are chosen to 2, 5 and 50, respectively. For the rest instances, they are set to 5, 8 and 300, respectively. Both of COA and ICOA are independently executed 30 times. To assess performance of COA and ICOA, the average error (AE) that shows the error between the average result (f_{av}) in 30 runs and the best-known result (f_{bk}) is used. The AE is defined as follows [31]:

$$AE = \frac{f_{av} - f_{bk}}{f_{bk}}.100\% \tag{13}$$

The comparison results of effectiveness between ICOA and COA for the TSPs are presented in Table 1. For the 14-city instance, both of ICOA and COA have determined the best tour in each execution. This is represented by the equality of the maximum (f_{max}), optimal (f_{op}), average and best-known values. However, the number of average convergence generations (G_{av}) of ICOA is smaller than that of COA. For the 30-city instance, ICOA and COA have also found the best solution in 30 runs, but the average fitness value in 30 runs of ICOA is lower than that of COA. The AE value of ICOA is 0.0013% that is 0.0057% less than that of COA. In addition, the standard deviation (std) of the fitness function value in runs of ICOA is also smaller than that of COA.

The superior performance of ICOA over COA is evident on the larger TSPs such as 48-, 52-, 76- and 100-city instances. For the 48-city instance, the optimal value obtained by both methods is the same, but the average fitness value of ICOA is smaller than that of COA. The AE value of ICOA is 0.0417% while this value of COA is 0.3354% that is 0.2937% higher than that of ICOA. Similarly, for the 52-city instance, the AE value of the ICOA is 0.0314% that is 0.0613% lower than that of the COA. The lower std values of ICOA indicate the stability and reliability of ICOA over COA in each run for the TSP problem. In addition, the number of convergence generations for both instances of 48- and 52-city of ICOA is lower than that of COA. This shows the effectiveness of ICOA in finding the best tour for the TSP problem. In term of computation time, ICOA takes longer than that of COA for the same problem. For the 76- and 100-city instances, all the statistics indexes of ICOA are better than those of COA. It is worth noting that the AE value of ICOA for the 76-city instance is 0.6066%, which is 5.8431% lower than that of COA. Similar to the 76-city instance, the AE value of the ICOA for the 100-city instance is much lower than that of the COA. While the AE of COA is 49.6668%, this value obtained by ICOA is only 0.2728%.

Received: May 4, 2022. Revised: June 14, 2022.

Table 1. Results of COA and ICOA for TSPs

Instance	f_{bk}	Method	f_{max}	f_{op}	f_{av}	std	AE	G_{av}	Time (s)
14	30.8785	ICOA	30.8785	30.8785	30.8785	0	0.0000%	2.2666	0.8437
		COA	30.8785	30.8785	30.8785	0	0.0000%	6.5666	0.7166
30	423.7406	ICOA	423.9117	423.7406	423.7463	0.0312	0.0013%	17.9333	2.3047
		COA	424.4643	423.7406	423.7704	0.1347	0.0070%	11.4	1.4771
48	33523	ICOA	33600.562	33523.709	33536.993	25.8082	0.0417%	142.6	142.6
		COA	33935.808	33523.709	33635.435	114.8451	0.3354%	180.8667	85.7792
52	7542	ICOA	7544.3659	7544.3659	7544.3659	0	0.0314%	103.0333	151.7625
		COA	7607.5616	7544.3659	7548.9918	15.2433	0.0927%	223.7333	99.676
76	108159	ICOA	110095.79	108159.44	108815.06	504.6571	0.6066%	220.0333	380.9646
		COA	122693.59	108793.41	115134.94	4186.6143	6.4497%	258.0667	259.7833
100	21282	ICOA	21414.805	21285.443	21340.049	43.0422	0.2728%	200.1	713.8401
		COA	39143.434	22603.445	31852.091	3483.872	49.6668%	279.7333	467.5365

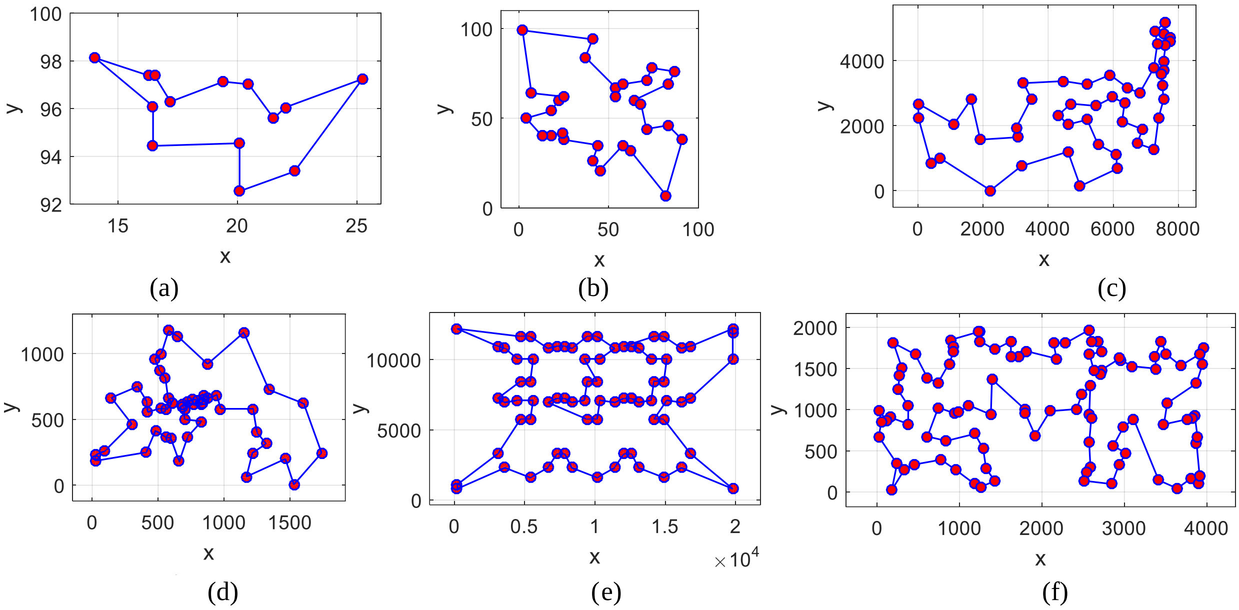

Figure. 5 The optimal tour for the instances: (a) 14-city, (b) 30-city, (c) 48-city, (d) 52-city, (e) 76-city, and (f) 100-city

The optimal tour of the 14-, 30-, 48-, 52-, 76- and 100-citiy instances gained by ICOA is shown in Fig. 5. The boxplot of ICOA and COA for all instances is presented in Fig. 6. The figure shows the stability and reliability of ICOA compared to COA. The corresponding average convergence characteristics in Fig. 7 shows that the improvement in the search mechanism has helped ICOA to converge to lower values with smaller numbers of generations than COA. The better f_{max}, f_{op}, f_{av}, std, G_{av} values of ICOA show the superior performance of ICOA over COA for the TSP problem.

The compared result between ICOA and other previous methods such as PSO [10], discrete spider monkey optimization (DSMO) [32], DTSA [15], ACO-GA [22], discrete artificial bee colony (DABC) [33], GA-PSO-ACO [20], SOS-SA [12], HHO-ACS [21], discrete bird swarm algorithm (DBSA) [34] is demonstrated in Table 2. For the 14-city instance, the performance of ICOA is similar to DSMO [32] while the f_{av} value gained by ICOA is better than that PSO and the ICOA's AE value is 0.1149% lower than that of PSO. For the 30-city instance, the performance of ICOA is similar to DABC [33]. The best tour achieved by ICOA is similar to that of PSO [10], DTSA [15] and ACO-GA [22] but the f_{av} value gained by ICOA is better than that of PSO [10], DTSA [15]. The ICOA's AE value is 2.0005% and 1.1218% lower than that of PSO [10] and DTSA [15]. For the 48-city instance, ICOA reaches the better results than GA-PSO-ACO [20] and the best tour length of ICOA is only 0.709 and 1.709 higher than that of SOS-SA [12] and HHO-ACS [21]. However, the ICOA's AE value is respectively 0.37290%, 0.00802% and 0.20585% lower than that of GA-PSO-ACO [20], SOS-SA [12] and HHO-ACS [21]. For the 52-city instance, the quality of the obtained solution gained by ICOA is only worse than DTSA

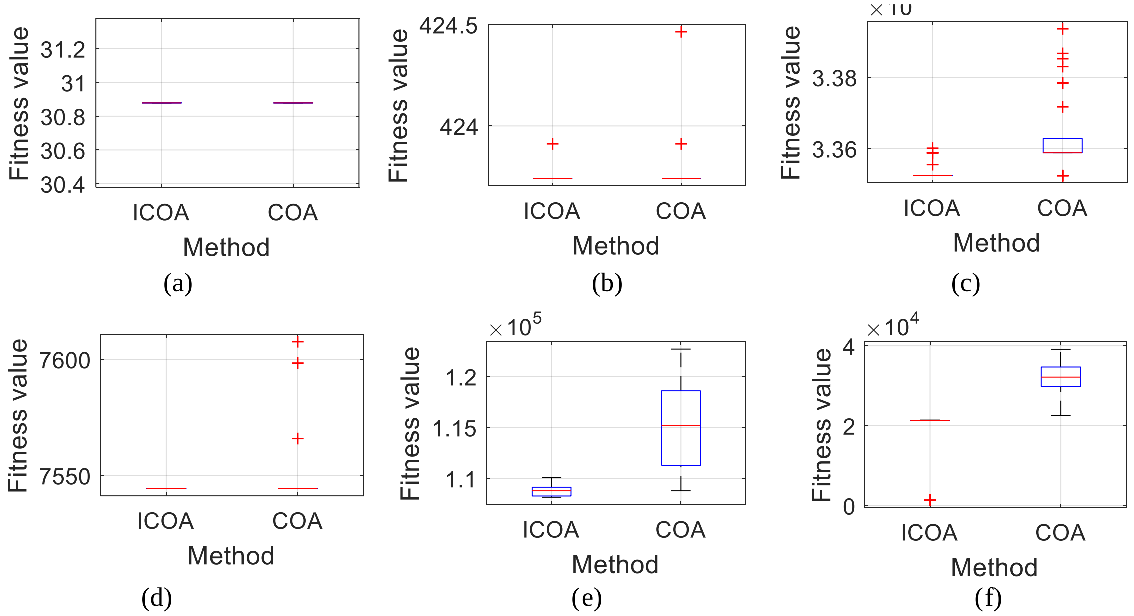

Figure. 6 Comparison of boxplot of ICOA and COA for the instances: (a) 14-city, (b) 30-city, (c) 48-city, (d) 52-city, (e) 76-city and (f) 100-city

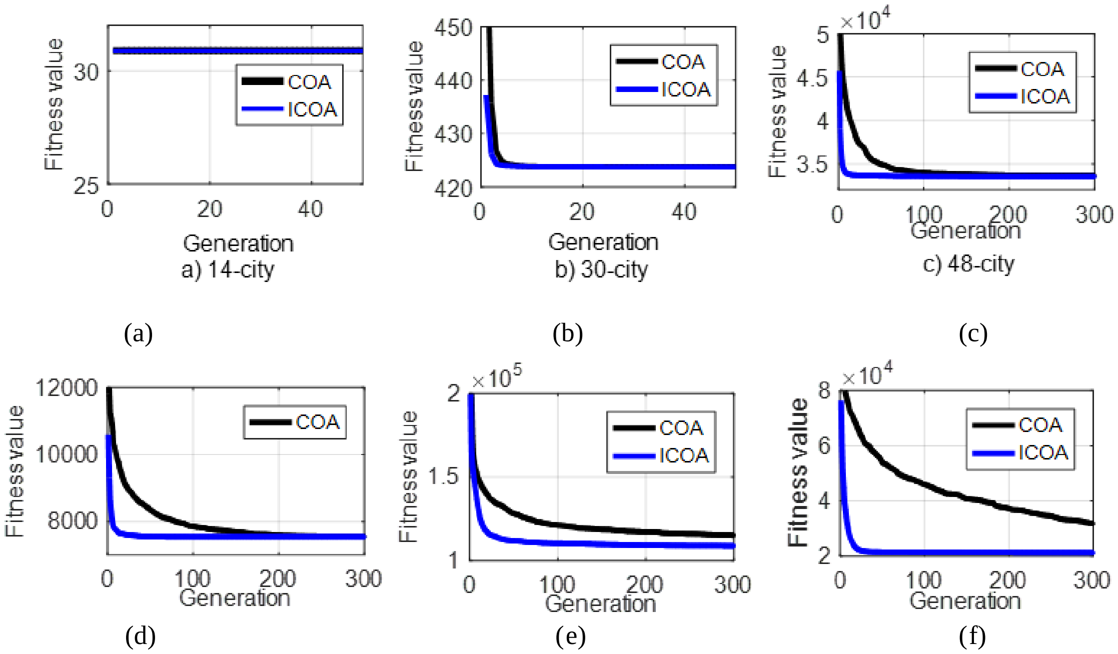

Figure. 7 Average convergence curve of ICOA and COA for the instances: (a) 14-city, (b) 30-city, (c) 48-city, (d) 52-city, (e) 76-city and (f) 100-city

[15] and SOS-SA [12] and identical to GA-PSO-ACO [20]. The ICOA's AE value is better than that of DTSA and HHO-ACS methods. Compared with DSMO [32], both of ICOA and DSMO [32] have determined the optimal result, but the mean value of ICOA is smaller than that of DSMO [32]. For 76-city instance, ICOA also outperforms to GA-PSO-ACO [20]. The ICOA's AE value is 1.1168% lower than that of the above method. In comparison with DSMO [32] and DBSA [34], all of three methods have found the optimal tour and the f_{av} value of ICOA is lower than that of DSMO [32]. For the 100-city instance, the performance of ICOA is better than that of DSMO [32]. The optimal solution gained by ICOA is 3.443 only higher than that of SOS-SA [12] but it is 567.557 lower than that of DTSA [15]. In addition, the ICOA's AE value is better than that of DTSA [15] and SOS-SA [12]. Based on the compared results with the aforementioned methods, ICOA has demonstrated that it is also an effective method for the TSP problem.

Table 1. The comparisons of ICOA with previous methods

City no.	Method	f_{op}	f_{av}	AE (%)
14	ICOA	30.8785	30.8785	0.0000
	PSO [10]	30.8785	30.9245	0.1490
	DSMO [32]	30.8785	30.8785	-
30	ICOA	423.7406	423.7463	0.0013
	PSO [10]	423.7406	432.2231	2.0018
	DTSA [15]	423.74	428.5	1.1232
	ACO-GA [22]	423.74	-	-
	DABC [33]	423.74	423.74	-
48	ICOA	33523.709	33536.993	0.04174
	GA-PSO-ACO [20]	33524	33662	0.4146
	SOS-SA [12]	33523	33539.68	0.0498
	HHO-ACS [21]	33522	33606	0.2476
52	ICOA	7544.3659	7544.3659	0.0314
	DTSA [15]	7542	7761.6	2.9117
	GA-PSO-ACO [20]	7544.37	7544.37	0.0314
	SOS-SA [12]	7540	7541.107	-0.0118
	HHO-ACS [21]	7542	7657	1.5248
	DSMO [32]	7544.37	7633.6	-
76	ICOA	108159.44	108815.06	0.6066
	GA-PSO-ACO [20]	109206	110023	1.7234
	DSMO [32]	108159.4	111299.3	-
	DBSA [34]	108159.44	108293.7	-
100	ICOA	21285.443	21340.049	0.2728
	DTSA [15]	21853	23213	9.0734
	SOS-SA [12]	21282	21424	0.6672
	DSMO [32]	21298.21	21878.83	-

6. Conclusion

In this paper, ICOA has been successful proposed for the TSP problem. In which, in order to enhance the candidate solution quality, the 2-opt algorithm is applied to adjust the created new solutions. In addition, the swapping search technique with varying exchange city number is also equipped for ICOA to improve the exploration and exploitation of the search space. The effectiveness of ICOA is compared with COA on the instances consisting of 14-, 30-, 48-, 52-, 76- and 100-city. The statistical results show that ICOA outperforms to COA for the TSP problem in terms of the optimal and average results as well as the number of convergence generations. Excepting for the 14-city case wherein the error between the average result and the best-known result in 30 runs of ICOA is equal to that of COA, all other larger-scale cases, this index of ICOA is always lower than that of COA. For the 30-, 48-, 52-, 76- and 100-city, this index of ICOA is 0.0057%, 0.2937%, 0.0613%, 5.8431% and 49.3940% lower than that of COA. Furthermore, the obtained results of ICOA are also better some of previous methods. Thus, ICOA is a potential tool for the TSP problem. For future work, ICOA can be validated for the larger-scale TSPs or practical applications.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, methodology, software, validation, writing-original draft preparation, writing-review and editing, QTN.

Acknowledgments

This study was financially supported by Van Lang University, Vietnam.

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, p. 113229, 2020.

[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, p. 106244, 2020.

[3] R. D. 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.

[4] C. H. Papadimitriou, "The Euclidean travelling salesman problem is NP-complete", *Theoretical Computer Science*, Vol. 4, No. 3, pp. 237–244, 1977.

[5] I. M. Oliver, D. J. Smith, and J. R. Holland,

"Study of permutation crossover operators on the traveling salesman problem", In: *Proc. of the second International Conference on Genetic Algorithms*, Massachusetts, USA, pp. 224–230, 1987

[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] 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.

[8] 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.

[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, p. 112867, 2020.

[10] 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.

[11] 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.

[12] 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.

[13] S. Bouzbita, A. E. E. 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.

[14] A. Ouaarab, B. Ahiod, and X. S. Yang, "Discrete cuckoo search algorithm for the travelling salesman problem", *Neural Computing and Applications*, Vol. 24, Nos. 7–8, pp. 1659–1669, 2014.

[15] 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.

[16] Q. T. Nguyen and M. P. Bui, "Performance of Equilibrium Optimizer for the Traveling Salesman Problem", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 5, pp. 46–55, 2021, doi: 10.22266/ijies2021.1031.05.

[17] Q. T. Nguyen and M. P. Bui, "Finding the best tour for travelling salesman problem using artificial ecosystem optimization", *International Journal of Electrical and Computer Engineering*, Vol. 11, No. 6, pp. 5497–5504, 2021.

[18] L. Strak, R. Skinderowicz, U. Boryczka, and A. Nowakowski, "A self-adaptive discrete PSO algorithm with heterogeneous parameter values for dynamic TSP", *Entropy*, Vol. 21, No. 8, pp. 1–21, 2019.

[19] O. Ilmandira, R. Farisi, B. Setiyono, and R. I. Danandjojo, "A hybrid approach to multi-depot multiple traveling salesman problem based on firefly algorithm and ant colony optimization", *IAES International Journal of Artificial Intelligence*, Vol. 10, No. 4, pp. 910–918, 2021.

[20] 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.

[21] S. A. Yasear and K. R. K. Mahamud, "Fine-Tuning the Ant Colony System Algorithm Through Harris's Hawk Optimizer for Travelling Salesman Problem", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 136–145, 2021, doi: 10.22266/ijies2021.0831.13.

[22] E. K. Elsayed, A. H. Omar, and K. E. Elsayed, "Smart solution for STSP semantic traveling salesman problem via hybrid ant colony system with genetic algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 476–489, 2020, doi: 10.22266/ijies2020.1031.42.

[23] 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.

[24] J. Pierezan and L. D. S. Coelho, "Coyote Optimization Algorithm: A New Metaheuristic for Global Optimization Problems", In *Proc. of 2018 IEEE Congress on Evolutionary Computation (CEC),* Rio de Janeiro, Brazil, pp. 1–8, 2018.

[25] T. T. Nguyen, Q. T. Nguyen, and T. T. Nguyen, "Optimal radial topology of electric unbalanced and balanced distribution system using

improved coyote optimization algorithm for power loss reduction", *Neural Computing and Applications,* Vol. 33, No. 18, pp. 12209-12236, 2021.

[26] J. Arfaoui, H. Rezk, M. A. Dhaifallah, M. N. Ibrahim, and M. Abdelkader, "Simulation-based coyote optimization algorithm to determine gains of PI controller for enhancing the performance of solar PV water-pumping system", *Energies*, Vol. 13, No. 17, p. 4473, 2020.

[27] M. H. Qais, H. M. Hasanien, S. Alghuwainem, and A. S. Nouh, "Coyote optimization algorithm for parameters extraction of three-diode photovoltaic models of photovoltaic modules", *Energy*, Vol. 187, p. 116001, 2019.

[28] G. A. Croes, "A Method for Solving Traveling-Salesman Problems", *Operations Research*, Vol. 6, No. 6, pp. 791–812, 1958.

[29] X. Zhou, D. Y. Gao, C. Yang, and W. Gui, "Discrete state transition algorithm for unconstrained integer optimization problems", *Neurocomputing*, Vol. 173, pp. 864–874, 2016.

[30] G. Reinelt, "TSPLIB. A traveling salesman problem library", *ORSA journal on computing*, Vol. 3, No. 4, pp. 376–384, 1991.

[31] I. M. Ali, D. Essam, and K. Kasmarik, "A novel design of differential evolution for solving discrete traveling salesman problems", *Swarm and Evolutionary Computation*, Vol. 52, p. 100607, 2020.

[32] 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 Journal*, Vol. 86, p. 105887, 2020.

[33] 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.

[34] M. Lin, Y. Zhong, J. Lin, and X. Lin, "Discrete bird swarm algorithm based on information entropy matrix for traveling salesman problem", *Mathematical Problems in Engineering*, Vol. 2018, p. 9461861, 2018.