TSP Solving Utilizing Improved Ant Colony Algorithm

Xuan-Shi Yao¹, Yun Ou¹ and Kai-Qing Zhou¹
¹College of Information Science and Engineering, Jishou University, Jishou, 41600, Hunan, China.

Email: yun_ou@163.com

Abstract. To solve the premature issue of TSP solving using the ant colony optimization algorithm (ACO), this paper proposes an improved ACO using particle swarm optimization (PSO) to solve the classic traveling salesman problem (TSP). The algorithm's strategy includes three stages: firstly, establishing a mathematical model according to the optimization objective, and then solving the optimal path obtained by the particle swarm optimization algorithm. Finally, the pheromone concentration of this path in the ant colony mathematical model is enhanced according to the particle swarm optimization algorithm's optimal path. A classic TSP case is used to compare the PSO and ACO. The results show that the proposed improved algorithm has a faster convergence speed and can converge to the optimal global solution, and its performance is better than that of ACO and PSO.

1. Introduction
Recently, it becomes a research hot-point of how to obtain faster and more accurate results of traveling salesman problem (TSP) by using soft computing techniques. Kamei and Ishikawa [1] combine the ACO with reinforcement learning to improve the implementation performance of each ant's exploration path capability. Zuliani et al. [2] integrate the core operators between genetic algorithm (GA) and ACO to achieve better results in the experiment of multi-objective optimization. Skinderowicz [3] uses a restart process to accelerate the convergence of the standard ACO and explore the influences between the parameters and the algorithm's convergence. Haroun et al. [4] propose a hybrid GA-ACO to solve TSP. In the hybrid GA-ACO algorithm, ACO guides the particle movement through pheromone, and the across mutation is used to enhance the population's diversity to avoid fall into the local optimum. Zhong et al. [5] use an improved PSO to solve the TSP problem and have achieved good results.

Inspired by the above literature, this research will use the improved ACO (IACO), which combines the ACO and PSO algorithms to solve the TSP issue. First, the PSO is used to solve the TSP problem for obtaining the initial solution. Then the gained initial solution is converted into the increment of pheromone in the ACO to implement further iterative optimization for obtaining the final optimal solution. According to these improvement strategies, it can effectively solve the shortcomings of the ACO's slow convergence speed in the early stage and the local extreme value of the PSO in the later stage.

2. Basic Problem Description
2.1. Traveling Salesman Problem
The TSP is a typical NP-hard problem [6], which could be described below.
Suppose a merchant needs to sell goods through n cities, and the merchant needs such a route:
1. Each city arrives only one time.
2. The starting position and ending position of the route must be the same city.
3. The solution of TSP is to seek the shortest line which meets above requirements.

The corresponding mathematical model of TSP issue could be expressed as follows.

Objective function: 
\[
\min Z = \sum_{i,j=1}^{n} x_{ij}d_{ij} 
\]
\[
x_{ij} = \begin{cases} 
1. & \text{If the salesmans visits the cities } i \text{ and } j \text{ sequentially;} \\
0. & \text{otherwise.} 
\end{cases} 
\]

Among them, \(d_{ij}\) is the distance from city \(i\) to city \(j\).

2.2. Application of PSO in TSP

The PSO [7] comes from the study of bird predation behavior, a simplified mathematical model inspired by the law of bird group activities. The information sharing mechanism of each particle in PSO makes the entire particle swarm get the optimal solution in the solution space. When using PSO to solve the TSP, each solution is the path length of each solution, and the TSP problem's solution space are expressed as a particle, the particle's fitness value, and a population composed of \(N\) paths respectively. The solution's fitness value performs the path selection, and each particle has a memory function that can remember the optimal solution found.

2.3. Application of ACO in TSP

The basis of ACO to solve the TSP is to express the path taken by each ant in the ant colony as a solution of the problem, and all paths of the entire ant colony are a solution space of the problem [5]. Ants release more pheromone on a short path, and they evaporate slowly. Therefore, the concentration is high. As time progresses, more ants choose the path, and the pheromone concentration of the path gradually increases. Finally, the entire ant colony will be concentrated to the optimal path under the action of the positive feedback mechanism, and this path is the optimal solution to the problem.

3. Application of IACO in TSP Problem

The presented IACO in this paper combines the advantages of the PSO with strong global search ability and good feedback ability of the ACO to speed up the convergence speed and to avoid falling into local extremum in the later stage. In solving the TSP, the obtained initial solution of TSP by using PSO is transformed into the enhancement of the pheromone in the ACO at first, and then the ACO is executed to further solve the optimal solution of the TSP problem. Additionally, the pheromone concentration is appointed as 1.3. Finally, select the optimal solution from the optimal path under each pheromone concentration. The steps of the IACO algorithm are as follows:

Step 1: Initialize the particle speed, position, and population size \(N\);

Step 2: Calculate the fitness value of each particle according to equation (2);
\[
d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} 
\]
\[
\text{Fitness} = \sum_{(i,j) \in \text{particle}} d_{ij} 
\]

Step 3: Each particle remembers own optimal fitness value \(P_{\text{best}}\) and the current global optimal fitness value \(G_{\text{best}}\), respectively. The particle continuously adjusts its position for new solutions through equation (3) below.
\[
v_{i+1} = wv_i + c_1r_1(P_{\text{best}} - x_i) + c_2r_2(G_{\text{best}} - x_i) 
\]
\[
x_{i+1} = x_i + v_{i+1} 
\]
Step 4: Determine whether the loop condition is satisfied, if yes, output the optimal path obtained by the particle swarm, otherwise, return to step 2;
Step 5: Convert the optimal path obtained by the particle swarm into the pheromone concentration of the ant colony algorithm, and appointed the pheromone concentration as 1.3;
Step 6: The ant colony is randomly distributed to each city;
Step 7: Each ant selects the next position according to the state transition probability equation (4);
\[ p_{ij}^t(t) = \frac{\left[ \tau_{ij}(t) \right]^\alpha \left[ \eta_{ij}(t) \right]^\beta}{\sum_{k \in \text{neibors}} \left[ \tau_{ik}(t) \right]^\alpha \left[ \eta_{ik}(t) \right]^\beta} \] (4)
Step 8: Calculate the path length obtained by each ant, record the global optimal path of this iteration, and update the taboo table;
Step 9: Update the pheromone on each path using equation (5);
\[ \tau_{ij}(t + n) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij} \]
\[ \Delta \tau_{ij} = \sum_{k-l} \Delta \tau_{ij}^k \] (5)
Step 10: Determine whether the iteration condition is satisfied, if not, return to step 7, if yes, output the optimal path, and the algorithm ends.

According to the algorithm steps, the following algorithm flow chart is illustrated as shown in Figure 1.

Figure 1. Entire flowchart of the ICAO

4. Experimental Results and Related Analysis
The experiment related parameters are assigned.
The number of particles and ants: \( N = 30 \);  
The number of cities: \( C = 14 \);  
The maximum number of iterations: \( NC_{\text{MAX}} = 100 \);  
City coordinates: \( C = [16.47, 96.10; 16.47, 94.44; 20.09, 92.54; 22.39, 93.37; 25.23, 97.24; 
22.00, 96.05; 20.47, 97.02; 17.20, 92.54; 16.30, 97.38; 14.05, 98.12; 16.53, 97.38; 21.52, 95.59; 
19.41, 97.13; 20.09, 92.55] \);

The experimental simulation results are obtained as shown in Figure 2.

Figure 2. Experimental results by implementing three algorithms

Figure 2 shows that the shortest path solved by ACO is 29.6889 meters, the shortest path solved by PSO is 31.8825 meters, and the shortest path solved by IACO is 29.3418 meters. It indicates that Figure 2 and the shortest path that IACO is better than a single ACO or PSO, respectively.

5. Summary
To solve the shortcomings of ACO in the early stage of convergence and the drawback of PSO, which is easy to fall into local extremes in the later stage, this paper presents an improved ant colony algorithm (IACO) by combining the highlights of ACO and PSO to address TSP. The experimental results show that the optimized solution obtained by the discussed IACO is better than a single ant colony algorithm or a single particle swarm algorithm.

References
[1] Kamei K, Ishikawa M 2010 International Conference on Neural Information Processing, pp. 609-617
[2] Zuliani J B Q, Weiss Cohen M, Batista L D S, Gadelha Guimarães F 2015 Comput. Aided Des. Appl. 12(6) pp 674-682
[3] Skinderowicz R 2016 International Conference on Computational Collective Intelligence pp 91-101
[4] Haroun S A, Jamal B, Hicham E H 2015 Int. J. Comput. Appl. 117 pp 28-35
[5] Zhong Y W, Yang J G, Ning Z Y 2006 Systems Engineering Theory & Practice 14(6) pp 88-94.
[6] Yabo L, Shikun Z, Feng Z 2013 Joint International Conference on Pervasive Computing and the Networked World pp 432-440
[7] Kennedy J, Eberhart R C 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation pp 4104-4108

Acknowledgement
This work was supported by the National Natural Science Foundation of China (No. 62066016), the Natural Science Foundation of Hunan Province, China (No.2020JJ5458), the Research Foundation of Education Bureau of Hunan Province, China (No.18B317).