An Improved Ant Colony Optimization Algorithm

Xin PAN, Xusheng WU, Xinguo HOU, Yuan FENG

School of Electrical Engineering, Naval University of Engineering, Wuhan, Hubei, 430033, China

Abstract — **To overcome the faults of slow convergence rate and ease of falling into local optimal solutions in classical Ant Colony Optimization algorithms, an improved algorithm is proposed from three aspects: i) adjustment of the state transition rule, ii) alteration of the pheromone updating rule, and iii) integration of local optimization algorithms. The simulation results show that the proposed algorithm has much higher capacity of searching global optimal solution and faster convergence rate than the classical algorithms.**

Keywords: Ant Colony Optimization, the pheromone updating rule, global optimal solution

I. INTRODUCTION

The Ant Colony Optimization (ACO) algorithms, since created in the 90s of the last century, has been successfully used to solve searching the best tour, communication network routing problem and other optimization problems. The primitive form of the ACO algorithms is Ant System (AS), then Ant Colony System (ACS), Max-Min Ant System (MMAS) were created, but the defects mainly refers to slow convergence rate and easily fallen into local optimal solution have not been solved well yet. On the basis of basic ACO algorithms, the combination of them and other intelligence algorithms like Genetic Algorithm, Bees Colony Algorithm were discussed [1, 2], DT strategy were introduced and good simulation results have been obtained in solving CVRP problem [3], Augment Ant Colony Algorithm which contains crossover and mutation were proposed to update the pheromone trail changing mechanism [4,5], 2-opt optimization algorithm was also joined to help adjust route [6]. All the above studies have achieved some results, but the accuracy and convergence rate of the solution are still needed to further improvement.

In this paper, a new improved Ant Colony algorithm was proposed, the algorithm modifies the state transition rule, the pheromone updating rule and two optimization algorithms are integrated in the late stage of iterations. In the simulation, a typical kind of Traveler Salesman Problem (TSP) as an example was solved, the computational results show that the improved ACO algorithm performs a faster convergence rate and have a greater chance of searching global optimal solution.

II. THE IMPROVED ANT COLONY OPTIMIZATION ALGORITHM

The improved ACO algorithm differs from the previous ACOs because of the following three main aspects:

(1) Adjustment of the state transition rule. According to characteristics of the AS's and ACS's state transition rules, a dynamic invocation method is used in the new algorithm. The method can further accelerate the convergence speed of the algorithm.

(2) Alteration of the pheromone updating rule. In each iteration, the genetic crossover and mutation operator are introduced to influence the pheromone updating rule, which would expand the searching scope and enhance the convergence rate.

(3) Integration of local optimization algorithm. At the late stage of iterations, two local optimization algorithms are integrated to accelerate the population evolution rate and increase the probability of searching the global optimal solution.

A. Adjustment of the State Transition Rule

In AS, the probability with which an ant at node i chooses to move to node j is given by (1):

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{r\in allow_k}\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}, & j\in allow_k\\ 0, & j\notin allow_k\end{cases} \quad (1)$$

where $\tau_{ij}(t)$ is the pheromone between node i and node j at time t, $\eta_{ij}(t)$ is the inverse of the distance between node i and node j, $allow_k$ is the set of nodes which remains to be visited, α represents the importance of pheromone in edge selection, β represents the importance of heuristic information, this selection strategy reflects a certain randomness, known as a random-proportional rule. But the randomness makes the algorithm perform a low convergence rate, so in ACS a more bold transition rule is adopted known as a pseudo-random proportional rule. The algorithm introduces a parameter q_0 and compares it with a random between 0 and 1 when an ant moves between nodes, the state transition rule is given by (2):

$$s=\begin{cases}\arg\ max\left\{\left[\tau(i,j)\right]^{\alpha}\left[\eta(i,j)\right]^{\beta}\right\} & q\le q_0\\ S & q>q_0\end{cases} \tag{2}$$

where S is the random-proportional rule in (1), obviously the certainty of the path directed is increased which leads to a higher convergence rate. But, it wouldn't be good for algorithm's every stage. At the beginning of the algorithm, due to the evenly distribution of each pheromone amount in each edge, the algorithm given by (2) may induce ants to move to the nodes which have shortest distance from their current nodes, resulting in pheromone amount between certain nodes accumulation, further increasing the probability for ants to choose certain edges, which may bring ants misguided to select a not suitable route, finally cause the deterioration of convergence. On the other hand, the algorithm may converge to a local optimal solution in the late stage when the search for ants should be expanded, but the considerable degree of certainty would reduce the probability of other paths and it doesn't good to converge to the global optimal solution.

Based on the above considerations, a new state transition rule is put forward in the paper given by (3):

$$P=\begin{cases}\text{the state transition rule in } AS(S) & iter\le N_{min}\\ \text{the state transition rule in } ACS(s) & N_{min}\le iter\le N_{max}\\ \text{the state transition rule in } AS(S) & N_{max}\le iter\end{cases} \tag{3}$$

where the random-proportional rule is adopted in the initial stage, in order to search more paths, then the algorithm change to pseudo random-proportional rule to fast convergence and finally the random-proportional rule is reused to avoid falling into local optimal solution and search for global optimal solution.

B. Alteration Of The Pheromone Updating Rule

In ACS, the updating of pheromone includes global updating and local rules. The global updating rule indicates that once all ants have built their tours, the ant which constructed the global optimal tour is allowed to leave pheromone. The pheromone updated between two nodes (take i and j for an example) is as follows:

$$\tau_{ij}\left(t+1\right)=\left(1-\rho\right)\cdot\tau_{ij}\left(t\right)+\rho\cdot\Delta\tau_{ij}\left(t\right) \tag{4}$$

where

$$\Delta\tau_{ij}\left(t\right)=\begin{cases}1/L_{gb} & if\left(i,j\right)\in the\ global\ optimal\ solution\\ 0 & otherwise\end{cases} \tag{5}$$

Local updating rule means while all ants are building their tours, every ant update pheromone in tours meanwhile by (6):

$$\tau_{ij}\left(t+1\right)=\left(1-\xi\right)\cdot\tau_{ij}\left(t\right)+\xi\cdot\tau_0 \tag{6}$$

where we usually set $\tau_0=1/\left(n\cdot L_{nn}\right)$ [8], L_{nn} represents the distance between the current nodes and the nearest optional nodes. The local updating rule leads to the pheromone amount would be reduced after an ant have gone through the corresponding path. That means the rule would make the desirability of paths change dynamically so that more tours would be explored.

In the improved algorithm, the local updating rule is unchanged while the global updating includes two parts. In first part, the global sub optimal solution is added to the global pheromone updating mechanism in order to expand ants searching scope. The ant which constructed the optimal tour and the ant which constructed the sub optimal tour are both allowed to deposit pheromone with different weight coefficients. Detailed instructions are as follows, by comparing global optimal solution and global sub optimal solution, paths are divided into three kinds: the common edges named L_A, the edges belong to the optimal solution but not belong to the sub optimal solution named L_B, the edges belong to the sub optimal solution but not belong to the optimal solution named L_C, then weight coefficients of A, B and C are added separately to update the pheromone in global updating. The length of the global sub optimal tour is noted as L_{gsb}, then the first part of pheromone increment is given by (7).

$$\Delta\tau_{ij1}\left(t\right)=\begin{cases}A/L_{gb} & if\left(i,j\right)\in L_A\\ B/L_{gb} & if\left(i,j\right)\in L_B\\ C/L_{gsb} & if\left(i,j\right)\in L_C\end{cases} \tag{7}$$

In second part, crossover and mutation operators in Genetic Algorithm are introduced. In each iteration, the tours of the global optimal solution and the global sub optimal solution are taken as the initial individuals, then two sub individuals were created by two-point crossover operation, and basic mutation operation is carried out on each sub individuals, so that four sub individuals are created finally. Evaluate the four sub individuals and select the optimal one with its length named L_{ngb}. The weight coefficient in pheromone increment of this part is based on the compare of L_{ngb} and L_{gb} as given by (8).

$$\Delta\tau_{ij2}(t) = \begin{cases} D / L_{ngb} & if\ L_{ngb} < L_{gb} \\ E / L_{ngb} & if\ L_{ngb} \geq L_{gb} \end{cases} \quad (8)$$

In each iteration, the pheromone increment between two nodes (take nodes i,j for an example) at time t are $\Delta\tau_{ij1}(t)$ plus $\Delta\tau_{ij2}(t)$, and the global updating rule is adapted to (9).

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\left(\Delta\tau_{ij1}(t) + \Delta\tau_{ij2}(t)\right) \quad (9)$$

C. Integration of Local Optimization Algorithm

In [6] 2-opt optimization was added after the algorithm have fallen into local optimal solution in the late stage, which helps the ants converge to the global optimal solution quickly. But when the searching scope reach a certain scale, only rely on the 2-opt optimization algorithm may not achieve the desired results. Therefore, another local optimization algorithm (random-searching algorithm) is added with 2-opt optimization to the improved algorithm.

Take the tour 1-2-3-4-5-6 as an example. The 2-opt optimization algorithm generate new tours at first. It refers to two nodes are taken out randomly, such as node 2 and node 5, then the location of the two nodes are exchanged and the tour between two nodes are reversed, and the optimized tour is 1-5-4-3-2-6. When there are n nodes in a tour, the optimization algorithm would generate $n(n-1)/2$ tours, then update the optimal solution through evaluation of generated tours. In order to generate more tours for ants to choose, random-searching algorithm is added. The algorithm is defined as: one node is taken out randomly, such as node 3, then the node is inserted into any other locations of the original tour and other nodes keep unchanged, forming a new tour 3-1-2-4-5-6. The random-searching algorithm would also generate $n(n-1)/2$ tours. Although the tours generated by two local optimization algorithm may overlap in some cases, such as a tour 2-1-3-4-5-6, but the tours for ants to choose are still largely expanded. Therefore the combination of two local optimization algorithms would have a more probability and faster speed to convergence to the global optimal solution than [6].

In initial stage, the improved algorithm have a fast convergence rate already, in order to simplify the algorithm, choose to add two local optimization in the late stage when it would have fallen into local optimal solution. Set the integration moment is when the number of iteration $iter$ meets the timing given by (10):

$$iter \begin{cases} \geq N_e \\ Length\_best(iter) = \cdots = Length\_best(iter-10) \end{cases} \quad (10)$$

where $Length\_best$ is local optimal solution.

III. THE ALGORITHM IMPLEMENTATION

The implementation of the improved ACS algorithm are as follows:

Step 1. Initialize parameters, mainly including: the number of ants m, the number of nodes n, regulation factors α and β, pheromone volatile factor ρ and ξ, pheromone initial value matrix τ, the number of iterations $iter$, maximum iterations $iter\_max$, the state transition time N_{min} and N_{max}, the integration of optimization algorithm time N_e, weight coefficients A, B, C, D and E, tours record matrix $Table$, and calculate distance matrix D.

Step 2. Set the number of iterations $iter = iter + 1$.

Step 3. Randomly place ants in nodes, and mark in the record matrix $Table$.

Step 4. For each ant in construction tour, firstly adjust the set of visited nodes (the tabu list) $tabu$ and the set of nodes to be visited $allow$, and secondly ants are selected to next nodes based on (3), and thirdly pheromone are updated according to local updating rule (6).

Step 5. For each ant having constructed tour, calculate the lengths of tours.

Step 6. Calculate the shortest length and the average length of all tours, and update the global optimal solution L_{gb} and the global sub optimal solution L_{gsb}.

Step 7. Use crossover and mutation operators on the global optimal solution and the global sub optimal solution as introduced before.

Step 8. Calculate pheromone increment $\Delta\tau_{ij}(t)$, and update pheromone with the global updating rules.

Step 9. Judge whether the number of iterations $iter$ meet equation (9), if meet, integrate the local optimization algorithms and update the global optimal solution, then judge whether $iter$ reach $iter\_max$, if meet again, turn to Step 10, if not meet, turn to Step 2, if neither meet, turn to Step 2.

Step 10. Output results, end.

IV. SIMULATION RESULTS

TSP refers to: given a set of n cities and distances for each pair of nodes, a salesman begins his travel from one of the cities, and he tries to find a roundtrip of minimal total length visiting each cities exactly once [7]. In this paper, in order to evaluate the performance of the improved algorithm, TSP of 31 cities in China is taken as an example. The known optimal solution is 15378km [8].

In simulation, parameters were initialized as follows: $\alpha = 2$, $\beta = 5$, $\rho = \xi = 0.2$, $m = 50$, $iter\_max = 200$. After several tests it was found that when $[N_{min}\ N_{max}\ N_e]$ valued $[10, 60, 100]$ and $[A\,B\,C\,D\,E]$ valued $[1.2, 0.9, 0.6, 1.2, 0.8]$, the simulation achieved good results. Matlab is adopted for programming. The simulation consists of three verification tests, which are respectively aimed at adjustment of state transition rule, integration of local optimization algorithm and the whole improved ACO algorithm.

A. The state transition rule verification simulation

In order to evaluate the performance of the state transition rule in the improved algorithm, compare it with the rules in AS and ACS. To make the simulation results more convincing, combine the state transition rule in AS and other part in the improved algorithm as an algorithm named A1, and combine the state transition rule in ACS and other part in the improved algorithm as an algorithm named A2. The simulation was carried out three times. In the initial stage of simulation, the evolution curves of average length for A1, A2 and the improved algorithm are shown in Fig.1.

It can be clearly observed form Fig.1 that in the initial stage, the curve of A2 algorithm has a process of increasing then decreasing, although the convergence rate is fast but has a period of deterioration of performance. The curve of A1 algorithm shows that although the algorithm has convergence since the beginning, but the convergence rate has kept slowly. When iterations reach 60 times, the average length of two algorithms are basically the same which indicates their performance have been similar. The curve of the improved algorithm shows that the algorithm is in continuous convergence with a rate change from slow to fast, and the average length is significantly less than the other two algorithms. The above analysis shows that the improved algorithm has a better convergence effect.

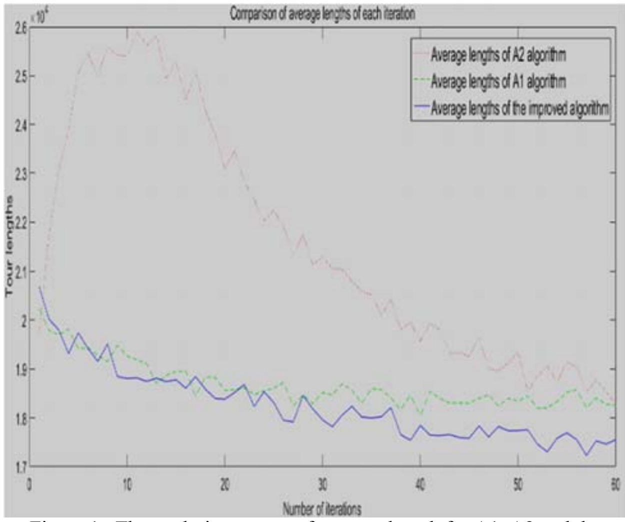

Figure 1. The evolution curves of average length for A1, A2 and the improved algorithm.

B. Local optimization algorithm verification simulation

In order to evaluate the performance of integration of local optimization algorithm, compare the simulation results of four kinds algorithms: the improved algorithm without any one, the improved algorithm with only 2-opt optimization, the improved algorithm with only random-searching optimization and the improved algorithm with both. The simulation for each algorithm was carried out with 20 times and specific data results are shown in Table Ⅰ.

.

TABLE I.

Parameters	Neither	Only 2-opt	Only random-searching	both
The iteration when first appeared the optimal solution	161	115	108	105
The average number of iterations for the optimal solution appeared	182	143	151	120
The total numbers of the optimal solutions appeared	13	17	16	20

It can be known that the integration of two local optimization algorithms would not only make the algorithm converge with a faster rate, but also increase the probability of convergence to the optimal solution. A variety of local optimal solutions appeared in the simulation as Fig.2 and Fig.3 shown below.

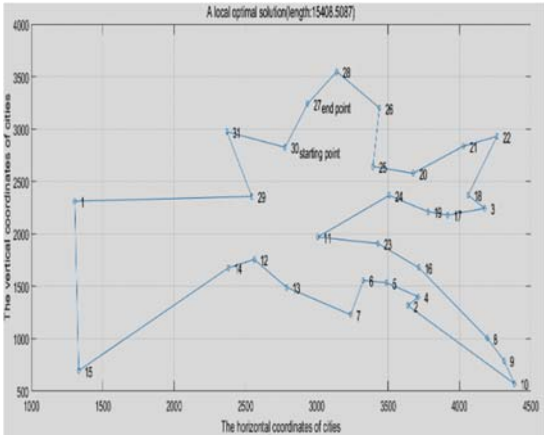

Figure 2. A local optimal solution with 15408km.

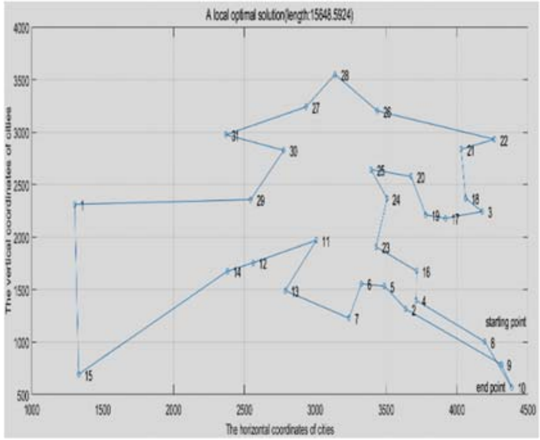

Figure 3. A local optimal solution with 15648km.

In figures, the numbers represent corresponding cities. In the right corner of figures, the global optimal solution 15377km corresponds to tours “5-2-10-9-8-4-16-23”. The tour “5-4-2-10-9-8-16-23” in Fig.2 would be optimized by integration of random-searching optimization, and the tour “5-2-9-10-8-8-4-16-23” in Fig.3 would be optimized by integration of 2-opt optimization, which is further proved that the integration of both local optimization is very necessary for fast convergence to the global optimal solution.

C. *The improved ACO algorithm verification simulation*

In order to evaluate the performance of the whole algorithm, compare the improved algorithm with ACS and the enhanced ACO in [5]. The simulation for each algorithm was carried out with 20 times and specific data results are shown in Table II.

TABLE Ⅱ

Parameters	AS	The enhanced ACO	The improved algorithm
The total numbers of the optimal solutions appeared	11	16	20
The average length of solution	15620km	15525km	15410km
The average number of iterations for the optimal solution appeared	176	158	118

As can be seen from Table II, the improved ACO algorithm has improved significantly in convergence rate and has much higher capacity of searching global optimal solution than others, when convergence to the global optimal solution, the tours are shown in Fig.4, and the evolution of the shortest length and the average length in some simulation are shown in Fig.5.

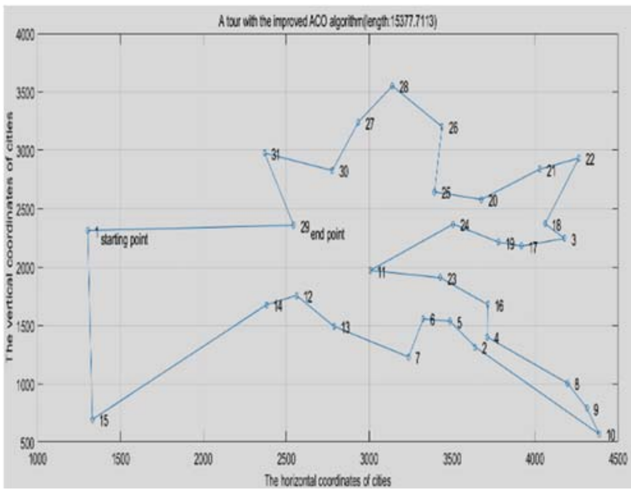

Figure 4. The global optimal solution with 15377km

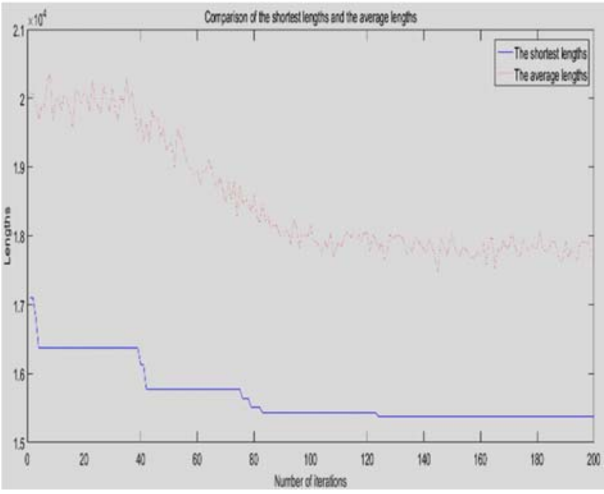

Figure 5. The evolution curves of the shortest and average lengths.

REFERENCES

[1] T. X. Jiang, "Application of improved Genetic Ant Colony Hybrid Algorithm in TSP," Computer and Modernization, vol. 12, pp. 30-33, 2013.

[2] Z. Y. HE and X. Wang, "Adaptive bee-ant colony optimization," Application Research of Computers, vol. 29, pp. 130-134, 2012.

[3] L. Chen and J. J. Zhou, "Improved Ant Colony Optimization Algorithm to solve CVRP," Journal of Military Transportation University, vol. 16, pp. 92-95, 2014.

[4] Z. Yan and C. W. Yuan, "An improved Ant Colony Optimization," Computer Engineering and Applications, vol. 23, pp. 62-64, 2003.

[5] H. J. Chen, J. Chen, X. H. Xu and L. Tu, "An improved augment Ant Colony Algorithm," Computer Engineering, vol. 31, pp. 176-178, 2005.

[6] H. Hua, X. L. Fu and D. Q. Wang, "Research on the MMAS algorithm solving TSP based on 2-opt," Journal of Inner Mongolia Agricultural University, vol. 35, pp. 142-146, 2014.

[7] C. B. Li, R. X. Guo and M. Li, "Application of improved Ant Colony Algorithm in travelling salesman problem," Journal of Computer Applications, vol. 34, pp. 131-132, 2014.

[8] L. Yu, F. Shi, H. Wang and F. Hu, 30 Cases Analysis of Intelligent Algorithm in Matlab. Beihang University Press, 2015.

[9] T. J. Liao and Krzysztof, "Ant Colony Optimization for mixed-variable optimization problems," Transactions on Evolutionary Computation, vol. 18, pp. 503-517, 2014.

[10] Delevacq, Delisle and Gravel, "Parallel Ant Colony Optimization on graphics processing units," Parallel Distribution Computation, vol. 73, pp. 52-61, 2013.

[11] J. Bai, G. K. Yang and Y. W. Chen, "A model induced Max-Min Ant Colony Optimization for asymmetric travelling salesman problem," Applied Soft Computing, vol. 13, pp. 1365-1375, 2013.

[12] Korytkowski and Rymaszewski, "Ant Colony Optimization for job shop scheduling using multi-attribute dispatching rules," Advanced Manufacturing Technology, vol. 67, pp. 231-241, 2013.

[13] Seckiner and Eroglu, "Ant Colony Optimization for continuous functions by using novel pheromone updating," Applied Mathematics and computation, vol. 219, pp. 4163-4175, 2013.