Solving TSP based on an Improved Ant Colony Optimization Algorithm

Hao Zhang, Yunfeng Gao*

College of Transport and Communications, Shanghai Maritime University, Shanghai, 201306, China.

*Corresponding author e-mail: yfgao@shmtu.edu.cn

Abstract. Traveling Salesman Problem (TSP) is a typical Problem in combinatorial optimization field in modern times. Most of the problems in reality can be transformed into TSP problems for solving. Such as postal problems, communication network design, etc. Ant colony algorithm, as a heuristic algorithm, has been successfully applied to solving TSP problems. Based on the improved ant colony algorithm, this paper solves the travel agent problem, evaluates the population according to the membership degree, and updates the pheromone in turn, so as to achieve a good balance in solving speed and quality.

Keywords: TSP, Ant colony algorithm, combinatorial optimization, solving speed.

1. Introduction

Combinatorial Optimization is a mathematical method to study the optimal arrangement, grouping, order and selection of discrete problems. The problems it studies involve many fields such as transportation and traffic networks. Since a large number of optimization problems in real life are actually selecting the best one from a limited number of states, the actual optimization problem can be described as a combinatorial optimization problem. Combinatorial optimization problems are often accompanied by the gradual expansion of the problem scale. The problem space shows the characteristics of combinatorial explosion, and the space and time complexity of solving the problem will show an exponential growth trend. The traveling salesman problem (TSP) is one of the most classic problems in combinatorial optimization problems.

2. Description and mathematical model of TSP

As one of the most typical problems in combinatorial optimization, the TSP problem is explained with the graph theory method: in a weighted completely undirected graph G, a Hamilton loop is found and the weight value of the loop is minimized. Take a graph G to represent a tuple \((V, E)\), where \(V\) represents the set of vertices and \(|V| = n\), \(E\) represents the set of edges. With non-negative weights \(d_{ij}\) on each \(e \in E(e = (i, j))\), look for the Hamilton loop of G graph and minimize the total weight \(W(C) = \sum_{i,j \in V} d_{ij}\) of the path of the loop. The TSP can be classified according to the distance matrix. When the condition \(c_{i,j} = q_{ij}(i,j \in V)\) is satisfied, it means that the distance between any city and back is
symmetric. The TSP is called symmetric TSP. When the condition \( c_{i,j} \neq c_{j,i} (i, j \in V) \) is satisfied, the problem becomes an asymmetric TSP.

The TSP problem can be defined as: in an undirected complete graph with given \( n \) vertices, a closed loop \( C \) composed of all vertices is required to be traversed in graph \( G \). And minimize the sum of the weights of all the edges on \( C \). Where \( V = \{1, 2, 3, \ldots, n\} \) represents the vertex set, \( E \) is the edge set connecting two different vertices in \( V \), and \( R \) is the weight set of edges. The mathematical model of TSP is described as follows:

\[
\begin{align*}
\min Z &= \sum_{i=1}^{n} \sum_{j=1}^{n} d_{i,j} \times_{i,j} \\
\text{s.t.} \sum_{i=1}^{n} x_{i,j} &= 1, (i = 1, 2, \ldots, n) \quad (2) \\
\sum_{j=1}^{n} x_{i,j} &= 1, (j = 1, 2, \ldots, n) \quad (3) \\
\sum_{i, j \in \delta} x_{i,j} &\leq Q - 1, 2 \leq |Q| \leq n - 2, Q \subset \{1, 2, \ldots, n\} \quad (4) \\
x_{i,j} &\in \{0, 1\}, (i, j = 1, 2, \ldots, n) \quad (5)
\end{align*}
\]

In the mathematical model of TSP, Equation (1) represents the objective function, represents the minimum length of the path traveled by the traveling salesman, where \( d_{i,j} \) represents the weight from city \( i \) to city \( j \); Equations (2) and (3) simultaneously represent that each city is passed by a travel agent only once; Equation (4) represents that the travel agent cannot form a cycle in the true subset of any city; Equation (5) represents the value of the decision variable; if \( x_{i,j} = 1 \), represents the cities that the traveling salesman has traversed; If \( x_{i,j} = 0 \), then represents a city that the travel agent has not experienced.

3. Ant Colony Optimization Algorithm

3.1. Introduction to ant colony optimization algorithms

Ant colony algorithms are a set of software agents called artificial ants that seek suitable solutions to specific optimization problems. Ant colony algorithms optimize complex problems by mapping the problem into a weighted graph, in which ants move along the edge to find the best path. Ant colony algorithms are based on the ability of ants to find the shortest path between their nest and the location of their food, which varies to varying degrees depending on the species of ants. In recent years, many researchers have conducted a lot of research on the application results of ant colony algorithm. The research results show that most of the artificial ant colonies it uses can not provide good solutions, while the elite ant colonies can provide the best solutions through repeated switching technology. Based on the above research theories, researchers proposed a special ant colony optimization method, which can simulate a specific population of ant colonies, in order to solve different types of combinatorial optimization problems.

3.2. The ant system

For the ant colony algorithm convergence speed is slow, easy to fall into the stagnation state defects. Based on the basic ant colony algorithm, the ant colony system has made some improvements in the aspects of state transition, local pheromone update and global pheromone update.

1) The additional random number strategy is introduced into the original rules of state transition to guide the ants to explore the path. This improvement can not only accelerate the ants to approach the better path, but also keep the diversity of the algorithm path solutions.

2) At the end of each cycle, the pheromone is updated only for the solved global optimal path, so the reduction and enhancement of pheromone concentration can only occur on the global optimal path, in order to inspire the ant colony to search the relevant node edges in the range near the optimal path.

3) In each round of circular traversal solution, ants in the ant colony need to update the pheromone locally on the urban road section they pass, that is, after the ants move from the city node \( i \) to the city
node j, they need to cut down the pheromone on the node edge (I, j), which shows that the nodes they have already traveled will become less and less attractive to ants. Then, the ants are guided to explore the untraversed nodes, so as to improve the possibility of the ant colony algorithm approaching the optimal solution.

3.2.1. State transition update rules. In the ACS algorithm, when the ant is located in city I, the next city j can be obtained according to Equations (6) and (7):

\[
J_j = \arg\max_i \left\{ \tau_{is}^\alpha \eta_{is}^\beta \right\}, \text{if } q \leq q_0
\]

\[
p_{ij}^k = \begin{cases} 
\frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum \tau_{ij}^\alpha \eta_{ij}^\beta}, & j \in \text{allowed}_k \\
0, & \text{otherwise}
\end{cases}
\]  

Where \( q_0 \in (0,1) \) and Q is a random number on (0, 1). If \( q \leq q_0 \), the ants in the ant colony will select the city nodes according to the maximum value of pheromone concentration multiplied by heuristic information. If \( q > q_0 \), the selected city node will be carried out probability operation according to Equation (7), and the node j of the next city will be obtained in the way of roulette according to the calculated probability.

3.2.2. Local pheromone update rules. During a traversal, when an ant k located in a city node I selects a city node j from the set of cities to be selected and moves to the city node, the pheromone concentration of node edge (i,j) will be updated immediately according to the following equation:

\[
\tau_{ij} = (1 - \rho_2)\tau_{ij} + \rho_2\tau_{ij}^{(0)}
\]  

Where \( \rho_2 \) represents the local pheromone volatilization coefficient, and \( \tau_{ij}^{(0)} \) represents the initial pheromone value before the algorithm starts.

3.2.3. Global pheromone update rules. In the ACS algorithm, after the end of each iteration cycle, the pheromone is updated according to Equation (9) for the best path from the beginning of the algorithm to the current iteration:

\[
\tau_{ij} = (1 - \rho_1)\tau_{ij} + \rho\Delta\tau_{ij}^{(0)}
\]

\[
\Delta\tau_{ij}^{best} = \begin{cases} 
\frac{1}{L_{best}}, & \text{If } (i,j) \text{ is the global optimal solution} \\
0, & \text{otherwise}
\end{cases}
\]  

Where \( \rho_1 \) is the global pheromone volatilization coefficient and \( L_{best} \) is the global optimal path length.

3.3. Ant Colony Algorithm to Solve TSP Problem

3.3.1. Basic idea of ant colony algorithm to solve TSP problem. 1) Set a relative number of ants according to the specific TSP problem, and assign ants for parallel search; 2) After each ant completes a cycle iteration, it releases pheromone along the edge of the path, ensuring that the quantity of pheromone is proportional to the quality of the solution; 3) Ant traversal path selection should be based on the pheromone concentration (the initial pheromone quantity is set to be equal), and considering the distance between two nodes in the city, a random local search strategy is adopted. As a result, the amount of pheromones on the shorter edge is larger, and the probability of the later ants to choose the edge of the path is also greater;
4) Each ant can only take the legal route (the city after traversal is no longer traversed), and a tabu table is set to control it;
5) If all ants are searched once, it means that a cycle iteration is completed, and each iteration requires a pheromone update for all path edges;
6) Updates of pheromones include the evaporation of original pheromones and the increase of pheromones along the path;
7) If the predetermined number of iteration steps is reached or the phenomenon of stagnation occurs (all ants choose the same path and the solution will not change), it represents the end of algorithm operation and the current optimal solution is taken as the optimal solution of the problem.

3.3.2. Flow chart of ant colony algorithm to solve TSP problem.

![Flow chart](image)

Figure 1. The flow chart

4. Case study

4.1. Problem description

Suppose a traveling merchant wants to visit 31 provincial capitals. He has to choose the route he wants to take. The limit of the route is that he can only visit each city once, and he has to return to the original city. The path selection requirement is: the selected path is the minimum value of all paths. The coordinates of provincial capitals across the country are as follows:

\[
\begin{align*}
[1304 & 2312; 3639 & 1315; 4177 & 2244; 3712 & 1399; 3488 & 1535; 3326 & 1556; 3238 & 1229; 4196 & 1004; 4312 & 790; 4386 & 570; 3007 & 1970; 2562 & 1756; 2788 & 1491; 2381 & 1676; 1332 & 695; 3715 & 1678; 3918 & 2179; 4061 & 2370; 3780 & 2212; 3676 & 2578; 4029 & 2838; 4263 & 2931; 3429 & 1908; 3507 & 2367; 3394 & 2643; 3439 & 3201; 2935 & 3240; 3140 & 3550; 2545 & 2357; 2778 & 2826; 2370 & 2975]
\end{align*}
\]
4.2. Simulation results based on MATLAB
The number of ants is $m=50$, the importance parameter $\alpha=1$, the heuristic factor importance parameter $\beta=5$, the pheromone evaporation coefficient $\rho=0.1$, the maximum iteration time $g=200$, the pheromone increase intensity coefficient $q=100$, and $C$ is the coordinate set of 31 target provincial capital cities. $M$ ants are placed on $N$ cities to calculate the probability distribution of the cities to be selected. $M$ ants select the next city according to the probability function and complete their respective tours. Record the best route of this iteration, update pheromone and clear tabu list. Judge whether the termination condition is met: if the termination condition is met, the search process is ended and the optimization value is output; If not, the iterative optimization is continued.

4.3. Simulation result analysis
1) When the simulation parameters are set as: the number of ants is $50$, $\alpha = 1$, $\beta = 5$, $\rho = 0.1$, the maximum number of iterations is $150$. The simulation results are shown in Figure 2;

![Figure 2. Simulation Experiment 1](image)

2) When the simulation parameters are set as: the number of ants is $100$, $\alpha = 1$, $\beta = 5$, $\rho = 0.1$, the maximum number of iterations is $300$. The simulation results are shown in Figure 3;

![Figure 3. Simulation Experiment 2](image)

3) When the simulation parameters are set as: the number of ants is $100$, $\alpha = 1$, $\beta = 5$, $\rho = 0.5$, the maximum number of iterations is $300$. The simulation results are shown in Figure 5;

![Figure 4. Simulation Experiment 4 $\rho = 0.5$](image)

4) When the simulation parameters are set as: the number of ants is $100$, $\alpha = 2$, $\beta = 5$, $\rho = 0.1$, the maximum number of iterations is $300$. The simulation results are shown in Figure 6;

![Figure 5. Simulation Experiment 5 $\alpha = 2$](image)

5) When the simulation parameters are set as: the number of ants is $100$, $\alpha = 4$, $\beta = 5$, $\rho = 0.1$, the maximum number of iterations is $300$. The simulation results are shown in Figure 7;
6) When the simulation parameters are set as: the number of ants is 100, alpha = 1, beta = 3, rho = 0.1, the maximum number of iterations is 300. The simulation results are shown in Figure 8;

7) When the simulation parameters are set as: the number of ants is 100, alpha = 1, beta = 7, rho = 0.1, the maximum number of iterations is 300. The simulation results are shown in Figure 4;

5. Conclusions
The number of ants is 50, the number of iterations is 150, and the number of ants is 100, the number of iterations is 300. The comparison of the two simulation experiments shows that with the increase of the number of ants and the number of iterations, the shortest distance of simulation results is shortened, from 15828.7082 to 15601.9195. When the number of ants is too large, the amount of pheromones on the searched path tends to average, resulting in the slow convergence of the algorithm; If the number is too small, it is easy to reduce the amount of path pheromones that have not been searched, which weakens the randomness of global search and leads to premature stagnation. Comparative analysis of beta = 3 and beta = 7 with the same number of ants and iterations.

When other factors remain unchanged and beta is changed to 3, it is found that the shortest path becomes smaller, and the starting point of the city sequence is 2 and the ending point is 10. The overall trend of the shortest distance and average distance in each generation is similar to beta = 5. In the shortest distance iteration, there is a sharp decrease in about 10 generations. When beta = 7, the shortest distance of each generation tends to be smooth within 10 generations. The value of beta reflects the importance of heuristic information in guiding ant search. If the beta is too small, the ant colony will fall into random search, so it is difficult to find the optimal solution. If the beta is too large, the more likely the ant is to choose the local shortest path at a local point, but the randomness of ant colony searching for the optimal path is weakened, and it is easy to fall into the local optimum, and the convergence of the algorithm may become worse. In debugging, it is found that the value of beta [3, 6] is more appropriate.
References
[1] PING D, YONG A I. Research on an Improved Ant Colony Optimization Algorithm and its Application [J]. International Journal of Hybrid Information Technology, 2016, 9(4): 223–234.
[2] Banglei Z, Haixia G, Huizong L, et al. Cold Chain Logistics Path Optimization via Improved Multi-Objective Ant Colony Algorithm [J]. IEEE Access, 2020.
[3] Zhaojun Z, Zhaoxiong X, Shengyang L, et al. Opposition-Based Ant Colony Optimization Algorithm for the Traveling Salesman Problem [J]. Mathematics, 2020.
[4] Guohong Y, Zhili F, Tiancan M, et al. Multi-AGVs path planning based on improved ant colony algorithm [J]. The Journal of Supercomputing, 2019.