Eng Int Syst (2018) 2-3: 131–137

Engineering
Intelligent Systems

A solution of travel route planning problem based on improved ant colony algorithms

Yinjun Zhang and Mengji chen*

School of physics and electrical engineering Hechi university Yizhou 546300, Guangxi, China

In order for the ant colony algorithm to be applied to the tourism route planning problem, we need the algorithm to obtain the optimal solution with a relatively high probability when solving the traveling salesman problem. The solution time of the algorithm is relatively short. In this paper, the path selection probability and pheromone updating rules of ant colony algorithm are improved, the local search of the optimal solution is carried out, the solving process of the algorithm is optimized, and the reasonable parameters of the algorithm are determined. Through performance simulation analysis this algorithm has higher search accuracy and a shorter solution time for the traveling salesman problem.

Keywords: Travel Route Planning, Improved Ant Colony Algorithms

1. INTRODUCTION

Tourism route planning can be divided into three parts using the regional scope: tourism that is inter-city, intra-city and also scenic spots. The tourism route planning between cities and scenic spots in the scenic area starts from one point, goes through other points, and only goes through once, and finally returns to the starting point of the route planning, that is, the TSP loop problem [1]. This is a classic NP problem. So there are many solutions. These include: ant colony algorithm, annealing algorithm, genetic algorithm, particle swarm optimization and the *tabu* search algorithm. In this paper, dynamic combinatorial optimization and ant colony algorithm are used to solve the problem. The ant colony algorithm is a simulated evolutionary algorithm. It was first proposed by Italian scholars M. Dorigo, V. Maniezzo, A. Colorini, etc. [2-4]. It is a heuristic search algorithm for combinatorial optimization problems. The algorithm solves TSP problems, assignment problems [5], job-shop scheduling problems and so on. Good results have been achieved. In tourism route planning, the system needs to give an optimal route planning in the shortest time after the user submits the request. In this paper, the improved ant colony algorithm is studied and an encounter algorithm is proposed. The encounter algorithm can improve the quality of an ant's one-time travel and shorten the operation time of the system. Through experiments, the improved ant colony algorithm is studied from two aspects: optimal path solving and dynamic programming. The experimental results

*Corresponding Author. E-mail: 18107786221@163.com

show that the improved ant colony algorithm can solve both the optimal path and the dynamic programming.

Good performance.

Ant colony algorithm (ACO) [6-8] is one of the effective algorithms for tourism route planning. In order to apply the ant colony algorithm to the tourism route planning problem, we need to improve the ant colony algorithm, so that the improved ant colony algorithm can obtain the optimal solution in the shortest possible time. The quality of ant colony algorithm for tourism route planning depends mainly on the path selection strategy, pheromone updating rules and the setting of relevant parameters. The improvement of ant colony algorithm mainly focuses on these three aspects. For the improvement of path selection strategy, aiming at the disadvantage that ant colony algorithm is easy to fall into local optimal solution, it can help ant colony algorithm jump out of the local optimal solution by improving the probability of path selection, local search and other strategies, so as to improve the quality of solution [9-12]. For the improvement of pheromone updating rules, the main focus is on the accumulation and feedback of pheromones, how to make the pheromones on the optimal path play a better role in guiding ants, while avoiding the influence of poor path pheromones on ants. For the setting of relevant parameters, the different traveling salesman problem can make the parameters of the algorithm solution more effective solution. At present, there is no good mathematical proof for the setting of parameters to help determine the quality of parameters. Generally, the range of parameters is determined by experience, and then the size of parameters is dynamically adjusted.

2. IMPROVEMENT STRATEGIES

2.1 Path selection probability

In ant colony algorithm, the probability of path selection of ant k from city i to city j at Time t is $p_{i,j}^{k}$:

$$p_{i,j}^{k} = \begin{cases} \frac{\tau_{i,j}{}^{\partial}\eta_{i,j}{}^{\beta}}{\sum_{s\in allowed_k}\tau_{i,s}{}^{\partial}\eta_{i,s}{}^{\beta}} & if\ j \in allowed_k \\ 0 & else \end{cases} \tag{1}$$

According to (1), we know there are mainly four parameters determining the probability of path selection: pheromone value $\tau_{i,j}$, heuristic function $\eta_{i,j}$, information heuristic factor α, and expectation heuristic factor β.

The heuristic function $\eta_{i,j}(t) = 1/d_{i,j}$, $d_{i,j}$ represents the length of edges (i, j). When the ant colony algorithm is initialized, we let the initial pheromone $\tau_{i,j} = \tau_0$ for each edge (i, j), where τ_0 is constant.

This means that in the initial stage, the ants will have a greater probability to choose the next city with a larger heuristic function (shorter edge distance) to move, so that the global pheromone updates after obtaining the initial solution will make the probability of the ants moving in a fixed direction exceed the probability of moving in other directions, thus forming a local maximum. In the middle and later stages, the size of pheromone in the current optimal path will be significantly different from that in other paths. At this time, ants will have a greater probability to choose the path with larger pheromone, which makes the algorithm stagnate.

In order to solve the above shortcomings, in the early stage, we need to make the ants move more randomly and explore more possibilities of solutions. At the same time, in the middle and late stages of the algorithm, it is possible for ant colony algorithm to jump out of the local optimum and find the optimal solution. Here we achieve the above goal by introducing the concept of random factor. The concrete determination process of the random factor is as follows: Assuming that there are three nodes in the next hop of a node, the calculation of probability passing formula (3.1) of the three nodes is 0.2, 0.3 and 0.5, respectively. According to ant colony algorithm, the node with a probability of 0.5 is chosen as the node of the next hop. This paper carries out the determination through the random factor. The first node is selected in the range of (0,0.2), the second node is selected in the range of (0.2,0.5), and the second node is selected in the range of (0,0.2).

The third node is selected in the range of (0.5,1). By judging the random factors, ants can choose other paths. The principle is to ensure that most ants choose their path according to the normal probability of path selection, but a small number of ants will randomly jump out of the original intrinsic path to choose a new path, which makes ants choose the next city and reduce the probability of going to the relatively short path. In the early stage of the algorithm, it reduces the sensitivity of ants to path length; in the middle and late stage of the algorithm, it reduces the sensitivity of ants to pheromones, avoids the influence of pheromones on ants, thus increasing the probability of jumping out of the local optimal solution.

2.2 Local search

The optimal solution path diagram and local optimal solution path diagram of Oliver30 are shown in Figure 1. Figure 2 shows a partial region of the Oliver30 optimal solution path map and the local optimal solution path map. Analysis and comparison Figure 2.2 (a) and (b) can be seen that point A is closer to point B and farther from point C. Once the amount of pheromone on path AB accumulates, then ants will select edge AB instead of side AC, which leads to the inability to obtain an optimal solution. Analysis of the comparison Figure 2.2 (c) and (d) can be seen that the optimal path is ABC, and the local optimal path is BAC. It can also be seen from the figure that the length of AC is less than the length of BC, so in the initial stage, the ant at point C will select point A as the next city. With the accumulation of pheromones, more ants choose the path CAB. Form a local optimal solution.

Based on the above analysis, we consider a local search method to improve the quality of the ant colony algorithm. Using 2-opt to optimize the optimal path for each cycle, and exchange the adjacent points of the optimal solution of the next cycle, a new solution can be obtained, if the new solution obtained is better than the optimal solution of the current cycle. , then replace the optimal solution of the current cycle. Taking the above example, the local optimal solution path sequence in Figure 2.2(d) is BAC. By 2-opt, the order of exchange points B and A is obtained, ABC is obtained, and then the total length of the path before and after the exchange is compared. The result is better than the current solution result, so that the exchanged path is recorded as the optimal path of the current cycle.

2.3 Pheromone updating rules

In the ant colony algorithm, after each cycle, the global pheromone update utilizes information about the path of all ants. Such an update method means that the pheromone is updated on the optimal path, and the poor path is also updated in the pheromone. Such an update method does not distinguish between high-quality solutions and inferior quality solutions. The accumulation of pheromones on the path of high-quality solutions weakens the guiding role of ants, and the accumulation of pheromones on the path of inferior solutions can mislead ants, which is easy. This means solving the local optimal solution.

This paper considers only using the optimal path to update the pheromone.

$$\tau_{i,j} = \begin{cases} \frac{Q}{L_{best}} & if \text{ edge } (i, j) \text{ is on the optimal path} \\ 0 & else \end{cases} \tag{2}$$

At the same time, we should consider the influence of pheromones on the algorithm. Too much pheromone on the path leads to an increase in the probability of ants choosing this path, while too little pheromone on the path leads ants to ignore the existence of this path when choosing the path, which may lead to the algorithm falling into local optimum. Based on the above analysis, we consider setting threshold for pheromones, which can effectively avoid the occurrence of too large or too small pheromones on the path, and control the positive feedback of pheromones.

$$\tau_{i,j}(t) \in \tau_{\min}, \tau_{\max} \tag{3}$$

In order to prevent the influence of the optimal path from being too large, we consider not only volatilizing the original pheromone on the path, but also volatilizing the new pheromone. The purpose of this is also to avoid the excessive increase or decrease of the size of the pheromone on the path.

$$\tau_{i,j}(t+1) = (1-\rho)\tau_{i,j}(t) + \rho\Delta\tau_{i,j}(t) \tag{4}$$

2.4 Algorithmic solution process

When the ant chooses the next city, it calculates the current path length of ant k immediately, compares the current path length with the current optimal path length. If the current optimal path length exceeds the current optimal path length, the search of ant K stops, which can reduce the solving time of the algorithm.

Set a sign Y to determine whether the optimal solution is updated in this cycle. If it is not updated, there is no need to repeat the calculation of the optimal solution that has been searched locally, which also reduces the solving time of the algorithm to a certain extent.

The termination condition of ant colony algorithm is $N_c \geq N_{\max}$. In fact, it is difficult to get the $N_{\max}$, If the $N_{\max}$ setting is too small, the algorithm has not yet been searched; if the setting is too large, the search is completed after an invalid cycle, which increases the algorithm's solving time. In this paper, the cyclic sign C is set. If the current optimal solution L_{best} is unchanged for 10 consecutive times, it is considered that the current cycle has been completed and the cycle ends.

2.5 Parameter selection

The setting of parameters has a great influence on the ant colony algorithm to solve the tourism route planning, so it is necessary to determine reasonable parameters for the ant colony algorithm. When determining the parameters, considering the importance of the parameters to the algorithm, we first determine the parameters that have great influence, and then determine the parameters that have little influence.

By Ref [13-14], heuristic factor α, expected heuristic factor β and pheromone volatilization coefficient ρ have greater influence, while the influence of ant colony number m and pheromone intensity Q is relatively small.

(1) Heuristic factor α and expectation heuristic factor β

The heuristic factor reflects the guidance function of pheromone on ant colony. The larger the heuristic factor α is, the greater the influence of pheromone on ant selection path, the more likely the ant will choose the optimal path before. When the heuristic factor α is too large, the convergence of the algorithm will be accelerated and the phenomenon of stagnation or prematurity will come about. When the heuristic factor α is too small, the ants are less sensitive to pheromones when choosing the path, and the ants search is more random, thus increasing the consumption time of the algorithm.

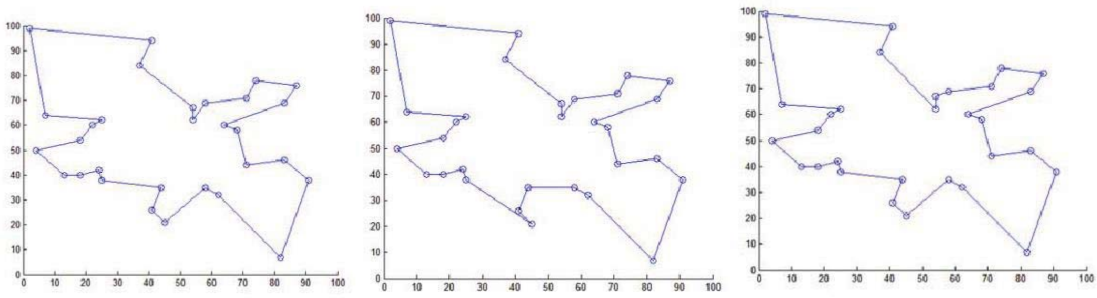
Expectation heuristic factor β reflects the guidance of path information to the ant colony. If the expected heuristic factor β is too large, the ant will choose the shortest path first, which increases the possibility of local optimal solution; if the expected heuristic factor β is too small, the ant will choose the path more randomly, thus increasing the time consumption of the algorithm.

Heuristic factor α and expected heuristic factor β affect the probability $\rho_{i,j}^{k}$ of path selection together. Therefore, when determining the parameters, we must consider the influence of the combination of the two parameters on the algorithm.

In order to select the appropriate heuristic factor α and expected heuristic factor β, we use Oliver-30 as test data. The default values of parameters are pheromone intensity $Q = 100$, pheromone Volatilization Coefficient $\rho = 0.3$, the maximum number of iterations $N_{c-\max} = 100$, ant number $m = 30$, and the combination of heuristic factor α and β expected heuristic factor is $(\alpha, \beta) \in \{(1,3), (1,4), (1,5), (2,3), (3,4), (2,5)\}$. The involves solving each group of combinations 10 times and getting the mean value. The results are shown in Table 1.

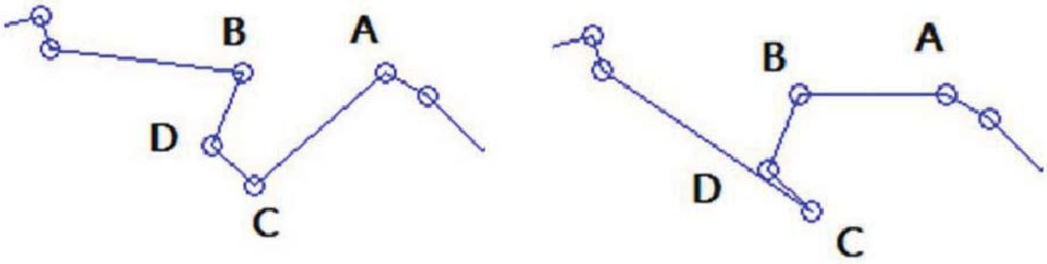
(2) Pheromone volatilization coefficient

Pheromone Volatilization Coefficient ρ and pheromone residue coefficient $1-\rho$ mainly affect the change of pheromone size. If the pheromone Volatilization Coefficient ρ is larger, it will make the pheromone on the path increase or decrease rapidly. It is the ants that cluster together when choosing the next path, so that the algorithm falls into the local optimal solution. If the pheromone volatilization coefficient is smaller, the pheromone on the path changes slowly, and the guidance effect on ants is not obvious. The algorithm is prone to stagnation. In order to select the appropriate pheromone Volatilization Coefficient ρ, we use Oliver 30 as the test data. The default parameters are information heuristic factor $\alpha = 1$, expected heuristic factor $\beta = 4$, pheromone

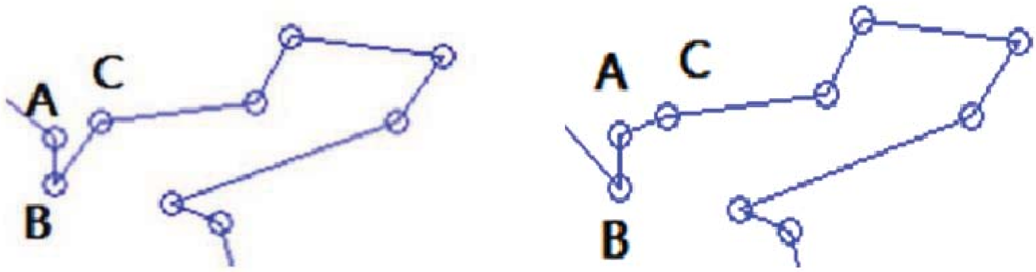


(a)Optimal solution path map (b) Local Optimal Solution Path1 (c) Local Optimal Solution Path2

Figure 1 Optimal and Local Optimal Path Diagrams of Oliver 30.



(a)Partial Regions of Optimal Solution Path (b)Partial region of local optimal solution path 1



(c)Partial Regions of Optimal Solution Path (d)Partial region of local optimal solution path 2

Figure 2 Partial Areas of Oliver 30 Optimal Solution Path Graph and Local Optimal Solution Path Graph.

Table 1 The relationship between heuristic factor α and expected heuristic factor β and average path length.

(α, β)	(1, 3)	(1, 4)	(1, 5)	(2, 3)	(3, 4)	(2, 5)
average path length	424.28	424.15	424.24	424.41	424.27	424.52

intensity $Q = 100$, and the maximum number of iterations of the algorithm $N_{c-\max} = 100$, the number of ants $m = 30$. The volatilization coefficients of pheromone intensity pheromone $\rho = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ are selected to solve 10 times and average the results are shown in Table 2.

Table 2 shows that the average path length is the shortest when the Volatilization Coefficient of pheromone is $\rho = 0.3$.

In the process of pheromone renewal, pheromone Volatilization Coefficient ρ is very important. In most algorithms, pheromone Volatilization Coefficient ρ is usually set as a constant coefficient. In the initial stage of the algorithm, we hope that the volatilization coefficient ρ of pheromone is smaller. In order to avoid the excessive increase or decrease of pheromones in the path, the volatilization coefficient ρ of pheromones should be increased in the middle and later stages, so as to speed up the search. Based on the above considerations, our pheromone Volatilization Coefficient ρ is first linked to the number of cycles, and the pheromone Volatilization Coefficient ρ is

$$\rho = \begin{cases} 0.2 N_c \in 0, 0.35 N_{c-\max} \\ 0.3 N_c \in 0.35 N_{c-\max}, 0.7 N_{c-\max} \\ 0.4 N_c \in 0.7 N_{c-\max}, N_{c-\max} \end{cases} \tag{5}$$

(3) Ant colony number m

When the number of ant colony m is small, the improved ant colony algorithm can get fewer path results each time, so it needs more cycles to find the optimal path. When the number

Table 2 The relationship between pheromone Volatilization Coefficient ρ and average path length.

(ρ)	0.1	0.2	0.3	0.4	0.5	0.6
average path length	425.21	424.17	424.23	423.45	423.76	424.21

of ant colony is large, the operation time of the algorithm itself will increase, so we should consider choosing an appropriate number of ants. In order to select the appropriate ant colony number m, we use Oliver 30 as the test data. The default parameters are information heuristic factor $\alpha = 1$, expected heuristic factor $\beta = 4$, pheromone volatilization coefficient $\rho = 0.3$, the maximum number of iterations $N_{c-\max} = 100$, pheromone $Q = 100$, and $m = \{5, 10, 15, 20, 25, 30, 35, 40\}$ to solve 10 times and average. The results are shown in Table 3.

As can be seen from Table 3, when the number of ants m is 25, the average path length is the shortest. Refs [15-16] show that it is reasonable to consider when the number of ants m is equal to or slightly less than the number of cities n.

(4) Pheromone intensity Q

The function of pheromone intensity Q is to reflect the feedback quantity of pheromone. The larger the pheromone intensity Q is, the larger the change amount of pheromone on the path is. In order to select the appropriate pheromone intensity Q, we use Oliver 30 as test data. The default values of parameters are information heuristic factor $\alpha = 1$, expected heuristic factor $\beta = 4$, pheromone volatilization coefficient $\rho = 0.3$, maximum iteration number $N_{c-\max} = 100$, ant number $m = 25$, and pheromone intensity $Q = \{1, 10, 100, 1000\}$, solve 10 times and get the mean value, the result is shown in Table 4.

As can be seen from Table 4, when the pheromone intensity $Q = 100$, there is an optimal solution, and other cases fall into the same local optimal solution. Ref [13] shows that pheromone intensity Q has little influence on the algorithm, but in small-scale traveling salesman problem, pheromone intensity $Q = 100$ is generally assumed/

3. IMPROVEMENT OF ANT COLONY ALGORITHMS TO REALIZE THE STEPS AND PROCEDURES OF TRAVELING SALESMAN PROBLEM

The steps of improving ant colony algorithm to realize traveling salesman problem are as follows:

Step 1: Initialization of parameters. Let the number of cycles $N_c = 0$, set the maximum number of cycles N_{c-max}, clear taboo table $tabu_K$. Let the initial pheromone $\tau_{i,j}(0) = \tau_{\max}$ for each edge (i, j) and the increment $\Delta\tau_{i,j}(0) = 0$ for the initial time pheromone.

Step 2: Number of cycles $N_c = N_c + 1$;

Step 3: Place m ants in n cities, and then add the city where ant k is located to the taboo table $tabu_k$ of ant k.

Step 4: Number of ants $K = 1$;

Step 5: Ant k calculates the probability of path selection $p^k_{i,j}(t)$ according to (1), decides to move to the next city j according to random factors, and then adds j to the table $tabu_k$ of Ant$k$, calculates the current path length L_k $(k = 1, 2, \ldots)$ of Ant k. If the current path length $L_k > L_{best}$, the search of Ant k stops.

Step 6: The number of ants $K = K + 1$;

Step 7: If $K \geq m$ is satisfied, step 5 is executed; otherwise, step 8 is executed.

Step 8: Calculate the path length L_k of each ant and record the current optimal solution L_{best}. If the optimal solution is updated, the update mark $Y = 1$ and the cycle mark $C = C + 1$; otherwise, the update mark $Y = 0$ and the cycle mark $C = 0$;

Step 9: If the update flag $Y = 1$, the current optimal solution is searched locally to determine whether the optimal solution L_{best} needs to be updated.

Step 10: Update the path pheromone according to formula (2) and formula (4). After updating, determine the pheromone a of each side (i, j), which is larger than $\tau_{\max}$ and amend to $\tau_{\max}$.

Less than $\tau_{\min}$ is amended to $\tau_{\min}$.

Step 11: If $N_c \geq N_{c-max}$ or $C \geq 10$ is satisfied, step 12 is executed; otherwise, the table $tabu_k$ is emptied and step 2 is executed.

Step 12: Output the shortest path. The process ends here.

4. PERFORMANCE SIMULATION ANALYSIS

4.1 Experimental Environment and Program

The test environment of this chapter's simulation experiment is: Core i5-4210M 2.60GHz, 4GB RAM, Win7, Matlab.R2010b.

In order to verify the search accuracy and solution time of this algorithm, the hybrid algorithm of the proposed algorithm and the ant colony algorithm [37] and the particle swarm ant colony algorithm [5] are applied to solve the traveling salesman problem respectively. Oliver 30 problem and eil51 problem are selected as examples. The three algorithms were tested 10 times for each case, and the results of each solution were recorded. The parameters of the simulation are shown in Table 5.

4.2 Search accuracy

Comparing the optimal solution of the algorithm and the ant colony algorithm in the literature [37], we can see that the

Table 3 The relationship between Ant colony number m and average path length.

Ant number	5	10	15	20	25	30
average path length	425.59	425.15	424.53	424.17	424.21	424.18

Table 4 The relationship between Pheromone intensity Q and average path length.

Q	1	10	100	1000	10000	100000
average path length	424.26	424.23	424.22	424.24	424.25	424.23

Table 5 Arithmetic Parameter Settings.

Parameter name	parameter value
information heuristic factor α	1
expected heuristic factor β	4
maximum number of iterations N_{c-max}	100
pheromone intensity Q	100
ant number m	25

algorithm in this paper can get the theoretical optimal solution, while the ant colony algorithm will fall into the local optimal solution. At the same time, comparing the average value and the average error, we can see that although the algorithm in this paper cannot guarantee that every operation can converge to the optimal solution, but only a small probability. The rate falls into the local optimal solution. Comparing with the PSO-ACO hybrid algorithm in literature [5], it can be seen that the average error of PSO-ACO hybrid algorithm is less than that of this algorithm, which shows that PSO-ACO hybrid algorithm has better effect in the solving traveling salesman problem. The reason is that the parameters are fixed in this algorithm, and the hybrid PSO ant colony algorithm determines the parameters dynamically by PSO. At the same time, comparing tables 3.6 and 3.7, we can see that the average error of ant colony algorithm increases rapidly with the increase of the number of cities, which shows that when the number of cities increases, the possibility of ant colony algorithm falling into the local optimal solution is greatly improved; and the average error of this algorithm and particle swarm ant colony hybrid algorithm is also somewhat higher when the number of cities increases. This shows that the hybrid PSO and PSO ant colony algorithm are effective in solving the traveling salesman problem.

Figures 3 and 4 are the optimal path diagrams obtained by the improved ant colony algorithm for solving Oliver 30 and eil51.

4.3 Solution time

By comparing the solution time of the proposed algorithm with that of the Ref [5], the solution time of the proposed algorithm is less than that of the particle swarm Ant Colony Hybrid algorithm. The reason is that the parameters of the proposed algorithm are fixed, and the parameters of the particle swarm ant colony hybrid algorithm are determined dynamically by the particle swarm algorithm, so the search of the proposed algorithm is carried out. The accuracy is slightly lower than that of PSO Ant Colony Hybrid algorithm, but the solution time is faster. Comparing the search accuracy and the solution time, the algorithm proposed in this paper is suited to solving the actual travel route planning problem.

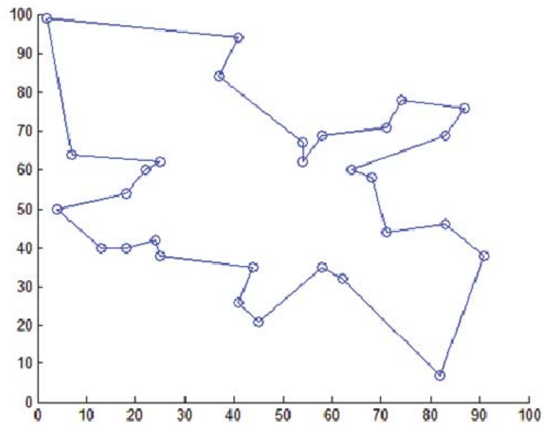


Figure 3 Optimal Path Map of Oliver 30.

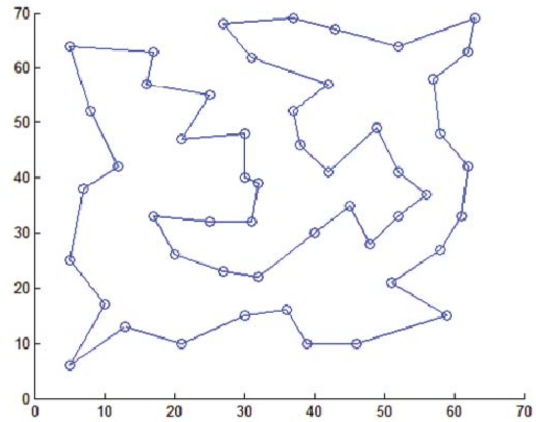


Figure 4 Optimal Path Map of eil51.

5. CONCLUSIONS

In this chapter, a traveling salesman problem based on an improved ant colony algorithm is proposed. On the basis of ant colony algorithm, the probability of path selection is improved by random factors; the optimal path is searched locally after a cycle is completed; the pheromone on the optimal path is updated only, and the threshold of pheromone is set at the same time; the solving process of the algorithm is optimized; and the reasonable parameters of the algorithm are determined. Through the performance simulation analysis, compared with the particle swarm optimization (PSO) ant colony algorithm, the algorithm in this paper has some shortcomings in search accuracy, but the solution speed is faster, which has a better practical value for solving the tourism route planning problem.

Acknowledgments

The work was supported by the Hechi Tourism Master Foundation (2018LG004), Hechi University Foundation (XJ2016ZD004), Hechi university Youth teacher Foundation(XJ2017QN08), the Projection of Environment Master Foundation (2017HJA001,2017HJB001), Guangxi Youth teacher Foundation (2018KY0495).

Authors'contributions

All authors contributed equally and significantly in writing this article. All authors read and approved the final manuscript. Mengji Chen is corresponding author.

Compliance with ethical standards

Conflict of interests The authors declare that there is no conflict of interests regarding the publication of this paper.

REFERENCES

1. Kefi S, Rokbani N, Krömer P, et al. Ant supervised by PSO and 2-Opt algorithm applied to Traveling Salesman Problem [C]. IEEE International Conference on Systems, Man and Cybernetics, 2016: 4866-4871.
2. Mavrovouniotis M, Müller F M, Yang S. Ant Colony Optimization With Local Search for Dynamic Traveling Salesman Problems [J]. IEEE Transactions on Cybernetics, 2016:1-14.
3. Lammel B, Gryzlak K, Dornberger R, et al. An ant colony system solving the travelling salesman region problem [C]. International Symposium on Computational and Business Intelligence, 2016:125-131.
4. Krieger M J, Billeter J B, Keller L. Ant-like task allocation and recruitment in cooperative robots.[J]. Nature, 2000, 406(6799):992-5.
5. Garcia M A P, Montiel O, Castillo O, et al. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation[J]. Applied Soft Computing, 2007, 9(3):1102-1110.
6. Duan H, Yu Y, Zhang X, et al. Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm[J]. Simulation Modelling Practice & Theory, 2010, 18(8):1104-1115.
7. Bertsimas D, Tsitsiklis J. Simulated Annealing[J]. Statistical Science, 1993, 8(1):10-15.
8. Deb K. A fast elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2):182-197.
9. Yang H, Yang H. An Self-organizing Neural Network with Convex-hull Expanding Property for TSP[C]. International Conference on Neural Networks and Brain, 2005:379-383.
10. Qin H, Zhou S, Huo L, et al. A New Ant Colony Algorithm Based on Dynamic Local Search for TSP[C]. Fifth International Conference on Communication Systems and Network Technologies, 2015:913-917.
11. Shi H, Zheng L, Liu G. Research on TSP based on ant colony algorithm[C]. IEEE International Conference on Information and Automation, 2015:2048-2051.
12. He M, Pan D, Yang S. An improved hybrid ant colony algorithm and its application in solving TSP [C]. Joint International Information Technology and Artificial Intelligence Conference, 2014:423-427.
13. Volgenant T, Jonker R. A branch and bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation [J]. European Journal of Operational Research, 1982, 9(1):83-89.
14. Bertsimas D, Tsitsiklis J. Simulated Annealing [J]. Statistical Science, 1993, 8(1):10-15.
15. Deb K. A fast elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2):182-197.
16. Yang H, Yang H. An Self-organizing Neural Network with Convex-hull Expanding Property for TSP[C]. International Conference on Neural Networks and Brain, 2005:379-383.