

Application of an Improved Ant Colony Algorithm in TSP Problem Solving

Weide Ren*, Wenxin Sun

Hebi Polytechnic, Hebi, 458030, China
 rwd121@126.com

Ant Colony Algorithm (ACA) is a kind of heuristic search algorithm based on population search. As a new bionic algorithm, ACA has attracted a lot of experts and scholars' attention. And they deduct many different improved versions. Because of the advantages of ant colony algorithm in solving combinatorial optimization problems, it has been widely applied to various fields such as academic, industrial and commercial fields. Traveling salesman problem (TSP) is one of the classical problems in the field of combinatorial optimization. The traditional ant colony algorithm is early used to solve the TSP problem, but it is limited by its own defects such as slow convergence, easy to fall into the local optimal solution and so on. This paper improves the traditional ant colony algorithm, and it is applied to solve the TSP problem. First of all, we use genetic algorithm to improve ant colony algorithm. By introducing the mutation factor in the algorithm, we can adjust the numerical value of the mutation factor with the algorithm. Therefore, in the initial stage of the algorithm, we can guarantee that the search scope of the ant colony is large enough to prevent the local convergence of the algorithm. Secondly, in order to improve the operation speed of the algorithm, we can optimize the operation process by uniform design in the initial stage of the algorithm. Finally, by integrating these two methods together, we use the improved ant colony algorithm to solve the TSP problem effectively.

1. Introduction

Ant colony algorithm was a kind of bionic optimization algorithm which simulated the foraging behavior of ants in the real world (Ma et al, 2008). There was a very important feature of ant in nature. No matter how complex environment was, ant could always find the shortest path between the nest and the food (Li, 2003). It was well suited to solve large-scale distributed optimization problems without providing global models and without centralized control (Wu and Wang, 2004). In 1991, the Italy scholar Dorigo et al, (1991) put forward the ant colony algorithm. Subsequently, many scholars had entered this field to study the algorithm. Dorigo (1992) further illustrated the core idea of ant colony algorithm in the doctoral dissertation. On the basis of previous studies, Maniezzo and Dorigo (1996) further summarized and put forward an ant algorithm model which was Ant colony optimization (ACO) for solving combinatorial optimization problems. Bullnheimer et al (1999) proposed Rank-based Version of Ant System (ASrank). According to the different levels of the solution, the algorithm updated the pheromone differently. Stützle and Hoos (1996) proposed another improved ant colony algorithm, namely, MAX-MIN Ant System (MMAS). The algorithm defined the upper and lower bounds of the value of the information, and the Trail Smoothing Mechanism (TSM) was adopted in the algorithm. Watanabe I and Matsui S (2003) proposed Adaptive Ant Colony Algorithm (AACA). Li Yue-guang (2008) proposed Quantum and Ant Colony Algorithm (QACA).

TSP problem was always the first choice for the research of ant colony algorithm. In a fixed urban transportation network, we chose the shortest route for the salesman, which started from a point of the network, and went back to it after going through each node (Chen and Dai, 2006). The search space of TSP is the set of all permutations of n city, the size is $n!$. A single arrangement of any n cities produces a solution, which is interpreted as a full path, and is implied to return to the starting point immediately after visiting the last city. TSP problem is an abstract form of complex engineering optimization problems in many fields. It is widely used in the fields of chip design, network routing and vehicle routing (Zhang et al, 2009). Now, TSP is more widely used in how to plan the most reasonable road construction to reduce the traffic congestion. Besides, it

is adopted in how to set the node to spread the information better in the Internet environment, etc. In this paper, based on the traditional ant colony algorithm, the ant colony algorithm is improved to meet the need of expanding the TSP application field.

2. Ant colony algorithm

2.1 The principle of ant colony algorithm

According to the long-term observation and research of biologists, ants search path of backing home relying on crawling pheromone which is secreted by their selves in their path. Studies have shown that ants in finding the source of food, always release a special secretions information- pheromone, to make the other ants within a certain range perceive and affect their future behavior. The ants will give preference to those paths with high concentration of pheromone, according to the paths' pheromone concentration. The pheromone will gradually evaporate with time, and the intensity will gradually weaken (Hossein and Taherinejad, 2009). The shorter the path, the higher the pheromone concentration, and there will be more of the ants. The foraging process of ant colony is a positive feedback process. Figure 1 describes the operation mechanism of ant colony algorithm through the foraging process of ants.

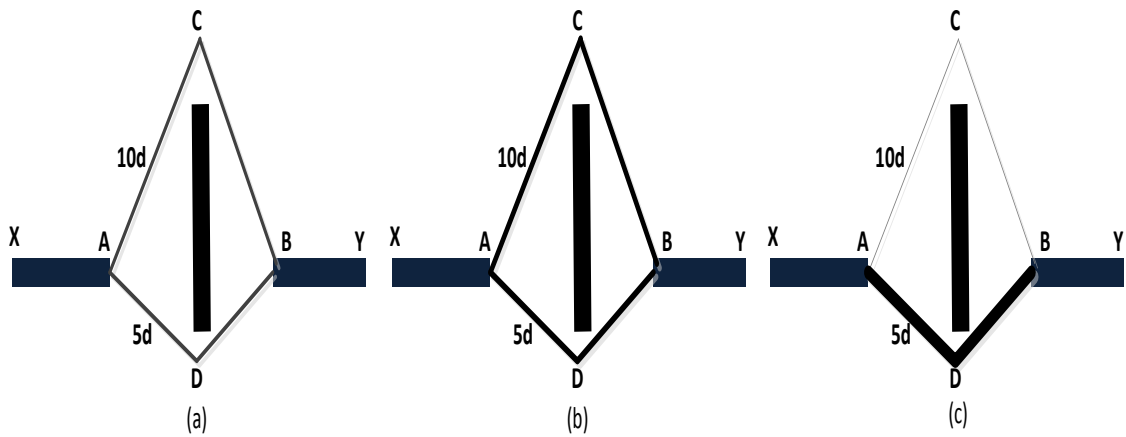


Figure 1: The foraging process of ants

As shown in Figure 1 (a), suppose that X is a nest, Y is food, CD is an obstacle. Initially, ants look for food out of the nest, and they will randomly select the ACB line and ADB line. As shown in Figure 1 (b), over time, the distribution of ants on the two routes will be more uniform. Because the ADB route is shorter, ants on the route come and go faster, leaving behind the pheromone will be more and more. After a period of time, the ants will gradually choose a more pheromone distribution path. As shown in Figure 1 (c), all the ants will eventually choose path ADB, which is the shortest path from their nest to food source.

2.2 Model of ant colony algorithm

Ant colony algorithm is first used to solve the TSP problem. Through the TSP problem, we further analyze the model of ant colony algorithm. Suppose m is the number of ants in the ant colony; d_{ij} is the distance between city i and city j ; $b_i(t)$ represents the number of ant in city i in time t ; τ_{ij} represents the residual amount of pheromone in path (i,j) ; $\Delta \tau_{ij}^k$ is the amount of pheromone released by the k th ant on the path (i,j) ; ρ ($\rho \in (0,1)$) indicates the degree of volatilization in the unit time of residual pheromone; η_{ij} represents heuristic information of path (i,j) , general $\eta_{ij} = 1/d_{ij}$; α represents the weight of the pheromone, β represents the weight of the heuristic information, $tabu_k$ records the cities ant k has passed by; $allowed_k$ represents the next city location of the ant k selection; l_k represents the distance the ant k has walking by; p_{ij}^k represents the choice probability of ant k transferred from position i to position j .

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in allowed_k} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta} & j \in allowed_k \\ 0 & j \notin allowed_k \end{cases} \quad (1)$$

If ants complete a traversal, that is, the ants find a traversal of all the cities, the pheromone update on each path according to the formula 2.

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}, \Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (2)$$

In ant colony algorithm, the update strategy of pheromone is directly related to the efficiency and success of the algorithm, and update strategy of pheromone will be chosen based on the characteristics of the problem to be solved. Some scholars proposed three different basic ant colony algorithm models: Ant-Quantity System, Ant-Density System, and Ant-Cycle System. Figure 2 is the basic framework of ant colony algorithm.

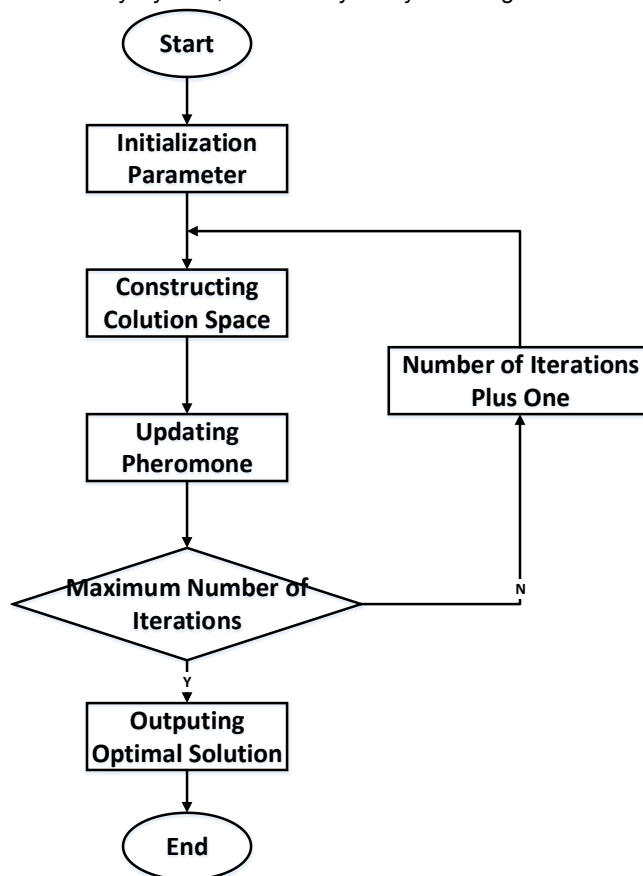


Figure 2: The basic framework of ant colony algorithm

3. Uniform discrete process optimization based ACA

Uniform design is an experimental design method. Uniform design only considers the uniform dispersion of the test point in the experimental range, the selection of the starting point of the representative point is evenly dispersed, without considering the uniformity ratio. Using uniform discrete process optimization based ant colony algorithm to solve the TSP problem, we first build a uniform discrete process, which can form the deeply uniform discrete sub population search map and globally uniform discrete search map. Secondly, in each sub population, we implement the deeply uniform discrete search, that is, the repeated uniform discrete search. We judge the change tendency of the objective function value, and then seek the extreme to obtain the local optimal solution of the sub population. Finally, we put the local optimal solution of each sub population association into the global space uniform discrete node path, and then carry on the global uniform discrete search to find out the optimal solution of objective function.

4. Genetic algorithm

Genetic algorithm is a random search algorithm, which introduces biological evolution into the mathematical model, and has a global and strong robustness. The general genetic algorithm consists of 4 parts: the coding mechanism, the control parameter, the fitness function and the genetic operator. According to the

characteristics of global search capability of genetic algorithm, this paper applies the idea of genetic algorithm in ant colony algorithm, which strengthens the search ability of ant colony algorithm, and avoids the premature convergence.

5. Improvement and Implementation of algorithm

In this paper, the improved algorithm combines the advantages of fast discrete search based on uniform discrete and the global search capability of genetic algorithm. The improved ant colony algorithm not only has fast convergence rate, but also avoids the defects of fast convergence that is easy to fall into local optimum in solving TSP problem.

- 1) First, ants are distributed in the solution space according to the uniform discretization method, and we record the coordinates of ants in the problem space position at a fixed time.
- 2) Second, the pheromone value on each path is not unlimited, but gives it a range $[\tau_{min}, \tau_{max}]$. In this way, we can prevent the colony from falling into local convergence which is caused by an increase in the local pheromone.
- 3) Global pheromone update

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot Q \quad (3)$$

Among them, ρ is the volatile coefficient of pheromone, and its value range is $[0, 1]$. Q is the value of the initial pheromone. In order to increase the search range at the start time, Q is usually taken $Q = \tau_{max}$.

- 4) Local pheromone update

$$\Delta \tau_{ij}^{best} = \begin{cases} \frac{1}{C^{ib}} \\ 0 \end{cases} \quad (4)$$

If (i, j) belongs to the optimal path of this iteration, local pheromone will take $1/C^{ib}$. After the end of the iteration process, only the optimal path is updated, and the C^{ib} is the length of the optimal path found in this iteration.

- 5) Adding mutation factor

$$C_{ij} = \begin{cases} \arg \max_{j \in allowed} \{ \tau(i, j) \cdot \eta^\beta(i, j) \} & q \leq q_0 \\ 0 & otherwise \end{cases} \quad (5)$$

C_{ij} is a state variable. q_0 is a constant, and its value range is $[0, 1]$. q is a random number $(0, 1]$, ant colony in the selection of the next city randomly generate q value. When the value of q is less than or equal to q_0 , the next shortest distance is selected. If q is greater than q_0 , we select the city following the following formula.

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum [\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta} & j \in allowed_k \\ 0 & j \notin allowed_k \end{cases} \quad (6)$$

Adding the mutation operation can make the ants in the first will not blindly follow the guidelines of pheromone search path. This will expand the scope of the search, and avoid falling into quickly local optimum that leads to stagnation.

- 6) Depth-uniform discrete sub-populations search
- 7) Global uniform discrete search

According to the principle of dynamic transition probability and pheromone update, we move all ants within the ant colony.

6. Experiment and analysis

6.1 Introduction of experimental environment

First, before the experiment, the parameters of the algorithm need to be determined. In this paper, the selected experimental examples are Oliver30 in the TSPLIB. According to the uniform design method, the

optimal parameter combination is obtained through repeated iteration. $\alpha=4.5$, $\rho=0.47$, $m=18$, $\beta=3.8$, because of the effect of the total amount of pheromone Q on the algorithm is not obvious, we take $Q=89$.

6.2 Experimental results and analysis

This paper compares the traditional ant colony algorithm and the improved ant colony algorithm. The iteration times of the experiment are 50, 100, 300, 500 and 1000 times. Table 1 shows the experimental results of the traditional ant colony algorithm, table 2 shows the experimental results of the improved ant colony algorithm.

Table 1: The results of traditional ant colony algorithm

Shortest path length	Average path length	Number of iteration
472.436	495.147	50
464.592	483.223	100
454.267	468.812	300
448.677	462.954	500
442.217	452.651	1000

Table 2: The results of improved ant colony algorithm

Shortest path length	Average path length	Number of iteration
427.368	443.429	50
425.445	438.186	100
423.678	428.741	300
423.173	426.985	500
420.863	423.434	1000

It can be seen from the tables that the improved ant colony algorithm has obvious advantages in dealing with the TSP problem. First of all, through the uniform design of the idea to set up a good parameter in advance. At the same time, we add the mutation factor to enhance the searching ability of the ant colony algorithm, and the convergence speed is improved. In this paper, the algorithm has obvious advantages in dealing with TSP problem.

7. Conclusions

The convergence speed of ant colony algorithm is greatly improved by using the uniform design idea to add uniform discrete process and the reasonable setting of the parameters of ant colony algorithm. By introducing the idea of genetic algorithm and adding the variation factor, the improved ant colony algorithm has a fast convergence speed and very good global optimization ability, thus avoiding the local convergence of the situation. Through experiments, we verify that the improved ant colony algorithm has a very good performance in dealing with the TSP problem. And it is proved that the improved algorithm can effectively solve the problem of other combinatorial optimization problems.

References

- Bullnheimer B., Hartl R.F., Strauss C., 1999, A new rank-based version of the ant system: a computational study. *Central European Journal for Operations Research and Economics*, 7(1): 25~38
- Chen W.L., Dai S.G., 2006, Review of traveling salesman problem algorithm [J]. *Journal of Chuzhou University*, 8(3): 1-6.
- Colomi A., Dorigo M., Maniezzo V., et al, 1991. Distributed optimization by ant colonies [C]. *Proceedings of the 1st European Conference on Artificial Life*: 134-142
- Dorigo M., Maniezzo V., Colomi A., 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Transation on Systems, Man, and Cybernetics-Part B*, 26(1): 29-41
- Dorigo M., 1992. Optimization, learning and natural algorithms. Ph.D. Thesis, Department of Electronics, Politecnico di Milano, Italy.
- Hossein M.N., Taherinejad N., 2009. New robust and efficient ant colony algorithms: Using new interpretation of local updating process. *Expert System with Application*, 36: 481-488
- LI S.Y., 2003, Progresses in Ant Colony Optimization Algorithm with Applications [J]/ *Computer Automated Measurement & Control*, 11(12): 911-913.
- Li Y.G., 2008, Quantum Ant Colony Algorithm and Its Application [D]. Lanzhou University of Technology.
- Ma L., Zhu G., Ning A.B., 2008, *Ant Colony Optimization Algorithm*, Beijing: Science Press.
- Stützle T., Hoos H., 1996, Improving the ant system:a detailed report on the MAX-MIN ant system. Technical Reprot AIDA-96-11, FG Intellektik, TH Darmstadt,
- Watanabe I., Matsui S., 1996, Improving the performance of ACO algorithms by adaptive control of candidate set. *Proceedings of the 2003 Congress on Evolutionary Computation*, 2: 1355-1362
- Wu Q.D., Wang L., 2004, *Intelligent ant colony algorithm and Application* [M]. Shanghai Scientific and Technological Education Publishing House, 1-200
- Zhang Y.D., Wu L.N., Wei G., 2009, Comparison on solving TSP via intelligent algorithm [J], *Computer Engineering and Applications*, 45(11): 11-15.