Implementation and Improvement of ACO Algorithm

Dai Xiangting 44211580

April 2, 2023

Abstract

In computational intelligence, evaluation algorithm is an important subset of evolutionary computation. Ant colony optimization called ACO algorithm is a famous algorithm for optimization problems.This report aims to implement ACO algorithm to resolve TSP problem and analysis the performance, disadvantages and imperfections of original ACO implementation. Otherwise, MMAS, a classic improvement of ACO algorithm model is implemented, can avoid numerous disadvantages of traditional ACO algorithm. How to implement MMAS and compared it with traditional ACO algorithm is also a significant part of this report.

1 Introduction

Ant colony system(ACS) is a classical intellectual algorithm which is enlightened from ant colony. Marco Dorigo put forward this idea [1] in 1992 to find a method to resolve Traveling Salesman Problem(TSP). ACS method has a set of agents, called ants, search in parallel for good solutions to the TSP, and cooperate through pheromone-mediated indirect and global communication.[1] TSP is a typical optimal problem which is aimed to find the minimal path to travel all cities only ones. With the searching by ant colony again and again, ants tend to an optimal path because this path has higher concentration of pheromone. So we can use ACS to find optimal path to resolve TSP. ACS is the core of ACO algorithm. Surely, ACS has some disadvantages and now there are some modified model to improve ACS to realize higher performance for TSP.

2 ACO Algorithm

2.1 Basic Principle

Ant System is a self-organized algorithm which can find the optimal resolution as the number of ant system iterations increases. Ants can secrete pheromone to hint other ants that there is a better way. After many iterations, all ants trend to move to optimal way with higher concentration of pheromone. So, AS is commonly used in TSP problem. In TSP, we need travel all cities but we only can travel every cities one time and we need find the optimal length to accomplish this task.

Firstly, we need consider the details of TSP. Commonly, we should know the position data of each cities and we can use these data to get the distance matrix showing distance relation with

each city. the distance data between two cities can be defined as:

$$d_{i,j} = \sqrt{[(x_i - x_j)^2 + (y_i - y_j)^2]} \tag{1}$$

where $d_{i,j}$ is the distance between city i and city j. We can construct distance matrix by this equation(1).

Secondly, ants will be distributed into different cities. The number of ants is usually smaller than the number of city. Ants will choose their next cities by distance. we assume $\eta_{i,j} = 1/d_{i,j}$,the probabilities of choosing next cities for ants are determined by η called heuristic value. We assume a *tabu* matrix to show the degree of residual pheromone, The specific calculating expression of transition probability $P_{i,j}(t)$is:

$$P_{ij}(t) = \frac{\tau_{ij}(t)^{\alpha} \left(\eta_{ij}\right)^{\beta}}{\sum\limits_{j \in \text{ allowed nodes}} \tau_{ij}(t)^{\alpha} \left(\eta_{ij}\right)^{\beta}}, \quad \eta_{ij} = \frac{1}{d_{ij}} \tag{2}$$

where t is the number of cities which are travelled. The next city j will be chosen by this probability (rather than choose the argmax$(p^k_{i,j}(t))$ directly).

α, β are two adjusted parameters. α shows the level of relative importance of pheromone, and β represents level of relative importance of heuristic value. After choosing next cities, ants will secret pheromone. At the same time we need define the evaporating parameter Q which can influence residual ratio of pheromones from ants. Based on Ant Cycle model, the pheromone deposits are calculated by:

$$\Delta\tau^k_{i,j}(N) = \begin{cases} \frac{Q}{L_k} & \text{if } (i,j) \text{ The k-th ant passed (i,j)} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where L_k is the total length that the k-th ant travelled in the N-th iteration.The other factor influencing pheromone evaporation is length between two cities. If the length is too long, the residual ratio of pheromone will be too small. Ants can remember cities which they have travelled.

If all ants have finished travelling of all cities, we need update tabu matrix by:

$$\tau_{ij}(N+1) = (1-\rho)\tau_{ij}(N) + \Delta\tau_{ij}(N), \ \Delta\tau_{i,j}(N) = \sum_{k=1}^{m} \Delta\tau^k_{i,j}(N) \tag{4}$$

where $0 < \rho < 1$ is a parameter which expresses the evaporation degree of pheromone. This parameter can make this algorithm has a negative feedback. After iterating again and again, the better path will have higher residual pheromone. Therefore, ant colony tends to choose better path with higher residual pheromone. It is the key of ACO algorithm and it is the reason why ACO algorithm can find the optimal path to resolve TSP after enough iterations.

2.2 Implementation

Firstly, we set and initialize some parameters: $\alpha, \beta, \rho, \tau_0, m, n, N_{max}$. τ_0 is the initial value of *tabu* matrix. m is the number of ants and n is the number of cities. Then, we calculate the distance matrix *dist* and heuristic value matrix *eta*. Besides, the path matrix and best path length record list are also should be created. Secondly, m and n are given by the scale of TSP.

Ants are distributed into different cities randomly. All ants choose next cities by expression (2) and add next cities into their visited lists respectively until every ant travelled all cities. Thirdly, we need to update the *tabu* matrix by expression (4) until the current iteration number n is equal to N_{max}. Furthermore, if some ants find better path than before, update the best path list and best length list. After N_{max} iterations, we can get the relative optimal resolution of TSP.

Pseudo-code:

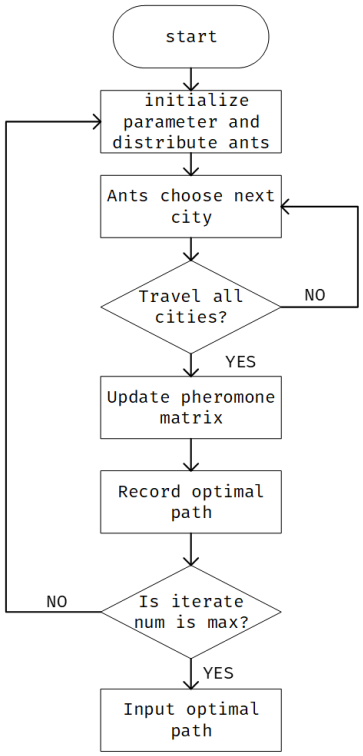
Algorithm 1 Basic ACO algorithm

Input: $\alpha, \beta, \rho, \tau_0$, m:current iteration number, n:city number, N_{max}:maximum iteration number;

Output: best_path_list; best_length_list;

```
N = 0; initialize best_path_list, best_length_list; distribute ants into cities;;
while N<N_max do
    t = 0; N = N + 1   while t < n do
        k = 0; t = t + 1   while k < m do
            k = k + 1; k-th ant ants choose next cities by expression (2)
            add next city into its visited lists.
    Every ant travelled all cities, update tabu matrix by (4);
    update best_path_list and best_length_list;
return best_path_list and best_length_list
```

Flow chart:



2.3 Parameter Analysis

There are six main parameters which we need set and control. Especially, α, β and ρ will influence the convergence performance of ACO algorithm. The city number is determined by TSP scale. Commonly, we set ant population at about two-thirds of the number of city. The iteration number is set as 500 in order to has enough times for convergence.

α shows the importance of residual pheromone and β shows the importance of heuristic value, α reflects the randomness in searching path. If α is equal to zero, ants will tend to choose previous path. This algorithm will degenerate to greed algorithm and small α can lead to local optimization.

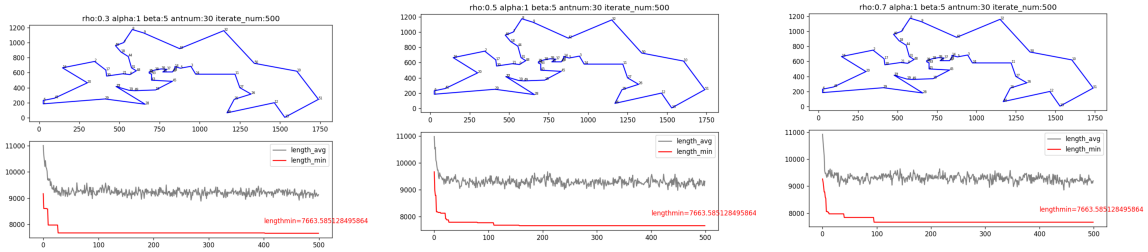
β shows the certain factors in this algorithm. If β is so small, algorithm will become random searching algorithm. It means we are hard to find a optimal resolution. This algorithm is difficult to converge. However, if β is too large, algorithm will converge fast and get a local optimal resolution.

ρ determines the evaporation speed of pheromone. If ρ is small(approach zero), the positive feedback of pheromone will be strong and convergence speed is also fast but the quality of convergence is not good. Larger ρ value means higher negative feedback of pheromone and the randomness is higher. So, the it will make this algorithm is hard to find optimal resolution. Q is the intensity of pheromone which is shows the pheromone quantity secreted at path in one iteration. we often set a constant value like 1. Higher Q means fast convergence and lower Q will improve the randomness of searching.

2.4 Result

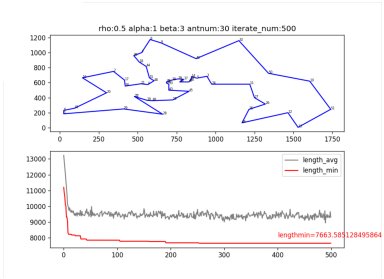
I using python as the basic programming language to implement the ACO algorithm and use the matplotlib library to plot the iteration processing. The TSP data is "Berlin.tsp" which has position data of 52 German cities. Because of 52 cities, I set ant number as 30, the maximal iteration number is 500. As for α, β and ρ, I set α = 1, 2, 0.5. (actually α=1 is the best and other value will lead very bad result.) β is set as 3, 5, 7, and ρ is set as 0.3, 0.5, 0.7. Q is a constant value set as 1.

Figure 1-11 show the final path of results and the iteration processing of ACO algorithm. Figure 1-3 shows the results where α=1, β=5 and ρ = 0.3, 0.5, 0.7

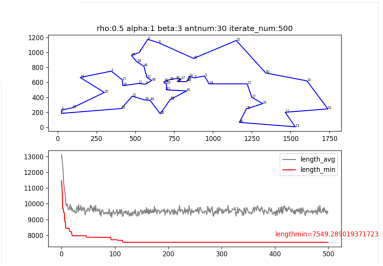


(a) figure 1 (b) figure 2 (c) figure 3

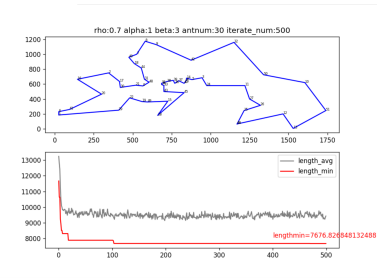
Figure 4-6 shows the results where α=1, β=3 and ρ = 0.3, 0.5, 0.7



(d) figure 4

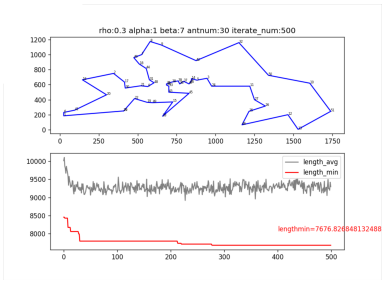


(e) figure 5

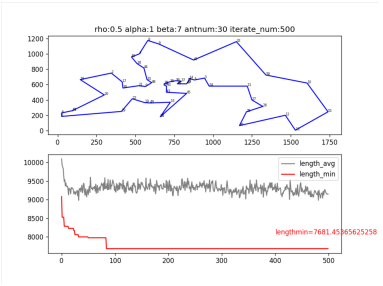


(f) figure 6

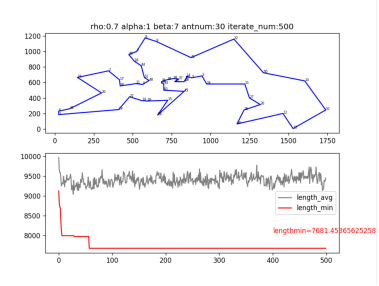
Figure 7-9 shows the results where α=1, β=7 and ρ = 0.3, 0.5, 0.7



(g) figure 7

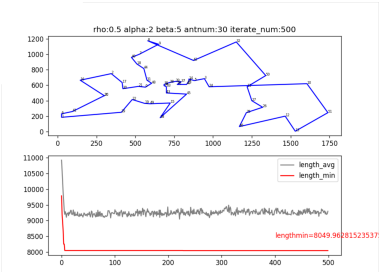


(h) figure 8



(i) figure 9

Figure 10 shows the results where α=2, β=5 and ρ =0.5



(j) figure 10

We can see the parameter and result table at follow:

parameter	parameter	parameter	parameter	parameter	parameter
antnum=30	antnum=30	antnum=30	antnum=30	antnum=30	antnum=30
Nmax=500	Nmax=500	Nmax=500	Nmax=500	Nmax=500	Nmax=500
rho=0.3	rho=0.5	rho=0.7	rho=0.3	rho=0.5	rho=0.7
alpha=1	alpha=1	alpha=1	alpha=1	alpha=1	alpha=1
beta=3	beta=3	beta=3	beta=5	beta=5	beta=5
Q=1	Q=1	Q=1	Q=1	Q=1	Q=1
length=7549	length=7549	length=7676	length=7663	length=7663	length=7663
parameter	parameter	parameter	parameter	parameter	
antnum=30	antnum=30	antnum=30	antnum=30	antnum=30	
Nmax=500	Nmax=500	Nmax=500	Nmax=500	Nmax=500	
rho=0.3	rho=0.5	rho=0.7	rho=0.5	rho=0.5	
alpha=1	alpha=1	alpha=1	alpha=2	alpha=0.5	
beta=7	beta=7	beta=7	beta=5	beta=5	
Q=1	Q=1	Q=1	Q=1	Q=1	
length=7676	length=7663	length=7681	length=8050	length=7793	

2.5 Discussion

We can easily see that higher α value will lead wrong result and the iteration gradient is so steep. Higher ρ value can make algorithm more random, lower ρ value means more steep gradient. The changing of β has a little influenced on iteration result, I guess the changing scale of β is not large enough and there are other factors which have stronger effects of algorithm.

To sum up, ACO algorithm with ACS can get a relative better resolution with enough iteration times and suitable parameters. Although ACS model has improved original Ant system, there are also not idea. These results are just closed to optimal resolution and the result is not stable. This algorithm is easy to find a local optimal resolution and hard to converge a stable resolution. In addition, high steep gradient also means it is not a great algorithm, though it can easily find an approximate resolution.

3 MMAS Model

3.1 introduction

MAX–MIN Ant System (MMAS) algorithm discussed a improved pheromone update way and set a range of to restrict the value of each iteration τ. It uses a rather simple mechanism for limiting the strengths of the pheromone trails effectively avoids premature convergence of the search. There are three key steps of MMAS model[2]:

1. After each iteration, only one single ant adds pheromone. This ant may be the one which

found the best solution in the current iteration or the one which found the best solution from the beginning of the trail.

2. To avoid stagnation of the search the range of the possible pheromone trails on each solution component is limited to an interval [τmin, τmax]

3. Initialize the pheromone trails to τmax , achieving in this way a higher exploration of solutions at the start of algorithm.

3.2 Improvements

There are 3 main advantages of MMAS model.

1. it only allows one ant which is find the best path in this iteration to secrete the pheromone. the pheromone in other paths only evaporate in current iteration.

2. the value of pheromone in each path is limited between τmin and τmax. If the value exceeds this range, the value will be set as τmin or τmax. It can avoid the pheromone value of path has too higher concentration of pheromone which can lead local optimal resolution.

3. The initial value pheromone of each path is set as τmax. Pheromones in worse paths will be reduced by pheromone update(evaporating). Meanwhile, the evaporating parameter ρ is set as a small value to make sure the MMAS model has a stronger searching ability at the begin of iteration.

3.3 Implementation

Compared with ACS model and traditional AS model, MMAS has different pheromone update expression:

$$\tau_{ij}(N+1) = (1-\rho)\tau_{ij}(N) + \Delta\tau_{ij}^{best} \tag{5}$$

$$\Delta\tau_{i,j}^{best} = \begin{cases} \frac{1}{L^{best}} & \text{if } (i,j) \in \mathrm{T}^{best} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

For each iteration, we only just update the pheromone in the best path. Actually, we can choose the local best path to update and we also can choose current global best path.

The other problem of MMAS model is how to preset the τmax and τmin. According to MAX–MIN ant system[2], the τmax and τmin can be equal to:

$$\tau_{max} = \frac{1}{1-\rho} \cdot \frac{1}{f(s^{opt})} \tag{7}$$

$$\tau_{min} = \frac{\tau_{max}(1 - \sqrt[n]{P_{best}})}{(avg-1) \cdot \sqrt[n]{P_{best}}} \tag{8}$$

where ρ is residual ratio of pheromone rather than evaporating ratio. So, evaporating ratio is equal to 1-ρ in this expression (7). As for $f(s^{opt})$, it is the global best length. And the avg is one second of city number. P_{best} is usually initialized to 0.05 [2]. Otherwise, the initial value of $\tau(0)$ of tabu matrix is set as an arbitrary value[2].

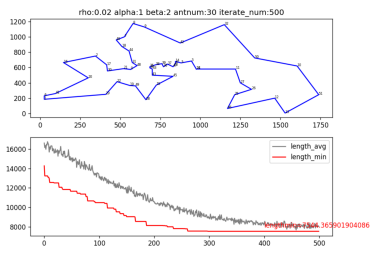
The basic algorithm flow of MMAS model is similar to ACS or traditional AS model, We only need to adjust the rule of pheromone update and limited the pheromone value[2].

3.4 Result and Comparison

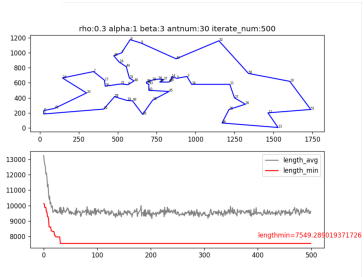
In MMAS experiment, the parameters are set as follow:

1. antnum = 30, $N_{max} = 500$.(similar to AS model)
2. $\alpha = 1$ $\beta = 2$
3. $\rho = 0.05$ This ρ is the evaporating parameter(equal to 1 - residual ratio in MMAS)

The MMAS result is shown in Figure 11. As a comparison, the figure 12 is a result by traditional ACO algorithm. Compared with the best result of traditional AS algorithm, it is



(k) figure 11



(l) figure 12

easily to be found that ACO algorithm by MMAS model is better because of smoother iteration curve and smaller length. What's more, MMAS model has higher algorithm stability. When I did original ACO algorithm experiments, I usually got different results of convergence. But in MMAS model, almost every time the algorithm by MMAS model can converge the same result and this result is better than any results by original ACO algorithm.

4 Summary

In this report, The main part is the basic principle and implantation of ACO algorithm. ACO algorithm is an intellectual algorithm which can evolve continually with iterations. It is similar to Simulated Annealing (SA) algorithm, Genetic Algorithm (GA) and Particle Swarm Optimization(PSO) algorithm. In ACO algorithm, different parameters have different influence on results. So We always need enough time to try to find suitable parameters. Furthermore, improvement is also an important part of ACO algorithm. Based on original Ant System, people have developed many advanced model such as MMAS model to improve ACO algorithm.

ACO algorithm is inspired by ant colony. It uses biological knowledge intelligently to resolve NPC or NP-hard problem. Choosing a creative model enlighten by nature creatures and improve it constantly until it has a great performance for targets. This is a useful train of thought in science research.

References

[1] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53–66, 1997.

[2] T. Stützle and H. H. Hoos, "Max–min ant system," *Future generation computer systems*, vol. 16, no. 8, pp. 889–914, 2000.