

A New Optimization Algorithm Based on Ant Colony System with Density Control Strategy

Ling Qin^{1,3}, Yixin Chen², Ling Chen^{1,3}, Yuan Yao¹

¹ Department of Computer Science, Nanjing University of Aeronautics and Astronautics,
210093, Nanjing, China
qllynne@nuaa.edu.cn

² Department of Computer Science and Engineering, Washington University
in St. Louis St. Louis, MO 63130-4899, USA
chen@cse.wustl.edu

³ Department of Computer Science, Yangzhou University, 225009, Yangzhou, China
lchen@yzcn.net

Abstract. A new optimization algorithm based on the ant colony system is presented by adopting the density control strategy to guarantee the performance of the algorithm. In each iteration of the algorithm, the solutions are selected to have mutation operations according to the quality and distribution of the solution. Experimental results on the traveling salesman problem show that our algorithm can not only get diversified solutions and higher convergence speed than the Neural Network Model and traditional ant colony algorithm, but also avoid the stagnation and premature problem.

1 Introduction

Ant Colony algorithm(AC) was introduced by Dorigo, Maniezzo, and Colormi to solve the Traveling Salesman Problem (TSP) [1]. With the further study in this area, ant colony algorithm has been widely applied to complicated combinatorial optimization problems. However, the classical ant colony algorithm also has its defects, excessive positive feedback could cause premature solutions and local convergence. The major factor causes this two problems mentioned above is its lack of a diversity protection mechanism, which can keep the balance between the convergence speed and the quality of the solutions.

In this paper, we present a new type of ant colony algorithm using the idea of density control strategy. In this algorithm, the individuals are selected to have crossover and mutation operations according to their fitness value and diversity. Experimental results on the traveling salesman problem show that our density controlled ant colony algorithm(DACA) can get diversified solutions and higher convergence speed than

* Supported in part by the Chinese National Natural Science Foundation under grant No. 60473012, Chinese National Foundation for Science and Technology Development under contract 2003BA614A-14, and Natural Science Foundation of Jiangsu Province under contract BK2005047.

the Neural Network Model [2](NNM) and traditional ant colony algorithm[3](TA), but also avoid the stagnation and premature problem.

2 The Classical Ant Colony Algorithm

Here we use TSP as an example to illustrate AC and its application. We denote the distance between city i and j as d_{ij} ($i, j = 1, 2, \dots, n$). Let $\tau_{ij}(t)$ be the intensity of trail information on edge (i, j) at time t , and use $\tau_{ij}(t)$ to simulate the pheromone of real ants. Suppose m is the total number of ants, at time t the k th ant selects from its current city i to city j according to the following probability distribution:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{r \in allowed_k} \tau_{ir}^\alpha(t) \eta_{ir}^\beta(t)} & j \in allowed_k \\ 0 & otherwise \end{cases} \quad (1)$$

Where $allowed_k$ is a set of the cities can be chosen by the k th ant at city i for the next step, η_{ij} is a heuristic function which is defined as the visibility of the path between cities i and j , for instance it can be defined as $1/d_{ij}$. The relative influence of the trail information $\tau_{ij}(t)$ and the visibility η_{ij} are determined by the parameters α, β . The intensity of trail information should be changed by the updating formula, where $0 < \rho < 1$, and $(1-\rho) \tau_{ij}(t)$ represents the evaporation of $\tau_{ij}(t)$ between time t and $t+1$, $\Delta\tau_{ij}$ is the increment of $\tau_{ij}(t)$. And $\Delta\tau_{ij}^k$ is the trail information laid by the k th ant on the path between city i and city j in step t , it takes different formula depending on the model used. in the most popularly used model called ‘‘ant circle system’’ it is given as Eq.(4), where Q is a constant, L_k is the total length of current tour traveled by the k th ant.

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \quad (2)$$

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k & \text{if the } k \text{th ant passes edge } (i, j) \text{ in current tour} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

3 The Density Controlled Ant Colony Algorithm

Here, we present the density controlled ant colony system to improve the optimization capability of the ant colony algorithm. And we use it to solve different kinds of TSP to evaluate its performance. In fact, the experimental results have proved that our algorithm is suitable for solving not only TSP but also QoS routing problem, clustering problem and continuous optimization problems etc.[5].

3.1 Framework of The Density Controlled Ant Colony Algorithm

It's obvious that the ant colony algorithm is mainly composed by an iterative course including searching and verification for the solutions. Although evolution of the solutions of each individual can be accelerated by positive feedback, sometime it may also be degenerated inevitably. Thus, we make full use of the characteristic of the problems to be solved and adopt the conception and theory of density control strategy [4] to restrain the degeneration in the iteration. The framework of the algorithm is given as follows:

Begin

1. initialize
 - 1.1 Generate N_p initial solutions randomly, compute their fitness and set the initial trail information on edge(i, j)
2. **While** ($t \leq N_c$)
 - 2.1 **for** $k=1$ to N do //for n cities
 - for** $i=1$ to m do //for m ants
 - 2.2.1 The ant i select the next city j with the probability given by Eq.(5);
 - 2.2.2 Locally update the trail information on edge(i, j) according to Eq.(4);
 - end for** i
 - end for** k
 - 2.2 Select some solutions in current colony to have mutation operation according to Eq.(10)
 - 2.3 have global trail information updating operation, update the pheromone on the edges between each city according to Eq.(10);
 - 2.4 $t++$; if suffice the convergence condition then break the while loop;
 - end while**

End

In the line 2.2.1, the next city j will be selected according to formula(5):

$$j = \begin{cases} \arg \max \tau_{ij}(t), j \in allowed_i & q \leq q_0 \\ \text{select the next city } j_0 \text{ according to Eq.(1)} & \text{otherwise} \end{cases} \quad (5)$$

Here, q is stochastically generated number between (0,1), $\arg \max \tau_{ij}(t)$ is the available adjacent city of city i who has the highest trail information, q_0 is a probability of selecting the optimal individual, for example, if $q_0=0.7$, the city who has the highest trail information will be chosen with a probability over 0.7.

3.2 Pheromone Updating Based on The Density Control Strategy

The biological density control mechanism uses the individual density in its colony. The higher individual density the ant has, i.e. the more approximate individuals this ant has, the lower diversity the colony has. Suppose $ant_i^t = (a_{i1}, a_{i2}, \dots, a_{iN})$ is the solution of i th ant at time t , where a_{ih} is the h th vertex that the i th ant chooses. We let $distance(ant_i^t, ant_j^t)$ be the distance between the i th ant and the j th ant, $density(i)$ denotes the the density factor of ant i at time t . Here dis_{max} is a constant determined by

the size of problem N . It was proved by large number of experimental results that $dis_{max} \in [0, N/5]$.

$$distance(ant_i^t, ant_j^t) = \sum_{h=1}^N (a_{ih} \oplus a_{jh}). \quad (6)$$

$$density(i) = \sum_{j=1}^m \varepsilon(i, j). \quad (7)$$

$$\text{here } \varepsilon(i, j) = \begin{cases} 1 & distance(ant_i^t, ant_j^t) < dis_{max} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

In the traditional ant colony algorithms, there are several methods to update the pheromone: increasing the pheromone on the tours of all the ants [1], or increasing the pheromone on the tours where the best ant passed, or updating the pheromone based on a certain rank [6], or increasing the pheromone on tours passed by the ants whose fitness are above the average fitness and decreasing the pheromone on other tours. All these methods only consider the fitness of solutions, but ignore their distribution. In our algorithm, we present an adaptive pheromone updating strategy based on density factor reflecting the quality and distribution of the solutions. Our strategy can adjust the pheromone on the tours dynamically, and avoid the pheromone being centralized on some individuals.

$$\Delta_{ik}^t = \begin{cases} \frac{Q \cdot density(i)}{length_i} & \text{if ant } i \text{ chooses city } k \text{ on current iteration} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Here $length_i$ is the length of the solution of the i th ant in the current iteration. The algorithm makes full use of the information of the solutions to update the pheromone. The density factor reflects the distribution of solution. Compared with other ants, the larger the ant's density is, the more similar its solution is. By the density factor, the tour of the ant with the high fitness and low-density solution will be greatly increased. This adaptive pheromone updating method not only can strengthen the pheromone on the tour of the best solution, but also can guarantee the diversity of solutions.

3.3 Density Controlled Mutation Operation

In line 2.2, the algorithm have mutation operation with the mutation probability of :

$$P_{mutation} = e^{-\left(1 - \frac{density(i)}{m}\right) \cdot \frac{f(i,t)}{sumf(t)}} \quad (10)$$

Which is adaptively defined according to the density and fitness value of the solutions, here $f(i,t)$ is the fitness of the i th ant at the t th iteration, $sumf(t)$ is the total fitness value of the whole colony at time t i.e. the t th iteration, and m is the total number of the ants in current iteration. It can be seen that the ants that have high individual density and low fitness value will have high mutation probability.

4 Experimental Results

The algorithm presented in this paper was applied to symmetric and asymmetric TSP benchmarks provided by TSPLIB. We also test these TSP benchmarks using the classical ant colony algorithm and neural network model to compare the performance with our algorithm. From our experience obtained by a large numbers of experiments [5], we let the parameters $\rho=0.4, Q=5, \alpha=1$ and $\beta=1$.

Table 1. The experimental results of symmetric TSP

Instance	Algorithm	Best solution	Average solution	Number	Average time(s)	allowed time(s)
d198	NNM	15780	15780.4	22	112.4	300
	TA	15780	15780.4	20	117.3	
	DACA	15780	15780.2	23	107.5	
att532	NNM	27686	27686.3	23	521.5	1000
	TA	27686	27686.3	22	533.0	
	DACA	27686	27686.2	24	472.6	

Table 2. The experimental results of asymmetric TSP

Instance	Algorithm	Best solution	Average solution	Number	Average time(s)	allowed time(s)
ry48p	NNM	14422	14424.7	20	76.4	250
	TA	14422	14425.3	19	71.1	
	DACA	14422	14422.0	25	35.9	
ftv170	NNM	2755	2755.1	25	114.9	500
	TA	2755	2755.0	25	109.2	
	DACA	2755	2755.0	25	103.4	

Here “number” is the number of trials obtaining the optimal solution “average solution” is the average route length of the solution obtained, “average time” is the average time required to find the best solution of the trials, “allowed time” is the longest time allowed to be executed in each trial. The numbers included in the name of the problems denote the numbers of cities for its corresponding problem.

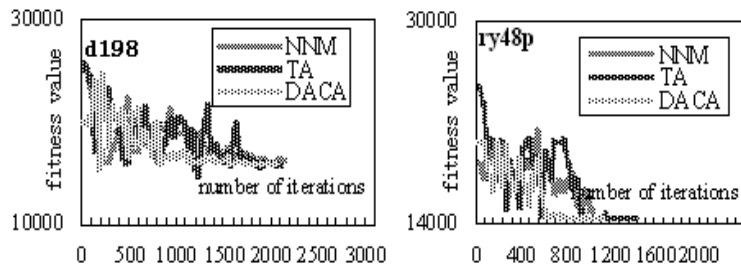


Fig. 1. It shows the evolutionary process of the best solution for d198 and ry48p using NNM, TA and DACA respectively. It is obvious that using DACA can not only get the same fitness valued solution, but also has a higher speed of reaching the best solution than NNM and TA.

From the tables we can see that the quality of our algorithm is much higher than that of NNM and TA, and it has much higher processing speed, stronger capability of finding optimal solutions, since DACA reaches the optimal solution in much less iterations.

Fig.1 gives the evolutionary process of the best solution for symmetric and asymmetric TSP problems. In DACA, after reaching the best solution, the curve fluctuates within a very narrow scope around the best solution, this confirms the conclusion that DACA has higher stability and is suitable for solving large scaled TSP problems.

5 Conclusion

In this paper we propose a new density controlled ant colony algorithm by adopting the density control strategy to guarantee the quality and diversity of the solutions. In this algorithm, the individuals are selected to have mutation operation according to the diversity and quality of the solutions. Experimental results on TSP benchmarks show that our algorithm has strong capability of optimization.

Nevertheless, the ant colony algorithm is still a newly emerging optimization algorithm, though it is developed rapidly and has demonstrated its superiority of solving complex optimization problems. The ant colony algorithm lacks of a solid mathematic foundation, its selection of the parameters usually depends on experiments and experiences. All these indicate that there are lots of open problems of ant colony optimization worth of deep research in both practical and theoretical area.

References

1. Dorigo, M., Maniezzo, V., Colomi, A.: Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*.26(1)(1996) 29-41
2. Tan, K.C., Huajin, T., Ge, S.S.: On parameter settings of Hopfield networks applied to traveling salesman problems. *Regular Papers, Vol.52. IEEE Transactions on Circuits and Systems*. 16(8)(2005)994 – 1002
3. Stützle, T., Hoos, H.H.: MAX-MIN Ant System. *Future Generation Computer Systems Journal*. 16(8)(2000)889-914
4. Xiao ping, L., Wei, W.: A new optimization method on immunogenetics. *Acta electronica Sinica*.31(1)(2003) 59-62
5. Ling, C., Ling, Q., Hong jian, C., Xiao hua, X.: An Ant Colony Algorithm With Characteristic of Sensation and Consciousness. *Journal of System Simulation*. 15(10)(2003)1418-1425
6. Bullnheimer, B., Hartl, R.F., Strauss, C.: A New Rank Based Version of the Ant System-A Computational Study. *Central European Journal for Operations Research and Economics*. 7(1)(1999) 25-38