Ant Colony Optimization Algorithm Model Based on the Continuous Space

Xuepeng Huang
Hubei University of Police, Wuhan, China

Abstract—Ant colony algorithm is a heuristic algorithm which is fit for solving complicated combination optimization. It showed great advantage on solving combinatorial optimization problem since it was proposed. The algorithm uses distributed parallel computing and positive feedback mechanism, and is easy to combine with other algorithms. This ant colony algorithm has already been widespread used in the field of discrete space optimization, however, is has been rarely used for continuous space optimization question. On the basis of basic ant colony algorithm principles and mathematical model, this paper proposes an ant colony algorithm for solving continuous space optimization question. Comparing with the ant colony algorithm, the new algorithm improves the algorithm in aspects of ant colony initialization, information density function, distribution algorithms, direction of ant colony motion, and so on. The new algorithm uses multiple optimization strategy, such as polynomial time reduction and branching factor, and improves the ant colony algorithm effectively.

Index Terms—ant colony algorithm; pheromone; continuous space optimization

I. INTRODUCTION

Using for aging behaviour of ant colony as inspiration, Italian scholar Marco Dorigo and Vittorio Maniezzo designed the first ACO algorithm: Ant System (AS), which is a highly innovative meta-heuristic algorithm. The researchers have tried various methods to improve the algorithm since AS, the first algorithm that conforms to ACO, was proposed. Since the ant colony algorithm has developed for many years, the researches of ant colony algorithm have been extended from single TSP field to a lot of fields of application. The ant colony algorithm has developed form one-dimensional static optimization to multidimensional dynamic combinatorial optimization, from research in discrete domain to research in continuous domains. The algorithm makes break through on implementation of the hardware, also, there are many significant results on model improving and integration of the algorithm model with bionics evolution algorithm or local search algorithms. The great advance of research of ant colony algorithm makes a widespread application, this paper will optimize traditional ant colony algorithm for characteristic of continuous spaces in aspects of ant colony initialization, information density function, distribution algorithms, ant colony direction of motion, and so on.This paper will improve the ant colony algorithm by using optimization strategy of polynomial time reduction, branching factor, etc.

II. DESCRIPTION OF TRADITIONAL ANT ALGORITHM

Assume that there are m ants, the characteristics of each ant are given as follows: it chose next city according to the distance between two cities and probability functions for whose variables the amount of the pheromone on the connected bound ($\tau_{ij}(t)$ is the pheromone intensity of connected bound $e(i,j)$ at moment t).The ants are provided to walk along the legitimate route. They can't turn to the city which they have visited except they have travelled all over the cities, and this is controlled by tabu list($tabu_k$ is tabu list for ant k and $tabu_k(s)$ is the element of the tabu list).

Assume that:

$B_i(t)(i=1,2,3\text{L},n)$ is the amount of the ants for city i at moment t;

$$m = \sum_{i=1}^{n} b_i(t)$$

is the amount of all ants, and for each ant, it has characteristics as follows:

it chose next city according to the distance between two cities and probability functions for whose variable is the amount of the pheromone on the connected bound;

it is provided to walk along the legitimate route and can't turn to the city which it has visited except it has travelled all over the cities, and this is controlled by tabu list;

the ant leaves pheromone on each connected bound that it has visited.

According to the assumption, the amounts of information for each path are equal to each other at initial time. Assume that:

$$\tau_{ij}(0) = C, \ C = \text{constant} \tag{1}$$

For ant $k(k=1,2,\text{L},m)$, it chose the turn direction according to amount of information when it is moving. Assume that $p_{ij}^k(t)$ is probability of that ant k turn from position i to position j at momentt, and it can obtain that:

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{is}^{\partial}(t)\cdot\eta_{is}^{\beta}(t)}{\sum \tau_{is}^{\partial}(t)\cdot\eta_{is}^{\beta}(t)}, & if \quad j\in allowed_k \\ \\ 0 & ,elseif \end{cases} \tag{2}$$

where $allowed_k = \{0,1,\text{L},n\} - tabu_k$ is the next city

chosen by ant k;

Different actual ant colony, the artificial ant colony has ability of memory, and $tabu_k(k=1,2,\mathrm{L},m)$ is used to record the visited city for ant k and $tabu_k$ is adjusted dynamically with the evolution process;

η_{ij} is the visibility of arc (i, j) and it can be calculated by using meta-heuristic algorithm. Usually, it is:

$$\eta_{ij} = 1/d_{ij} \tag{3}$$

where d_{ij} is the distance between city i and city j; ∂ is relative importance of the traces, β is relative importance of visibility, ρ is persistence of the traces and $1-\rho$ is information decay rate which means that the previous information will disappear gradually with time.

After time n, the ant finishes one cycle, and the amounts of information for each path are adjusted by using the following formula:

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \tag{4}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^k \tag{5}$$

where $\Delta\tau_{ij}^k$ is the amount of information of path ij left by ant k in this cycle, $\Delta\tau_{ij}$ is the information gain for path ij for this cycle, L_k is the path length for one travel cycle, and it can be obtained that:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L}, & \text{if ant k goes through path ij in this cycle} \\ 0, & \text{eles if} \end{cases} \tag{6}$$

III. ANT COLONY OPTIMIZATION ALGORITHM BASED ON THE CONTINUOUS SPACE

Definition 1: for a problem Π of finding minimum, if the feasible solution $s \in \tilde{S}$ that meet $f(s) \le \rho$ exist or not for given triplet (S, f, Ω) and parameter ρ?

This paper proposes useing polynomial time reduction to simplify the problem, i.e. increase the calculation efficiency by decreasing the accuracy of the optimal solution.

Construction algorithm is a method that add solution element into solutions iteratively from initially solution space until the solution was completed

Usually, one used greedy construction heuristic method to construct completed solution, i.e. in every construction procedure, one adds heuristic function to evaluate the solution, so that, one can obtain the solution which has maximum profit and uses this solution to construct the completed solution. The algorithm is given as follows:

Procedure Greedy Construction Heuristic

$s_p \leftarrow ChooseFirstComponent$

while(s_p is not a complete solution)do

$c \leftarrow GreedyComponent(s_p)$

$s_p \leftarrow s_p \otimes c$

end-while

$s \leftarrow s_p$

returns

end-procedure

where:

function Choose First Component chooses the first solution element c according to random generation or some constructional-heuristic-greedy chosen method.

Greedy Component returns the best solution element c;

Operator $\otimes$ means adding element c to partial solution s_p, and the whole process returns a completed solution.

Definition 2: neighbourhood construction is expressed by $N : S$ a 2^s, and for each $s \in S$ there is an corresponding set $N_s \subseteq S$. Set N_s becomes neighbourhood of s.

Usually, the optimal solution obtained by local search algorithmic local optimal solution and is not global optimal solution. Therefore, definition3 is given as:

Definition 3: a local optimal solution (local minimum) of a minimization problem s should meet:

$$\forall s' \in N(s) : f(s) \le f(s') \tag{7}$$

Similarly, a local optimal solution (local maximum) of a maximization problem s should meet:

$$\forall s' \in N(s) : f(s) \ge f(s') \tag{8}$$

Local search algorithm also needs to definite a neighbourhood cheek used to decide how to search neighbourhood and which adjacent solution can be accepted. Usually, the rule for accepting adjacent solution is best-improvement rule or first-improvement rule.

In the system, ant k in city i chooses city j as next city according pseudorandom proportional rule, i.e.:

$$j = \begin{cases} \arg\max\{\tau_{ij}[\eta_{il}]^\beta\}, & \text{if } q \le q_0 \\ J, & \text{else if} \end{cases} \tag{9}$$

Where q is a random which follows uniform distribution in [0,1], $q_0(0 \le q_0 \le 1)$ is a parameter and J is a random generated by probability distribution

By adjusting parameter q_0, one can determine to search the surroundings of the optimal path so far or to search other path.

For pheromone, the update rule of global pheromone is given by:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs}, \forall (i,j) \in T^{bs} \tag{10}$$

where $\Delta\tau_{ij}^{bs} = 1/C^{bs}$.

For local pheromone, the update rule is given by:

$$\tau_{ij} \leftarrow (1-\xi)\tau_{ij} + \xi\tau_0 \tag{12}$$

Where ξ and τ_0 are two parameters, ξ meets $(0 \le \xi \le 1)$ and τ_0 is the initial value of pheromone.

In addition, there is a recessive limitation for pheromone, i.e.:

$$\forall(i,j):\tau_0 \le \tau_{ij} \le 1/C^{bs} \tag{13}$$

The long-term effect of pheromone can reduce the scale of search space with good parameters. However, if the effect is too strong, the algorithm will be in the state of stagnates rapidly. So, one must take some methods to evaluate the algorithm for checking whether the algorithm is in the state of stagnates or not.

So, this paper used $\lambda-$ branching factor $(0< \lambda <1)$ which can calculate the distribution of pheromone more directly. For given city i, $\lambda-$ can be defined as:

Definition 4: $\lambda-$ is the amount of boundaries which is related to node i and meets $\tau_{ij} \ge \tau_{min}^i + \lambda(\tau_{max}^i - \tau_{min}^i)$, where τ_{max}^i and τ_{min}^i are maximum and minimum pheromone of city I, respectively, $\lambda\in[0,1]$, $\lambda-\in[2,n-1]$, and n is the number of nodes.

On the continuous space, the optimization process of ant colony algorithm includes: initialization of assign ant colony, determination of information distribution function, analysis of distribution algorithm and motion direction of ant colony. The algorithm is given by:

Firstly, the amount of ant colony N should be determined according to the problem. The problem domain should be divided to N sub domain, so, ant $i,i\in[1,N]$ is placed in the N subdomain. For each ant, the motion distance is given by:

$$X_{iR} = Start + iD_{iR} \tag{14}$$

Therefore, the change of overlap degree between the motion time interval and neighbouring sub domain is Δn when a single ant moves Δx. In neighbouring time interval:

$$\Delta n = \frac{\Delta x}{\Delta D_{MRL}} = \frac{\Delta x}{\Delta D_{RL}} \tag{15}$$

Then, one can determine the information distribution of current ant colony according the quality of ant colony in solution space, further more, one can determine the peak value of the remaining information distribution function and obtain corresponding distribution function.

IV. SIMULATION STUDY

In the initialization, each pheromone is evaluated an initial value τ_0. In the iteration process of the algorithm, the ants are distributed to each nodes according to the algorithm rule, and for each node, the probability that it is chosen as next node is given by:

$$P_{\mathrm{T}}(c_{h+1} = j|x_h) = \begin{cases} \dfrac{F_{ij}(\tau_{ij})}{\sum\limits_{(i,l)\in N_i^k} F_{il}(\tau_{il})}, & if(\mathrm{i},\ \mathrm{j})\in N_i^k \\ 0, & else\ if \end{cases} \tag{16}$$

Forderivation, the probability above can be written as:

$$P_{\mathrm{T}}(c_{h+1} = j|x_h) = \begin{cases} \dfrac{\tau_{ij}^a}{\sum\limits_{(i,l)\in N_i^k} \tau_{il}^a}, & if(\mathrm{i},\ \mathrm{j})\in N_i^k \\ 0, & else\ if \end{cases} \tag{17}$$

(1) Value convergence

After a random iteration, the maximum increase value of pheromone of bound $(\mathrm{i},\ \mathrm{j})$ is $q_f(s^*)$. So that, the maximum increase value of pheromone in first iteration is given by:

$$(1-\rho)\tau_0 + q_f(s^*) \tag{18}$$

In the second iteration, the maximum is:

$$(1-\rho)^2\tau_0 + (1-\rho)q_f(s^*) + q_f(s^*) \tag{19}$$

Similarly, due to the evaporation of the pheromone, the pheromone value in cycle θ should not larger than:

$$\tau_{ij}^{max}(\theta) = (1-\rho)^\theta\tau_0 + \sum_{i=0}^{\theta} (1-\rho)^{\theta-i}q_f(s^*) \tag{20}$$

Due to $0<\rho<1$, the summation of above equation advance gradually converge:

$$\lim_{\theta\to\infty}\tau_{ij}(\theta) \le \tau^{max} = \frac{q_f(s^*)}{\rho} \tag{21}$$

So that, there exist value boundaryfor pheromone as τ_{min},τ_{max} to ensure that $p_{min}>0$, where p_{min} is the probability of any feasible chose for any past decomposition x_h.

An ordinary floor probability of p_{min} is given by:

$$p_{min} \ge \hat{p}_{min} = \frac{\tau_{min}^a}{(N_c-1)\tau_{max}^a + \tau_{min}^a} \tag{22}$$

where N_c is potential of set C.

So, the probability of obtaining a general solution s', including any best solution $s^*\in S^*$, where $n\in+\infty$ is the maximum length of sequence, is $\hat{p} \ge \hat{p}_{min}^n > 0$. $\hat{P}^*(\theta)$ has a floor as:

$$\hat{P}^*(\theta) = 1-(1-\hat{p})^\theta \tag{23}$$

It can obtain a probability larger than any $1-\varepsilon$ when θ is large enough.So that:

$$\lim_{\theta\to\infty}\hat{P}^*(\theta) = 1$$

(2) Solution convergence

Set event E_θ express that the algorithm first obtain the optimal solution in iteration θ. Event $\bigwedge\limits_{\theta=1}^{\infty}\neg E_\theta$ express that the algorithm does not obtain any optimal solution s^*. So that, $P\left(\bigwedge\limits_{\theta=1}^{\infty}\neg E_\theta\right)$ has a up boundary as:

$$P\left(\bigwedge_{\theta=1}^{\infty}\neg E_{\theta}\right)\leq \text{P}(s^* \text{isnevertraversed})$$

In an ordinary iteration θ, any optional chose should meet the following probability limitation:

$$p_{\min} \geq \hat{p}_{\min}(\theta)$$

$$= \frac{\tau_{\min}^{a}(\theta)}{(N_c-1)\tau_{\max}^{a} + \tau_{\min}^{a}(\theta)}$$

$$\geq \frac{\tau_{\min}^{a}(\theta)}{N_c\tau_{\max}^{a}} = \hat{p}_{\min}'^{\theta}$$

(24)

The probability floor of optimal solution s^* is $\hat{p}(\theta) \geq (\hat{p}'(\theta))^n$, where is the maximum length of sequence. This floor is independent on the event occurs before iteration θ. So that, $P\left(\bigwedge_{\theta=1}^{\infty}\neg E_{\theta}\right)\leq$ P(s^* is never traversed) has a floor as:

$$\text{P}(s^* \text{ is never traversed}) \leq \prod_{\theta=1}^{\infty}(1-(\hat{p}_{\min}'(\theta))^n)$$

$$=\leq \prod_{\theta=1}^{\infty}(1-(\frac{\tau_{\min}^{a}(\theta)}{Nc\tau_{\max}^{a}})^n)$$

If $\tau_{\min}(\theta) = d/\ln(\theta+1)$, it can obtain that:

$$\sum_{\theta=1}^{\infty}\ln(1-(\frac{\tau_{\min}^{a}(\theta)}{Nc\tau_{\max}^{a}})^n) = \sum_{\theta=1}^{\infty}\ln\left[1-\left(\frac{(d/\ln(\theta+1))^a}{Nc\tau_{\max}^{a}}\right)^n\right]$$

$$= \sum_{\theta=1}^{\infty}\ln\left[1-\left(\frac{d}{\ln(\theta+1)^a}\right)^n\right]$$

$$\leq -d_1^n\sum_{\theta=1}^{\infty}\ln\left[\left(\frac{1}{\ln(\theta+1)^a}\right)^n\right] = -\infty$$

(25)

$P^*(\theta)$ is the probability that it obtained optimal solution in the preceding θ iteration, and it must meet:

$$\lim_{\theta\to\infty}P^*(\theta) = 1$$

Similarly, θ^* is the iteration for obtaining first optimal solution, $P(s^*,\theta,k)$ is probability that it obtained optimal solution in the θth iteration of ant k, and it must meet:

$$\lim_{\theta\to\infty}P(s^*,\theta,k) = 1$$

Simulation test:

Comparing algorithm proposed this paper to ACO algorithm shows that the optimal algorithm is better than ACO algorithm on average and standard deviation of object function, as shown in Table 1 and Figure 1-3.

TABLE I.
COMPARISON OF TEST DATA

Algorithm type	Traditional algorithm	Improved algorithm
Optimal target	1.3652	1.3652
Colony size	10	10
Search cycle	20	20
The number of operations	1000	1000
Min J Mean	1.4403	1.4095
Lupin performance	1.4095	3.2449
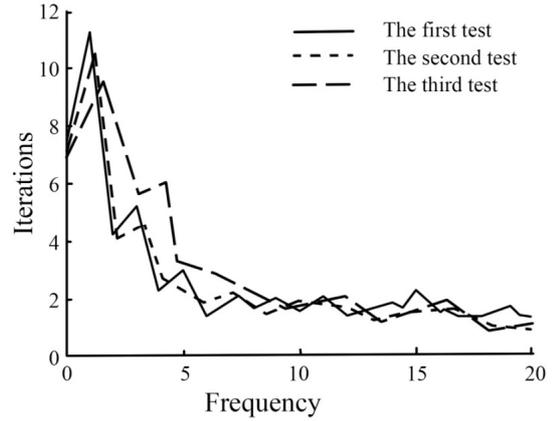
Figure 1.   Iteration process of object function
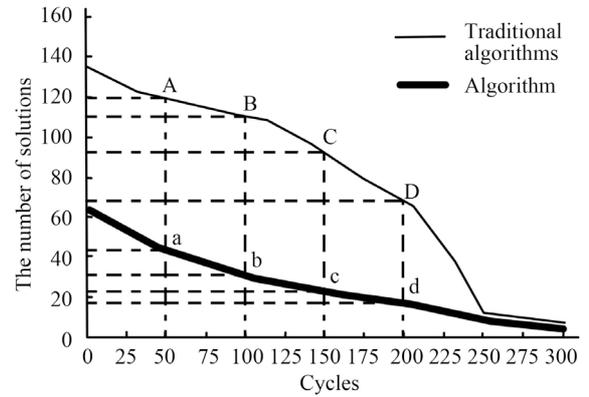
Figure 2.   Comparison of convergence speed
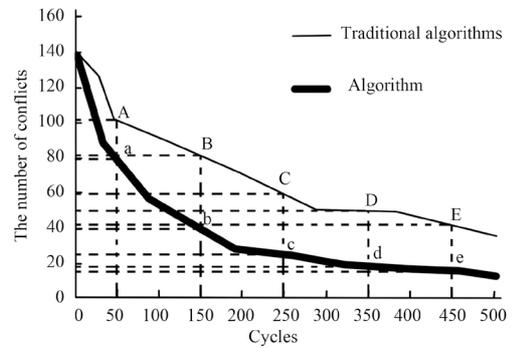
Figure 3.   Comparison of conflict

V. CONCLUSION

An ant colony algorithm for solving continuous space optimization question is proposed in this paper. The value convergence and solution convergence of the algorithm is confirmed by theory method. Simulation test shows that the optimal algorithm is better than ACO algorithm on average and standard deviation of object function.

REFERENCES

[1] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies[C]. Proc of European Conf on Artificial Life. Paris: Elsevier Press, 1991: 134-142.

[2] Colorni A, Dorigo M, Maniezzo V. An investigation of some properties of an "Ant algorithm"[C]. Proc of the Parallel Problem Solving From Nature Conference. Brussels: Elsevier Press, 1992: 509-520.

[3] Dorigo M, Maniezzo V, Colorni A. The ant system: Optimization by a colony of cooperating agents[J]. IEEE Trans on Systems, Man and Cybernetics, 1996, 26(1): 29-41. https://doi.org/10.1109/3477.484436

[4] Dorigo M, Gambardella L M. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. IEEE Trans on Evolutionary Computation, 1997, 1(1): 53-56. https://doi.org/10.1109/4235.585892

[5] XG Yue, S K Boddhu, Y Lu, F Xiao, T Memonen, M V Cañiv. Gas Outburst Prediction Based on OD Algorithm, Recent Patents on Computer Science, 2016, 9 (1): 25-39. https://doi.org/10.2174/2213275908666150218194450

[6] XG Yue, SL Zhao, GF Ren, A New Algorithm of Sensitivity Analysis Based on Neural Network for Safety Engineering, Journal of Computational and Theoretical Nanoscience, 2015, 12 (11): 4111-4113.

[7] XG Yue, G Zhang, Q Wu, F Li, XF Chen, GF Ren, M Li. Wearing prediction of stellite alloys based on opposite degree algorithm, Rare Metals, 2015, 34 (2): 125-132. https://doi.org/10.1007/s12598-014-0430-0

[8] Stützle T, Gambardella M. Distributed optimization by ant colonies[C]. Proc of the Int Conf on Artificial Neural Networks and Genetic Algorithms. Norwich: Czech Press, 1997: 245-249.

[9] Eberhart R C, Kennedy J. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. Nagoya: IEEE Press, 1995: 39-43.

[10] Kennedy J, Eberhart R C. A new optimizer using particles swarm theory[C]. Proc of 6th Int Symposium on Micro Machine and Human Science. Perth: IEEE Press, 1995: 1942-1948.

[11] Min K X, Ge H W, Zhang Y, et al. Solving traveling salesman problems by an ACO-and-PSO-Based hybrid algorithm[J]. J of Ji-lin University: Information Science Edition, 2006, 24(4): 402-405.)

[12] Chai B J, Liu D W. Application of an ant colony algorithm in TSP based on particle swarm[J]. Computer Simulation, 2009, 26(8): 89-91.

[13] Xia H, Wang H, Chen X. A kind of ant colony parameter adaptive optimization algorithm based on particle swarm optimization thought[J]. J of Shandong University: Engineering Science, 2010, 40(3): 26-30.

[14] Yang Y N. The parameter optimization of ant colony algorithm and its application[D]. Nanjing: School of Computer Science and Engineering, Nanjing University of Science and Technology, 2008.)

[15] Bonabeau E, Yang J B. Inspiration for optimization from social insect behavior [J]. Nature (S0028-0836), 2000, 406(6): 39-42. https://doi.org/10.1038/35017500

AUTHOR

X. P. Huang is with the Information & Network Center, Hubei University of Police, Wuhan , CO 430034 P.R. China (e-mail: 42482852@ qq.com).