Apply the Ant Feature of Sensation to Calculate the Minimum Value of Function

Chao-Yang Pang
Group of Gene Computation, Key Lab. of Visual Computation and Virtual Reality of Sichuan Province, Chengdu 610066, China;

College of Mathematics and Software Science, Sichuan Normal University, Chengdu 610066, China

Qiong Yang
Information & E-Education Department, Sichuan Tourism School, Chengdu 610041, China

Yan Zhang
College of Computer and Software, Shenzhen University Shenzhen, Shenzheng, 518060, China

Ben-Qiong Hu
College of Information Management, Chengdu University of Technology, Chengdu 610059, China

(corresponding author: hbq402@126.com)

Abstract—**Calculating the minimum (or maximum) value of functions is an important problem in optimization field. Applying the method of ant colony optimization (ACO) to solve the problem is an interesting research topic currently, and the main disadvantage is that solution is local optimal. To evade this disadvantage in some degree, in this paper, the ant feature of sensation is used. Experiment shows that the method presented in this paper generates high quality of solution.**

Keywords- Feature of Ant Sensation; ACO; Minimum Value

I. INTRODUCTION

Ant Colony Optimization (ACO) is initially proposed by Colorni, Dorigo and Maniezzo, its underlying idea inspired by the behavior of real ants [1]. ACO is a class of algorithms to solve optimization problems, such as Travelling Salesman Problem (TSP) [2], Quadratic Assignment Problem (QAP) [3], Job Scheduling Problem (JSP) [4], Vehicle Route Problem [5], and other problems [13-15]. For discrete problem, such as TSP, QAP and JSP, ACO has showed its high performance. Currently applying it to solve continuous problem becomes an interesting topic [6-8], such as calculating the minimum value of function. And in this application, the main disadvantage of ACO appears that easily its solution traps into local optimum, and currently there no good method to overcome this disadvantage. On the other hand, ant holds some feature of sensation. For example, if two paths deposit too much pheromone, even the difference of their amount of pheromone is clear, the ant can't still distinguish the two paths clearly by pheromone. The motivation of this paper is that applying the feature of ant sensation to overcome this disadvantage in some degree.

II. FEATURE OF ANT SENSATION

Ant has the following four sensation features [9, 11]:

1. Only when the quantity of pheromone reaches a threshold, the pheromone can be felt by the ant. And this threshold is called **absolute sensor threshold** (AST).
2. The change of pheromone quantity can be felt by ant, but the value of change must reach a threshold. And this threshold is called **contrast sensor threshold**, (CST).
3. **Weber Theorem**: Suppose I denote the pheromone quantity, ΔI is CST. If pheromone quantity is changed, we have
$$\frac{\Delta I}{I} = K,$$
where K is a constant number.
4. When quantity of pheromone on both edge reaches a threshold *IT*, the pheromone is too dense and the ant can not disguish the two different edges clearly.

Chen firstly applies the ant features to improve the solution quality of discrete problem [11]. And there is no report currently applying ant feature to improve solution quanlity of continus problem. The motivation of this paper is to develop a rule and apply ant feature to solve continus problem.

III. APPLY ANT FEATURE TO SOVLE CONTINUOUS PROBLEM

A. Initialization

In this paper, calculating the minimal value of function $y = f(x)$ ($x \in [a,b]$) will be illustrated, and the following method can be applied to other functions.

Firstly, divide interval $[a,b]$ into m small subintervals with same size. Set a constant number $const$ as the initial pheromone quantity of each small interval, denote by $\tau_i(0) = const$.

Secondly, put one ant at the center of each interval. Suppose the m ants are $a_1, a_2, \cdots a_m$, the center of the *i-th*

978-0-7695-4281-2/10 $26.00 © 2010 IEEE
DOI 10.1109/ICGEC.2010.96

interval is x_i and has function value $f(x_i)$. Let $Y^* = \min_i f(x_i)$.

Thirdly, calculate AST:

$$AST = C \cdot (\max_i f(x_i) - \min_i f(x_i)),$$

where C is a constant number.

The fourthly, develop a rule for ant's moving:

Suppose ant a_k is staying at the center of the *i-th* interval. Let

$\eta_{ij} = f(x_i) - f(x_j)$, $(i, j = 1 \cdots m)$, where j denotes the *j-th* interval .

The bigger η_{ij} is, the more possible the ant move to *j-th* interval. Denote the set of all intervals which ant a_k can move to by $allowed_k$. If $\eta_{ij} > 0$, then $j \in allowed_k$, otherwise $j \notin allowed_k$.

B. Ant Search New Interval to Move

Suppose the *j-th* interval ($j \in allowed_k$) will be considered as next candidate of traveled interval by ant a_k. Denote the *t-th* iteration of ACO by *t* and the quantity of pheromone on the the *j-th* interval is denoted by $\tau_j(t-1)$.

If $\tau_j(t-1) < AST$ for all $j \in allowed_k$, every interval has no enough pheromone to attract ant a_k, it selects a interval randomly to travel. Otherwise, do the following process:

Calculate CST of the j-th interval according to Weber Theorem

$$CST_j = \left[\tau_j(t-1)\right] \cdot K,$$

where *K* is set to 1/50 in this paper.

If $\tau_j(t-1) - \tau_j(t-2) \le CST_j$, the change of pheromone on the the *j-th* interval is not enough, the ant selects its candidate of traveled interval by the previous pheromone felt by it. Otherwise, it depends on current pheromone to select.

Let

$$\theta_{ij}^{(k)}(t) = \begin{cases} \left[\tau_j(t-1)\right]\eta_{ij}(t), & if \left|\tau_j(t-1) - \tau_j(t-2)\right| \le CST \\ \tau_j(t)\eta_{ij}(t), & Otherwise \end{cases},$$

Ant a_k moves to the *j-th* interval from the *i-th* interval according transition probability $p_{ij}^{(k)}(t)$:

$$p_{ij}^{(k)}(t) = \begin{cases} \dfrac{\theta_{ij}^{(k)}(t)}{\sum_{h \in allowed_k} \theta_{ih}^{(k)}(t)}, & j \in allowed_k \\ 0, & \end{cases}$$

When the quantity of pheromone goes up in excess of a threshold *IT* (i.e., $\tau_j(t-1) > IT$), the pheromone is too dense, and pheromone change on this interval can not be felt by the ant. In addition, if $\tau_j(t-1) > IT$, the pheromone will induce solution into local optimization possibly. Therefore, the transition probability rule should be changed to fit this situation:

Firstly, calculate threshold IT [11]

$$IT_j = h \cdot t_{\max} \cdot m \cdot AST,$$

where h is a constant number, $0 < h < 1$ and $t_{\max}$ is the maximum number of iteration.

Secondly, give a probability, let ant select its traveled interval which has maxmum pheromone randomly. If its selection fails, use common used rule to select its traveled interval. According to this discussion, the rule in ref.[11-12] is used and modified as below:

$$j = \begin{cases} \arg\max\left[\tau_j(t) \middle| j \in allowed_k\right] & if \quad q \le q_0 \\ s_{ij}^{(k)}(t) & Otherwise \end{cases}$$

, where q_0 is a constant number and q is random [12], and

$$s_{ij}^{(k)}(t) = \begin{cases} \dfrac{\left[\tau_j(t)\right]^\alpha \left[\eta_{ij}(t)\right]^\beta}{\sum_{s \in allowed_k} \left[\tau_s(t)\right]^\alpha \left[\eta_{is}(t)\right]^\beta}, & j \in allowed_k \\ 0 & Otherwise \end{cases}$$

C. Adjust The Position of Ant in Samll Interval

According to above rule, the ant will select a small interval to travel. After it reaches the new interval, it will stay at the center of this interval. However, the function value at the center is not small possibly, and there are positions near the center which have smaller function value. So, the ant can randomly search a new position to replace its old position. This adjusting is necessary because it can evade the error accumulation at each iteration step.

D. Update Pheromone at Every Small Interval

The rule is same with ACO.

IV. SIMULATION

Let

$f(x) = 5e^{-0.5x}\sin(30x) + e^{0.2x}\sin(20x) + 6$, where $x \in [0,8]$.

The aim here is to calculate minimum value of this function. The function is a typical test function, which at $x = 0.57254$ has minimum value 1.2573 [6-8].

The test environment listed as below:

Notebook TOSHIBA Satellite M200, CPU 1.73GHz and Mem. 502M.

Parameters:

$\alpha=1,\ \beta=10,\ \rho=0.7,\ C=0.4, h=0.5,\ Q=0.5$, $m=10$, $t_{\max}=20$

The function is shown at Fig.1 and test result is at Table 1. The authors of this paper do 1000 times of tests, and get the following performance of the method presented in this paper:

Compared with two excellent algorithms currently [6-8], the solution accuracy is high and runtime is short (average time 8.902531s). And the robustness is same (robustness: every time get nearly same solution is called robust.)

The authors of this paper have done many tests, get the same conclusion (see the appendix of this paper).

V. CONCLUSION

Calculating the minimal value of function is an important problem of the field applied mathematics. Applying ant colony optimization (ACO) to process this problem is interesting, and one defect of this application is that solution accuracy is not high. To improve the accuracy, in this paper, the feature of ant sensation is embedded into ACO method to develop a hybrid algorithm. Simulation shows that the hybrid algorithm generates high solution quality.

ACKNOWLEDGMENT

The work is supported by the fund of key developing program of technology innovation from Education Dept. of Sichuan Province (No. 09zz028) and the open fund of key lab. of modern communication and information process of Shenzheng city . The authors appreciate the help from Prof. J. Zhang and Prof. J. Zhou. The authors appreciate the help from the other members of gene computation team: W. Hu, C.-B. Wang, X. Li, H. Liu, G. Jiang, S.-P. Wang, L.-J. Ye, J.-L. Zhou, P. Shuai, Y. Li, Z.-C. Dan, M.-X. Chen, D.-X. Zhang, L. Yang, and X.-P. Xia.

REFERENCES

[1] A. Colorni, M. Dorigo, V. Maniezzo, "Distributed optimization by ant colonies," Proc. European Conference on Artificial Life(ECAL91), PARIS, FRANCE, ELSEVIER PUBLISHING, 1991, pp. 134-142.

[2] M. Dorigo, V. Maniezzo and A.Colorni, " Ant system: optimization by a colony cooperating agents, " IEEE Trans. On System, Man, and Cyberetics Part B: Cybernetics, vol. 26 , Jan., 1996, pp. 29-41.

[3] V. Maniezzo, A. Colorni , " The ant system applied to the quadratic assignment problem," IEEE Transactions on Knowledge and Data Engineering, vol. 1, May, 1999, pp. 769-778.

[4] A. Colorni, M. Dorigo , V. Maniezo, " Ant system for job-shop scheduling, " Belgian J Oper Res Stat Comp Sci, vol. 34, 1994, pp.39-53.

[5] B. Bullnheimer, R. Hartl, C. Strauss, " An improved ant system algorithm for the vehicle routing problem, " Annals of Operations Research, vol.89, 1999, pp. 319- 328.

[6] Y.-I. Li, T.-J Wu, " An Adaptive and Colony System Algorithm for Continuous-Space Optimization Problems, " Pattern Recognition and Artificial Intelligence , vol.14, April, 2001, pp. 423-427.

[7] W.-Q. Xiong, S.-H. Yu, P. Wei, " A Design Based in Ant Colony Algorithm for Function Optimization," Microelectronics & Computer, vol.20, Jan., 2003, pp.23-25.

[8] H.-B. Duan, G.-J. Ma, D.-B. Wang, X.-F. Yu, " Improved Ant Colony Algorithm for Solving Continuous Space Optimization Problems, " Journal of System Simulation, vol.19, May, 2007, pp.974-977.

[9] L.-S. Chen, J.-X. Ma, New Psychology. Beijing Normal University Press, Beijing, 2002

[10] H.-B. Duan, Ant Colony Algorithms Theory and Applications . Science Press, Beijing, 2005.

[11] L. Chen, L. Qin, J.-H. Chen, and X.-H. Xu, " Ant Colony Algorithm with Characteristics of Sensation and Consciousness," Journal of System Simulation,vol.15, Oct., 2003, pp. 1418-1245,

[12] L. Gambardella, M. Dorigo, " Ant-Q: a reinforcement learning approach to the traveling salesman problem". Proc. of the 12th International Conference on Machine Learning, California, USA , 1995, pp. 252-260.

[13] Hameed Al-Qaheri, Abhijit Mustafi, and Soumya Banerjee, "Digital Watermarking using Ant Colony Optimization in Fractional Fourier Domain," Journal of Information Hiding and Multimedia Signal Processing, Vol. 1, July, 2010, pp. 179-189.

[14] S. C. Chu, John F. Roddick and J. S. Pan, "Ant Colony System with Communication Strategies", Information Sciences, Vol. 167, Issues 1-4, 2004, pp. 63-76.

[15] Shu-Chuan Chu, John F. Roddick, Che-Jen Su and Jeng-Shyang Pan, "Constrained Ant Colony Optimization for Data Clustering", 8th Pacific Rim International Conference on Artificial Intelligence, LNAI 3157, 2004 , pp. 534-543.

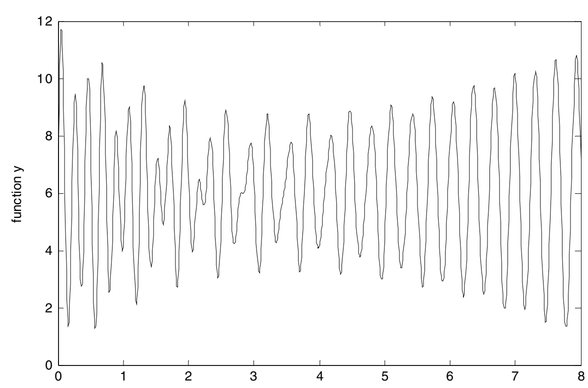

Figure 1. Function *f(x)*

TABLE I. PERFORMANCE COMPARISON OF THREE ALGORITHMS

Algorithm	Minimum	Average	Robust
Ref. [6]	f'=1.3652	1.4403	5.5011%
Ref. [8]	f'=1.3652	1.4095	3.2449%
This Paper	f'=1.2573	1.3147	4.5653%

a. T he true minimal value of the function is 1.2573.
b. 1000 times of tests are done for each algorithm
c. The average runtime of the method in this paper is 8.902531s.
d. Robustness is defined as $\left|\left(average - f'\right)/f'\right| \times 100\%$

APPENDIX

The minimum function values of following functions are calculated and are listed in table 2.

$f_1(x) = \sin(x)$, $x \in [0, 2\pi]$; $f_2(x) = (x+1)^2 + 1$, $x \in [-1, 1]$;

$f_3(x) = (x+1)(x+2)(x+3)(x+4)(x+5) + 5$, $x \in [-5, 0]$;

$f_4(x) = (x+2)\cos(9x) + \sin(7x)$, $x \in [0, 4]$;

$f_5(x) = 5x^6 - 36x^5 + 82.5x^4 - 60x^3 + 36$, $x \in [0, 3.5]$;

$f_6(x) = x\sin(10\pi x) + 2$, $x \in [-1, 2]$; $f_7(x) = -3x^2 e^{-x}$, $x \in [0, 3]$;

$f_8(x) = e^{-0.1x}\sin(5x+1) + 2$, $x \in [0, 8]$; $f_9(x) = \sin(10x^2)\cos(2.5x+1) + 10$, $x \in [0, 2]$

$f_{10}(x) = \ln(x+1)\cos(x^2)$, $x \in [1, 5]$; $f_{11}(x) = 20(x^2 - 10)\sin(5x)$, $x \in [0, 5]$;

$f_{12}(x) = 2x\cos(x^2 + 1) + 1$, $x \in [0, 5]$;

TABLE II. THE MINIMUM FUNCTION VALUE CALCULATED BY THE METHOD PRESENTED IN THIS PAPER

	True $(x_0, f(x_0))$	Best $(x'_0, f(x'_0))$	Average $(\bar{x}, \bar{y})$	Error(%) (e)	Robust(%) (R)
$f_1(x)$	4.7123890 -1	4.7123890 -1	4.7123890 -1	0	0
$f_2(x)$	-1 1	-0.9999999 1	-0.9977546 1.0000114	0.00114	0.00114
$f_3(x)$	-1.3555671 1.3685678	-1.3555671 1.3685678	-1.3557603 1.3689232	0.02597	0.02597
$f_4(x)$	2.44888 -5.4342747	2.4488850 -5.4342746	2.4516879 -5.4294614	0.08857	0.08857
$f_5(x)$	3 -4.5	3 -4.5	2.9997861 -4.4976480	0.05227	0.05227
$f_6(x)$	1．95052 0.04974027	1.9505202 0.0497403	1.9503416 0.0506953	1.92	1.91997
$f_7(x)$	2.0000004 -1.6240234	2.0000004 -1.6240234	1.9999883 -1.6240212	1.36e-004	1.36e-004
$f_8(x)$	0.7384806035 1.0713726891	0.7384785251 1.0713726891	0.7386819417 1.0727065898	0.0012	0.0012
$f_9(x)$	0.88564851186 9.0026814281	0.88564937194 9.0026814283	0.8856594350 9.0029180638	2.63e-005	2.63e-005
$f_{10}(x)$	4.69061469471 -1.73871840686	4.69063320224 -1.73871839715	4.6907230249 -1.737348764	7.88e-004	7.88e-004
$f_{11}(x)$	4.74253155872 -247.00008821	4.74252577955 -247.00008812	4.7431862918 -246.95756121	1.72e-004	1.72e-004
$f_{12}(x)$	4.58420358205 -8.1658163005	4.58420718596 -8.1658162998	4.5833556398 -8.1561878909	0.00118	0.00118

e. 1000 times of tests are done for every function.

f. The first column data is calculated by the method in MatLab, the 2nd column is the best solution of 1000 times of tests, and the 3rd column is the average of 1000 times.

g. Error is defined as $e = \frac{|\bar{y} - f(x_0)|}{|f(x_0)|} \times 100\%$, robustness is defined as $R = \frac{|\bar{y} - f(x'_0)|}{|f(x'_0)|} \times 100\%$.

h. The runtimes for the 12 functions respectively are (second) 7.043813, 7.438375, 8.280219, 7.991266, 8.663391, 8.142, 7.854922, 8.009766, 8.165343, 8.433985, 7.978656, 7.817375.