International Journal of Computer Applications (0975 – 8887)
Volume 74– No.13, July 2013

Modified Ant Colony Optimization Algorithm with Uniform Mutation using Self-Adaptive Approach

Ramlakhan Singh Jadon
Maharana Pratap College of Technology
Gwalior, India

Unmukh Dutta
Maharana Pratap College of Technology
Gwalior, India

ABSTRACT
Ant Colony Optimization (ACO) algorithm is a novel meta-heuristic algorithm that has been widely used for different combinational optimization problem and inspired by the foraging behavior of real ant colonies. Ant Colony Optimization has strong robustness and easy to combine with other methods in optimization. In this paper, an efficient ant colony optimization algorithm with uniform mutation operator using self-adaptive approach has been proposed. Here mutation operator is used for enhancing the algorithm escape from local optima. The algorithm converges to the optimal final solution, by gathering the most effective sub-solutions. Experimental results show that the proposed algorithm is better than the algorithm previously proposed.

General terms
Algorithm, Experimentation, verification

Keywords
Ant Colony optimization, ACO, Mutation operator

1. INTRODUCTION

The problem of optimization is the most crucial problem in today's era and a great work has been done to solve it. During last few years, many optimization algorithms, like Ant Colony Optimization (ACO), Particle Optimization Problem (PSO), Artificial Bee Colony Algorithm (ABC), Differential Evolution (DE), Genetic algorithm (GA) etc., has been proposed.
Ant Colony Optimization (ACO)[1] algorithm was proposed firstly in 1991 by Dorigo M. and was designed to simulate the foraging behavior of real ant colonies. ACO algorithms [6, 8, 9, 10] have been widely used for solving different combinational optimization problems such as Job-Scheduling Problem, Traveling Salesman Problem, and Vehicle Routing Problem etc. Various enhanced versions of the original ACO algorithms have been done over the years. For improving the quality of final solution and speedup of the algorithm, various strategies like dynamic control of solution construction[4], mergence of local search[3], partition of artificial ants into two groups: common ants and scout ants[11], strategies for updating new pheromone[7] and using strategies of candidate lists[2] are studied.
The principle of the phenomenon is shown in Fig (a), Fig (b), Fig(c) and Fig (d).

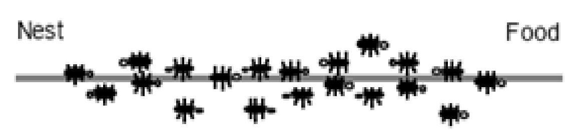

Fig (a) Real ants follow a path between nest and food source

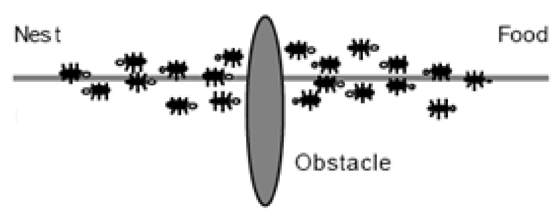

Fig (b) An obstacle appears on the path: ants choose whether to turn left or right with equal probability

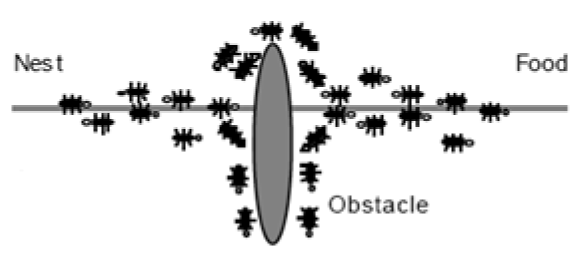

Fig (c) Pheromone is deposited more quickly on the shorter path

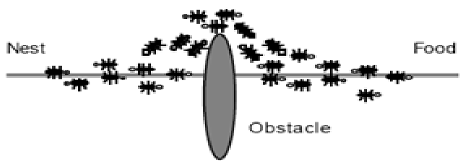

Fig (d) all ants have chosen the shorter path

The remainders of this paper are organized as follows. Section 2 introduces mathematical model of ant colony algorithm. Section 3explains self-adaptive approach. Section 4 explains benchmark functions and section 5 describes the proposed algorithm. Finally, in Section 6, experiments are conducted on benchmark functions, section 7 concludes the paper, section 8 and 9 acknowledge the paper and provide references used in this paper respectively.

2. MATHMATICAL MODEL OF ANT COLONY ALGORITHM

Several ACO algorithms have been proposedin the literature. The original ant colony optimizationalgorithm is known as Ant System and was proposed in the early 1990s. Here the transition probability from location i to location j for the k-th ant as follow:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\left[\tau_{ij}[(t)]\right]^{\alpha}\left[n_{ij}\right]^{\beta}}{\sum_{k \in \text{allowed}_k}\left[\tau_{ij}[(t)]\right]^{\alpha}\left[n_{ij}\right]^{\beta}} & \text{if } j \in \text{allowed} \\ 0 & \text{otherwise} \end{cases}$$

After the ants completed their tours, the pheromone trail values are updated according to following formula:

$$\tau_{ij}\,(t+n) = \rho.\tau_{ij}\,(t) + \Delta\tau_{ij}$$

Where ρ = local pheromone decay parameter and ρ ε (0, 1) then 1- ρ represents the evaporation of trail between time t and t+n,

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

Where$\Delta\tau_{ij}$ = quantity of per unit length of pheromone trail laid on edge (i, j). And calculated as

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{q}{L_k} & \text{if } k^{th} \text{ ant uses (i, j) in tour} \\ 0 & \text{otherwise} \end{cases}$$

Here Q is constant and L_k= tour length of k-th ant.

3. SELF-ADAPTIVE APPROACH

In Self-Adaptive Approach, the parameters are encoded into pheromones and undergo mutation and recombination. The idea is that better parameters leads to better pheromones for finding shortest path or largest path, according to combinational problem. In [5], a self-adaptive approach, a single mutation rate is used. With this mutation rate p [0, 1], a new mutation rate p' [0, 1] is found using equation (1). In this equation, γ is the learning rate which controls the adaption speed and it is taken as 0.22 in [5]. The mutation rate is not allowed to go below 1/L.

$$p' = \left(1 + \frac{1-p}{p}\exp\left(-\gamma.N(0,1)\right)\right)^{-1} \qquad (1)$$

4. BENCHMARK FUNCTIONS

From the standard set of benchmark problems available in the literature, five important functions are considered to test the accuracy and efficiency of the proposed work. All the problems are composed of continuous variables and have different degree of complexity and multi-modality. The set of test functions include unimodal and multimodal functions which are scalable (the problem size can be varied as per the user's choice). The problems are listed in Table 2.

5. PROPOSED ALGORITHM

In this proposed method, Ant colony optimization algorithm with uniform mutation operator using self-adaptive approach is used.Here mutation operator is used for enhancing the algorithm escape from local optima. In this method, an additional operator, mutation operator, is used and the new mutation rate is generated by the self-adaptive approach using equation (1). Here ACO algorithm generates the current solution (w). by using mutation operator, random position is changed by new mutation rate in current solution(w). After changing random position, new solution (w') is generated. Then the cost of this new solution (w') is compared by the current solution (w), if the cost of new solution is less than (or greater than) current solution, according to combinational problem, then new solution is replaced by current solution. This process is repeated until maximum iteration is not reached.

The proposed algorithm is given below:

Begin

Initialize

While stopping criterion not met **do**

Place each ant in a starting node;

Repeat

For each ant **do**

Choose next node by applying the state transition rule

Apply local pheromone update;

End for

Until each ant has built a solution

Update best solution (**w**);

If mutation criteria is met **then**
Select random position from current solution (**w**).
Apply mutation operator to generate new solution (**w'**) from current solution (**w**);

If ((Cost (**w'**) < Cost (**w**)) OR (Cost(w') > Cost(w)) then

Update w with **w'**;

Apply global pheromone update;

End While

End

6. EXPERIMENTAL RESULTS

This paper proposed a comparison between proposed modified ACO and reference [12] on a set of five benchmark functions shown in table 2. Here D, OPT, P, U and M indicates dimension, optimal value, peak number, unimodal and multimodal respectively. Control parameter values used in this experiment are shown in table 1.

Table 1. Control parameter values

Parameters	Value
ρ	0.8
q_0	0.5
α	0.9
Max number of iterations	100

7. CONCLUSION

This paper presents a modified ant colony optimization problem with uniform mutation based on self-adaptive approach. Here one additional step in the form of mutation operator is used in the original ACO algorithm for finding the global optimal solution. The use of mutation operator is for enhancing the algorithm escape from local optima. The experimental results showed that the proposed algorithm performs better than the previous algorithms presented in [12].

8. ACKNOLDGEMENTS

This work is supported by Maharana Pratap College of Technology, Gwalior-India.

9. REFERENCES

[1] A. Colorni, M. Dorigo, V. Maniezzo, "Distributed optimization by ant colonies".Proceedings of European Conference on Artificial Life, Paris, France, pp. 134-142, 1991.

[2] H. Md. Rais, Z. A. Othman, A.R. Hamdan, *Reducing Iteration Using Candidate List*, IEEE, 2008.

[3] H. Md. Rais, Z. A. Othman, A.R. Hamdan, *Improvement DACS3 Searching Performance using Local Search*,Conference on Data Mining and Optimization, IEEE, 27-28 October 2009.

[4] J. Han, Y. Tian, *An Improved Ant Colony Optimization Algorithm Based on Dynamic Control of Solution Construction and Mergence of Local Search Solutions*, Fourth International Conference on Natural Computation, IEEE, 2008

[5] Storn, R. and K. Price. 1995. Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimisation over Continuous Spaces. Technical Report TR-95-012, ICSI.availableviaftp://ftp.icsi.berkeley.edu/pub/techreports/1995/tr-95012.ps.z

[6] Storn R. and K. Price. 1997. Differential evolution – A Simple and Efficient Heuristic for Global Optimisation over Continuous Spaces. Journal of Global Optimisation, 11(4), 341-359.

[7] C-MihaelaPintea, D. Dumitrescu, "*Improving Ant System Using A Local Updating Rule"*, Proceedings of theSeventh International Symposium and Numeric Algorithms for Scientific Computing (SYNASC'05), IEEE 2005.

[8] V. Jakob, T. Ren, "A comparative study of differential evolution, particle swarm optimization, and evolutionaryalgorithms on numerical benchmark problems". Proceedings of Congress on EvolutionaryComputation, vol. 2, Poland, pp. 1980-1987, 2004.

[9] X-song, B. LI, H. YANG, *Improved Ant Colony Algorithm and Its Applications in TSP*, Proceedings of Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06), IEEE, 2006.

[10] Y. Zhang, Z-l.Pei, J-h.Yang, Y-c. Liang, *An Improved Ant Colony Optimization Algorithm Based on Route Optimization and Its Applications in Traveling Salesman Problem*, IEEE 2007. 1-4244-1509-8.

[11] R. Gan, Q. Guo, H. Chang, Y. Yi, *Improved Ant Colony Optimization Algorithm for the Traveling Salesman Problems*, Jouranl of Systems Engineering and Electronics, April 2010, pp 329-333.

[12] Y. Lin, H.Cai and J.Xiao, “Pseudo Parallel Ant Colony Optimization for Continuous Functions”, Third International Conference on Natural Computation (ICNC 2007), 0-7695-2875-9/07.

Table 2. Real Function List of Functions

Function	D	OPT	P
$F_1 = \sum_{i=1}^{D} x_i^2$	30	0	U
$F_2 = \sum_{i=1}^{n} mod x_i + \prod mod x_i$	30	0	U
$F_3 = \sum_{i=1}^{n} -x_{i\,\sin(\sqrt{mod\,x_i})}$	30	-12569.35	M
$F_4 = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	0	M
$F_5 = -20\exp\left(-0.2\sqrt{1/n \sum_{i=1}^{n} x_i^2}\right) - \exp\left(1/n \sum_{i=1}^{n} cos 2\pi x_i\right) + 20 + e$	30	0	M

Table 3 shows the experimental results between reference [12] and modified ACO

Function Name	Avg. Value using SACO	Avg. Value using PACO	Avg. Value using Proposed ACO
F1	2.8764e-258	0	2.8764e-231
F2	3.2511e-005	1.7532e-019	1.7329e-017
F3	-12569.4863	-12569.48661	-12569.48732
F4	0	0	0
F5	-78.3323	-78.3323	-78.3324