CO-T1

38th Annual Iranian Mathematics Conference

3-6 September 2007, University of Zanjan, Zanjan, Iran, 383-385.

COMPARING THE ANT COLONY ALGORITHM AND SIMULATED ANNEALING IN OPTIMIZATION OF MULTIMINIMA CONTINUOUS FUNCTIONS

MAHDI ABACHIZADEH *, FARHAD KOLAHAN, FARHAD RAZMPOUR

ABSTRACT. In this paper, an ant colony algorithm for optimizing multiminima continuous functions is presented. The quality of results is evaluated by comparing with the outcomes obtained by employing the simulated annealing method.

1. INTRODUCTION

Ant algorithms are a group of nature-inspired heuristics which have been efficiently applied on several combinatorial problems. However, the extension of their application to continuous search domains has been always a challenging concept. The incremental solution formation, pheromone deposition and discrete probability distribution are the main features of the standard ant algorithms which are contrary to decision making facing continuous functions. The enhanced versions of ACO such as CACO [1], API [4] and CIAC [3] being proposed to tackle this problem have all fundamentally diverged from the original ACO. In addition, for mixed discrete-continuous optimization problems, they fail to operate. Here, it is tried to propose a simple method based on the Ant Colony System (ACS) firstly introduced by Dorigo and Gambardella [2].

2000 *Mathematics Subject Classification*. Primary 68T20.

Key words and phrases. ant colony algorithm, simulated annealing, function optimization.

* Speaker.

2. Algorithm description

Let $R \ni f(x) : x \in R^n$ be defined where $x_{i\,\min} \leq x_i \leq x_{i\,\max}$ with $i = 1, 2, 3, ..., n$ is considered. The objective is to minimize the function $f(x)$. Depending on the requested accuracy of solution, each domain of design variable x_i is divided to k-1 sub-domains. Therefore, k different values are available for selection and evaluation. At each step, each ant builds a solution S which is composed of n design variables chosen from their corresponding discretized search space. It is noted that the decision making process for each variable x_i is performed independently based on the following state transition rule:

$$S_i = \begin{cases} \arg\max\,[\tau(i,j)] & \text{if } q \leq q_0 \\ p(\dfrac{\tau(i,j)}{\sum_{u=1}^{k}[\tau(i,u)]}) & \text{otherwise} \end{cases} \tag{2.1}$$

where $\tau(i,j)$ is the amount of pheromone on the jth element of the discretized domain for variable x_i. The parameter q_0 determines how deterministically, the first term, or stochastically, the second term, the algorithm works. In addition, q is a random number generated in the domain $[0, 1]$.

It is notable that the heuristic function which accelerates the convergence in standard ACS is safely neglected due to difficulties in its definition. Moreover, the concept of *tour* which is related to the sequence of cities in the TSP problem, the problem for which the ACS was first introduced and can be best understood, no longer exists. On the other hand, the local and global updating rules are performed just as they are introduced in the ACS.

3. Algorithm evaluation

To evaluate this proposed algorithm, it is employed to optimize some benchmark functions. Considering the fact that in continuous function optimization, the number of function evaluations and not the CPU time is regarded as the index for assessing the efficiency of an algorithm, the average amount of each objective function after definite numbers of function evaluations is the selected criterion. In addition, results acquired by utilizing the standard simulated annealing (SA) and the known global optima of the functions are all presented in Table 1 leading to a more meaningful comparison.

4. Discussion

The proposed method has been able to converge toward the global optimum for all the benchmark functions where in comparison to SA, it has resulted in much smaller amount of objective function for a predefined number of function evaluation. Regarding the CPU time, the ACO based method is less efficient especially when higher accuracy is required. The reason can be explained due to the nature of ACO calculations and also the severe increasing trend of memory volume to represent discretized variables with the increase in the required meaningful solution accuracy.

Table 1. Performance of ACO and SA algorithms for 5000 function evaluations averaged after independent runs

Function	Easom	Goldstein-Price	Ackley's path 10
Search domain	[-100,100]	[-2,2]	[-32,32]
Known optimum (x_1)	π	0	0
Known optimum (x_2)	π	-1	0
Known optimum $f(x)$	-1	3	0
x_1 by SA	3.2288	0.0188	0.3655
x_2 by SA	3.2917	-1.0244	0.1509
$f(x)$ by SA	-0.9427	3.4201	2.4061
x_1 by ACO	3.1511	0.0046	0.0070
x_2 by ACO	3.1471	-1.0014	0.0348
$f(x)$ by ACO	-0.9999	3.0111	0.1604
$\Delta f(x)$ by SA	0.0573	0.4291	2.4061
$\Delta f(x)$ by ACO	0.0001	0.0111	0.1604

References

[1] G. Bilchev and I. C. Parmee, *The ant colony methaphor for searching continuous design spaces*, Lect. Notes. Comput. Sc. **993**, (1995) 25-39.

[2] M. Dorigo and L. M. Gambardella, *Ant Colony System: A cooperative learning approach to the Traveling Salesman Problem*, IEEE. T. Evolut. Comput. **1**(1), (1997) 53-66.

[3] J. Dreo and P. Siaree, *A new ant colony algorithm using the heterarchical concept aimed at optimization of multiminima continuous functions*, Lect. Notes. Comput. Sc. **2643**, (2002) 216-221.

[4] N. Monmarche, G. Venturini and M. Slimane, *On how Pachycondyla apicalis ants suggest a new search algorithm*, Future. Gener. Comp. Sy. **16**, (2000) 937-946.

Department of Mechanical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran.

E-mail address: abachi@mail.com