2013 IEEE Congress on Evolutionary Computation
June 20-23, Cancún, México

An Efficient Ant Colony Optimization Algorithm for Function Optimization

A Guided search technique for high and low dimensional functions

Gautam Garai
Computational Science Division
Saha Institute of Nuclear Physics
Kolkata, India
gautam.garai@saha.ac.in

Shayantan Debbarman, Tamalika Biswas
I.T Department
Calcutta Institute of Engineering and Management
Kolkata, India
shayantangel@gmail.com, tamalika.1206@gmail.com

Abstract — **In this article we have proposed an efficient Ant Colony Optimization method, namely Guided Ant Colony Optimization (GACO) technique for optimizing mathematical functions. The search process of the optimization approach is directed towards a region or a hypercube in a multidimensional space where the amount of pheromone deposited is maximum after a predefined number of iterations. The entire search area is initially divided into 2^n number of hypercubic quadrants where n is the dimension of the search space. Then the pheromone level of each quadrant is measured. Now, the search jumps to the region (new search area) of maximum pheromone level and restarts the search process in the new region. However, the search area of the new region is reduced compared to the previous search area. Thus, the search advances and jumps to a new search space (with a reduced search area) in several stages until the algorithm is terminated. The GACO technique has been tested on a set of mathematical functions with number of dimensions upto 100 and compared with several relevant optimizing approaches to evaluate the performance of the algorithm. It is observed that the proposed technique performs better or similar to the performance of other optimization methods.**

Keywords— Ant Colony Optimization; global optimization; pheromone; optimization; mathematical functions

I. INTRODUCTION

The Ant Colony Optimization (ACO) is a metaheuristic search algorithm for solving various optimization problems [1, 2]. The algorithm is inspired by observing the behaviour of ant colonies in nature. The ACO basically imitates the behaviour of real ants searching for food. The ants communicate information about the source of food by depositing a substance called pheromone. The paths with higher pheromone levels have a greater possibility to be chosen and the pheromone intensity of paths that are not chosen is decreased by evaporation. This form of indirect communication is known as stigmergy and provides the ant colony with shortest path finding capabilities [2].

The ACO is employed with artificial ants to find the global solution for optimization problems [2]. The first algorithm following the principles of ACO is the Ant System [3] where the ants iteratively construct solutions and add pheromones to the paths corresponding to these solutions. The ACO has been applied for solving different problems [2] like vehicle routing, scheduling, timetabling, travelling salesman problem, data mining [4 – 8] and many more.

Apart from these, the researchers have always made an attempt to enhance the performance of the ACO with or without hybridization for solving problems in discrete as well as continuous domains. Dreo and Siarry [9] proposed the continuous interacting ant colony technique, where the ants communicate not only through indirect stigmergic information but also directly. Socha and Dorigo [10] developed an improved ACO algorithm for solving optimization problems in a continuous domain. Marimi et al. proposed a hybrid approach of ACO with tabu search method [11] to guide the search towards the area containing the global optimum. Another hybridized approach of ACO with Particle Swarm Optimizer was developed for finding the global minimum. In this process, both methods calculated the value of objective functions separately and the best value was selected as the global best.

We have proposed a new optimization algorithm based on ACO for solving global optimization problems in both discrete and continuous domains. The name of the algorithm is Guided Ant Colony Optimization (GACO). The search in GACO is directed in multiple stages by the amount of pheromone deposited in a region after the end of a pre-specified number of iterations in each stage. It can find the global optimum in a low dimensional as well as a high dimensional (upto 100) space for a set of mathematical functions. In GACO, the search advances and jumps to a new region (i.e. the neighboring hypercube in a multi-dimensional space) with maximum pheromone level. The space of the new search region is smaller than the previous (hypercubic) search area as shown in Fig. 1. The reduction of search space is done along all dimensions. The space is reduced in multiple stages with progress of the search process. If the search space is reduced slowly, then the possibility to come out of local optima and the convergence possibility to the global optimum are increased. On the other hand, if the search space is reduced faster, then there is a possibility to miss the global optimum since the process has no backtracking capabilities. We have compared our

978-1-4799-0454-9/13/$31.00 ©2013 IEEE

proposed method with some relevant optimization techniques testing on a set of standard benchmark functions.

The rest of the paper is organized as follows. Section II describes the algorithms of simple ACO and GACO techniques. It also discusses the search space partitioning process with progress of the search. The experimental results along with discussion is provided in Section III. Section IV concludes the paper.

II. FUNCTION OPTIMIZATION USING ACO

A global optimization problem can generally be formulated as a pair (S, f) where $S \subseteq R^n$ is a bounded set on R^n and $f : S \longmapsto R$ is an n-dimensional real-valued function. The objective of the problem is to find a point $\boldsymbol{x}_{opt} \in S$ on R^n such that $f(\boldsymbol{x}_{opt})$ is a global optimum on S. We have to find $\boldsymbol{x}_{opt} \in S$ according to the following equations, for minimization or maximization problems, respectively:

$$\forall x \subseteq S : f(\boldsymbol{x}_{opt}) \leq f(x) \tag{1}$$

$$\forall x \subseteq S : f(\boldsymbol{x}_{opt}) \geq f(x) \tag{2}$$

where f may not be a continuous function but it must be bounded.

A. Simple ACO Algorithm

The Simple ACO (SACO) is a global optimization method, which has been used for solving various optimization problems [1 – 10]. The SACO can be represented algorithmically as follows.

1) Algorithm of Simple ACO technique

```
1. Initialize pheromone trail.
2. For all ants
       Create solution using pheromone trail.
3. Evaporate the pheromone trails of all solutions by a certain amount.
4. Update pheromone trails according to the solutions created.
5. If stopping criteria are not satisfied, go to step 2.
6. Stop.
```

B. Optimization Using GACO Method

We have modified the simple ACO technique into a guided ACO approach by directing the search towards the area where the deposition of pheromone is maximum and reducing the search space according to the user's choice. The repetitive reduction of the search area in a 2-D space is described in the following section.

Initially, the GACO algorithm starts searching to find the optimum in the entire search space. The search space is divided into a number of quadrants (depending on the problem's dimensionality) in the multi-dimensional space where each quadrant will form a hypercube. The search space partitioning is necessary to measure the pheromone level in each partition. If the problem dimension is denoted by n, the number of quadrants (q) will be calculated as follows.

$$q = 2^n \tag{3}$$

The GACO method then runs for a certain number of iterations (say, I_k) and measures the pheromone level in each quadrant after the completion of I_k iterations. The pheromone level is measured to direct the search process towards the area with maximum amount of pheromone. We have considered the amount of pheromone deposited in each iteration as p, defined by the following expression.

$$p = \frac{1}{n} \tag{4}$$

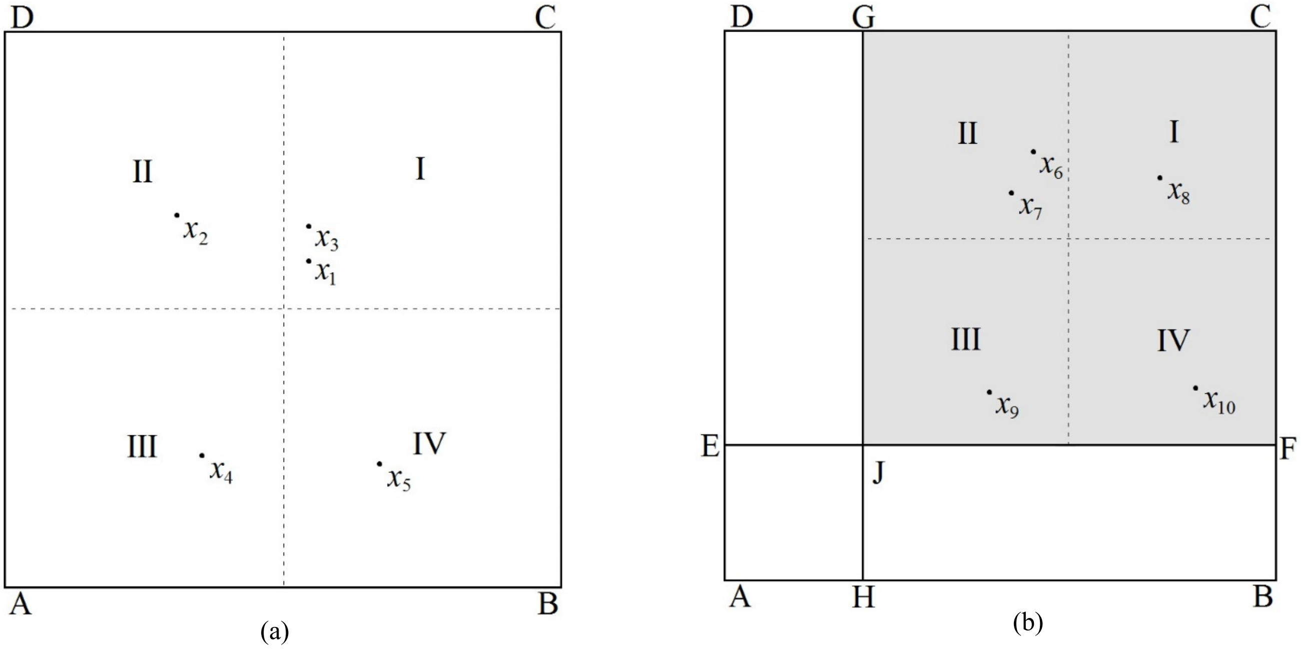
The GACO technique runs iteratively in multiple stages and in each iteration, we find the quadrant in which the best value of that iteration lies. Then the pheromone level of the corresponding quadrant is increased. The amount of p_j in the j-th quadrant where $j \in \{1,2, \cdots\cdots\cdots, 2^n\}$ is increased by $\frac{1}{n}$ in each iteration. Once the optimization method completes I_k iterations, the amount of pheromone $(p_j) \, \forall \, j \in \{1, 2, \cdots\cdots\cdots, 2^n\}$ deposited in each quadrant is calculated. If the amount of deposited pheromone p_m in the m-th (where $1 \leq m \leq 2^n$) quadrant is maximum, the search then moves towards the m-th quadrant. The search space is then redefined surrounding the m-th quadrant. The space is reduced along all dimensions (discussed in detail in Section II C). The ant population is regenerated except the elite one. The GACO approach restarts in the new search area and continues again for I_k times before it is transferred to another new search space considering the highest pheromone level. The GACO finally terminates after the completion of I_{max} iterations.

1) Algorithmic representation of the GACO technique

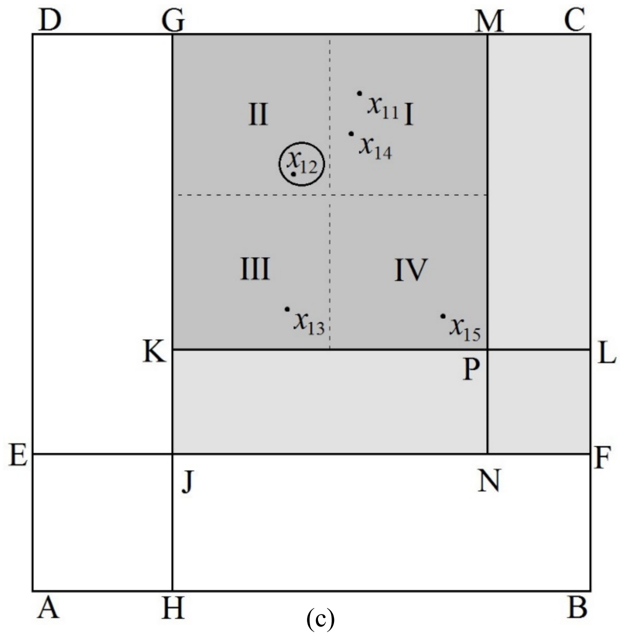
```
1. Initialize the population of ants, other parameters and iteration $I = 1$.
2. Create solutions for all ants and partition the search space into $2^n$ quadrants.
3. Find the quadrant where the best solution lies in each iteration. Increase the pheromone level of the corresponding quadrant by $\frac{1}{n}$ following (4).
4. Identify the quadrant $(q_m) \forall m \in \{1, 2, \cdots\cdots\cdots, 2^n\}$ with highest amount of pheromone deposited after the completion of $I_k$ iterations.
5. Redefine the search space surrounding the area of the quadrant $q_m$ and regenerate the population of ants following the elitist model.
6. Move the search to the new search space (see Fig. 1) which is smaller in size than the previous search space and restart the search process.
7. Increment $I = I + I_k$ . If $I \leq I_{max}$ , then go to step 2.
8. Stop.
```

C. *Search Space Reduction and Transferring of the GACO Method*

In Fig. 1, we have shown the search space reduction after every I_k ($= 5$, say) iterations in three stages and the shifting of GACO to the new space at each stage in 2-D space. Initially the entire space (ABCD) is divided into 4 quadrants since it is a 2-D space and the GACO technique starts searching in the area ABCD (see Fig. 1(a)). The search is continued for I_k i.e. 5 times. Let us now assume that the best values (x_1, x_3) of 1st and 3rd iterations lie on the quadrant-I as shown in Fig. 1(a). We have also assumed that the remaining three best values (x_2, x_4 and x_5) of 2nd, 4th and 5th iterations were found in the remaining 3 quadrants as depicted in Fig. 1(a). As a result, the amount of pheromone deposited in the quadrant-I is ($\frac{1}{2}+\frac{1}{2}$) i.e. 1 according to (4). The pheromone in each of the other three quadrants (II, III, and IV) is $\frac{1}{2}$. So, the GACO jumps towards the area surrounding the quadrant-I since this quadrant contains the maximum amount of pheromone. Once this is determined, the search space is redefined in the following way. We reduce the search space by a specific percentage (say, $u\%$) along all dimensions surrounding the quadrant-I of Fig. 1(a). In Fig. 1(b), it is shown that the search space is reduced along both *x* and *y* directions. Along x-axis, the length of AB is reduced by $u\%$ such that AH $=$ $u\%$ of AB and HB $=$ AB $-$ AH . Similarly, along y-axis BF $=$ $u\%$ of BC and FC $=$ BC $-$ BF , as shown in Fig. 1(b). The new search space is now JFCG (see Fig. 1(b)). We again partition the present search space JFCG into 4 quadrants as before (see Fig. 1(b)) and the GACO method restarts the search process at stage 2 with a new population of ants following the elitist model. In the space JFCG, let us consider that the search finds two best values (x_6 and x_7) of 6th and 7th iterations (out of 5 iterations) in the quadrant-II of Fig. 1(b) and each of the remaining 3 best values of the remaining three iterations lie in the other 3 quadrants as shown in Fig. 1(b). Similar to stage 1, the amount of pheromone deposited in the quadrant-II of Fig. 1(b) is 1 following (4) and in each of the remaining three quadrants is $\frac{1}{2}$. As a result, the search jumps towards the region surrounding the quadrant-II of Fig. 1(b) and the search space is redefined as previously described. In the final stage i.e. stage 3, the newly defined search space which is smaller in size (compared to the area of JFCG) is denoted by KPMG and it is partitioned into 4 quadrants as shown in Fig. 1(c). The search restarts in the space KPMG. Similar to the previous stages, we find the best solutions x_{11}, x_{12}, x_{13}, x_{14} and x_{15} in next five (11 – 15) iterations. Now the measurement of pheromone level is not necessary because now the search will not be transferred to another new region. Let the best solution at the end of stage 3 be x_{12}. Thus, x_{12} (circled in Fig. 1(c)) becomes the global or near-global optimum solution of the problem at the end of the final stage or at the end of the pre-specified number of iterations (I_{max} which in this case is 15).



(a) (b)



(c)

Figure 1. Reduction of search space and transferring of the GACO technique in 3 stages, (a) Partitioning of the space into 4 quadrants and best solutions of first 5 iterations (1 – 5) at stage 1, (b) Partitioning of the newly defined search space into 4 quadrants and best solutions of next 5 iterations (6 – 10) at stage 2, (c) Best solutions of last 5 iterations (11 – 15) and extraction of the global or near-global optimum solution at the final stage.

III. EXPERIMENTAL STUDIES

The performance of the GACO technique on a set of 20 mathematical functions was evaluated in two parts. In the first part, we have compared the results of GACO with several other relevant optimization algorithms to establish it as an efficient optimization algorithm. In the second part, we have optimized functions F2 – F15 to show the effectiveness of the proposed method in optimizing problems with higher dimensions (upto 100). We have considered the set of 20 standard benchmark functions taken from [12] which are listed in Table 1. The results of the optimization methods except the GACO in Table 3 were also taken from [12]. The functions F1 – F20 are a combination of unimodal and multimodal functions. The functions F16 – F20 are low dimensional compared to the other functions (F1 – F15) (see Tables 1 and 2). Table 2 depicts the basic parameters of the tested functions (F1 – F20).

1) *Parameter Setup*

In the experiment, we have considered that the population size of ants is μ ($= 50$) till the end of the proposed optimization algorithm. For each function in Table 1, the algorithm GACO was run 50 times and the average of the optimum results of 50 runs is tabulated in Table 3 for the GACO method. The dimension of the function is denoted by n as described in Table 2 for each function. We have run the GACO technique for I_k iterations to measure the amount of pheromone deposited in each quadrant (hypercube) for directing the search in the redefined search space for the next stage. The new search space is generated by reducing u% of the length along all dimensions of the previous search space (see Figs. 1(a) – 1(c)). For all tested functions in Table 1, we have considered the value of I_k to be 5 (except F14) and the value of u is set to 25 (except F14). If the value of I_k is increased towards the higher side of the range, the search process slows down i.e. more number of iterations are required to converge (by the algorithm). On the other hand, if u is set to a high value (more than 25), the algorithm (GACO) may not converge to a global or near-global optimum solution since the optimization method may have the possibility to miss the global or near-global optimum due to fast reduction of the search space after every I_k iterations. In the experiment, we have tried to maintain the values of I_k and u within the specified range such that the GACO approach converges faster to the global optimum for all the functions in Table 1.

TABLE 1. FUNCTIONS TESTED FOR OPTIMIZATION

$\mathbf{F1} = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$
$\mathbf{F2} = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$
$\mathbf{F3} = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + \exp(1)$
$\mathbf{F4} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
$\mathbf{F5} = \frac{\pi}{n}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2 \cdot [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\}$ $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ where $y_i = 1 + \frac{1}{4}(x_i + 1)$ and $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$
$\mathbf{F6} = \frac{1}{10}\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2 \times [1 + \sin^2(3\pi x_{i+1})]\right.$ $\left. + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$
$\mathbf{F7} = -\sum_{i=1}^{n}\sin(x_i)\sin^{20}\left(\frac{i \text{ x } x_i^2}{\pi}\right)$
$\mathbf{F8} = \sum_{i=1}^{n}\left[\sum_{j=1}^{n}(X_{ij}\sin\omega_j + \Psi_{ij}\cos\omega_j) - \sum_{j=1}^{n}(X_{ij}\sin x_j + \Psi_{ij}\cos x_j)\right]^2$ where $Xij \in [-100,100]$, $\Psi ij \in [-100,100]$ and $\omega ij \in [-\pi, \pi]$
$\mathbf{F9} = \frac{1}{n}\sum_{i=1}^{n}(x_i^4 - 16x_i^2 + 5x_i)$
$\mathbf{F10} = \sum_{i=1}^{n-1}[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$
$\mathbf{F11} = \sum_{i=1}^{n} x_i^2$

$$\mathbf{F12} = \sum_{i=1}^{n} x_i^4 + \text{random}[0,1)$$

$$\mathbf{F13} = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$$

$$\mathbf{F14} = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$$

$$\mathbf{F15} = \max\{|x_i|, i = 1, 2, 3, \cdots, n\}$$

$$\mathbf{F16} = 4x_1^2 - 2.1x_1^4 + \left(\frac{1}{3}\right)x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

$$\mathbf{F17} = \left(x_2 - \frac{5.1}{4\pi}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$$

$$\begin{aligned}\mathbf{F18} = {} & [1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \\ & \times [30 + (2x_1 + 3x_2)^2 \\ & \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]\end{aligned}$$

$$\mathbf{F19} = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$$

where $a_i, \cdots\cdots, a_{11} = [0.1957,\ 0.1947,\ 0.1735,\ 0.1600,\ 0.0844,\ 0.0627,\ 0.0456,\ 0.0342, 0.0323, 0.0235, 0.0246]$

and $[b_1, \cdots\cdots, b_{11}] = \left[4, 2, 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{6}, \frac{1}{8}, \frac{1}{10}, \frac{1}{12}, \frac{1}{14}, \frac{1}{16}\right]$

$$\mathbf{F20} = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2 \right]$$

where $[c_1, c_2, c_3, c_4] = [1, 1.2, 3, 3.2]$

$$[a_{ij}]_{4\text{x}6} = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}$$

and

$$[p_{ij}]_{4\text{x}6} = \begin{bmatrix} 1.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1415 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{bmatrix}$$

TABLE 2. BASIC PARAMETERS OF THE TESTED FUNCTIONS

Func.	Search Space	Dimension(n)	Optimal Value
F1	$[-500,500]^n$	30	-12569.5
F2	$[-5.12,5.12]^n$	30	0
F3	$[-32,32]^n$	30	0
F4	$[-600,600]^n$	30	0
F5	$[-50,50]^n$	30	0
F6	$[-50,50]^n$	30	0
F7	$[0,\pi]^n$	100	-99.2784
F8	$[-\pi,\pi]^n$	100	0
F9	$[-5,5]^n$	100	-78.33236
F10	$[-5,10]^n$	100	0
F11	$[-100,100]^n$	30	0
F12	$[-1.28,1.28]^n$	30	0
F13	$[-10,10]^n$	30	0
F14	$[-100,100]^n$	30	0
F15	$[-100,100]^n$	30	0
F16	$[-5,5]^n$	2	-1.0316285
F17	[-5,10]x[0,15]	2	0.398
F18	$[-2,2]^n$	2	3
F19	$[-5,5]^n$	4	$3.075\text{x}10^{-4}$
F20	$[0,1]^n$	6	-3.32

TABLE 3. COMPARATIVE RESULTS OF GACO WITH RELEVANT OPTIMIZATION ALGORITHMS

Func.	Algo.	M.F.E	G_{min}	S.D.
F1	ALEP	150,000	-11469.2	58.2
	FEP	900,000	-12554.5	52.6
	OGA/Q	302,116	-12569.453	$6.447\text{x}10^{-4}$
	M-L	655,895	-5461.826	275.15
	LEA	287,365	-12569.454	$4.831\text{x}10^{-4}$
	GACO	8,300	-12569.48	$1.9531\text{x}10^{-2}$
F2	ALEP	150,000	5.85	2.07
	FEP	500,000	$4.6\text{x}10^{-2}$	$1.2\text{x}10^{-2}$
	OGA/Q	224,710	0	0
	M-L	305,899	121.7575	7.7572
	LEA	223,803	$2.103\text{x}10^{-18}$	$3.359\text{x}10^{-18}$
	GACO	19,650	0	0
F3	ALEP	150,000	$1.9\text{x}10^{-2}$	$1.0\text{x}10^{-3}$
	FEP	150,000	$1.8\text{x}10^{-2}$	$2.1\text{x}10^{-2}$
	OGA/Q	112,421	$4.440\text{x}10^{-16}$	$3.989\text{x}10^{-17}$
	M-L	121,435	2.5993	$9.425\text{x}10^{-2}$
	LEA	105,926	$3.274\text{x}10^{-16}$	$3.001\text{x}10^{-17}$
	GACO	30,550	$3.641\text{x}10^{-14}$	$4.4704\text{x}10^{-14}$
F4	ALEP	150,000	$2.4\text{x}10^{-2}$	$2.8\text{x}10^{-2}$
	FEP	200,000	$1.6\text{x}10^{-2}$	$2.2\text{x}10^{-2}$
	OGA/Q	134,000	0	0
	M-L	151,281	$1.1894\text{x}10^{-1}$	$1.0404\text{x}10^{-2}$
	LEA	130,498	$6.104\text{x}10^{-16}$	$2.513\text{x}10^{-17}$
	GACO	9,950	0	$1.8411\text{x}10^{-2}$
F5	ALEP	150,000	$6.0\text{x}10^{-6}$	$1.0\text{x}10^{-6}$
	FEP	150,000	$9.2\text{x}10^{-6}$	$3.6\text{x}10^{-6}$
	OGA/Q	134,556	$6.019\text{x}10^{-6}$	$1.159\text{x}10^{-6}$
	M-L	146,209	$2.105\text{x}10^{-1}$	$3.6097\text{x}10^{-2}$
	LEA	132,642	$2.482\text{x}10^{-6}$	$2.276\text{x}10^{-6}$
	GACO	16,900	$1.5705\text{x}10^{-32}$	$1.5705\text{x}10^{-32}$
F6	ALEP	150,000	$9.8\text{x}10^{-5}$	$1.2\text{x}10^{-5}$
	FEP	150,000	$1.6\text{x}10^{-4}$	$7.3\text{x}10^{-5}$
	OGA/Q	134,143	$1.869\text{x}10^{-4}$	$2.615\text{x}10^{-5}$
	M-L	147,928	1.50534	2.25564
	LEA	130,213	$1.734\text{x}10^{-4}$	$1.205\text{x}10^{-4}$
	GACO	18,550	$1.3497\text{x}10^{-32}$	$1.3497\text{x}10^{-32}$
F7	OGA/Q	302,773	-92.83	$2.626\text{x}10^{-2}$
	M-L	329,087	-23.97544	$6.2875\text{x}10^{-1}$
	LEA	289,863	-93.01	$2.314\text{x}10^{-2}$
	GACO	9,250	-99.26889	$6.279\text{x}10^{-1}$
F8	OGA/Q	190,031	$4.672\text{x}10^{-7}$	$1.293\text{x}10^{-7}$
	M-L	221,547	$2.58778\text{x}10^{4}$	$1.73975\text{x}10^{3}$
	LEA	189,427	$1.627\text{x}10^{-6}$	$6.527\text{x}10^{-7}$
	GACO	3,350	$1.7886\text{x}10^{-7}$	$2.384\text{x}10^{-7}$
F9	OGA/Q	245,930	-78.300029	$6.288\text{x}10^{-3}$
	M-L	251,199	-35.80995	$8.9146\text{x}10^{-1}$
	LEA	243,895	-78.310	$6.127\text{x}10^{-3}$
	GACO	8,550	-78.332275	$9.2238\text{x}10^{-5}$
F10	OGA/Q	167,863	$7.520\text{x}10^{-1}$	$1.140\text{x}10^{-1}$
	M-L	137,100	2935.93	134.8186
	LEA	168,910	$5.609\text{x}10^{-1}$	$1.078\text{x}10^{-1}$
	GACO	15,450	0	$1.776\text{X}10^{-13}$

Func.	Algo.	M.F.E	G_{min}	S.D.
F11	ALEP	150,000	$6.32x10^{-4}$	$7.6x10^{-5}$
	FEP	150,000	$5.7x10^{-4}$	$1.3x10^{-4}$
	OGA/Q	112,559	0	0
	M-L	162,010	3.19123	$2.9463x10^{-1}$
	LEA	110,674	$4.727x10^{-16}$	$6.218x10^{-17}$
	GACO	49,650	0	0
F12	OGA/Q	112,652	$6.301x10^{-3}$	$4.069x10^{-4}$
	M-L	124,982	1.703986	$5.2155x10^{-1}$
	LEA	111,093	$5.136x10^{-3}$	$4.432x10^{-4}$
	GACO	6,350	$6.518x10^{-5}$	$7.6622x10^{-4}$
F13	FEP	200,000	$8.1x10^{-3}$	$7.7x10^{-4}$
	OGA/Q	112,612	0	0
	M-L	120,176	9.74160	$4.63769x10^{-1}$
	LEA	110,031	$4.247x10^{-19}$	$4.236x10^{-19}$
	GACO	42,350	$7.816x10^{-20}$	$8.336x10^{-20}$
F14	ALEP	150,000	$4.185x10^{-2}$	$5.969x10^{-2}$
	FEP	500,000	$1.6x10^{-2}$	$1.4x10^{-2}$
	OGA/Q	112,576	0	0
	M-L	155,783	2.21994	$5.0449x10^{-1}$
	LEA	110,604	$6.783x10^{-18}$	$5.429x10^{-18}$
	GACO	379,600	$2.512x10^{-20}$	$3.316x10^{-20}$
F15	FEP	500,000	$3.0x10^{-1}$	$5.0x10^{-1}$
	OGA/Q	112,893	0	0
	M-L	125,439	$5.5755x10^{-1}$	$3.9968x10^{-2}$
	LEA	111,105	$2.683x10^{-16}$	$6.257x10^{-17}$
	GACO	28,750	$1.834x10^{-17}$	$1.681x10^{-17}$
F16	ALEP	3,000	-1.031	0.00
	FEP	10,000	-1.03	$4.9x10^{-7}$
	M-L	13,592	-1.02662	$5.2656x10^{-3}$
	LEA	10,823	-1.03108	$3.364x10^{-7}$
	GACO	7,050	-1.0316285	$1.0505x10^{-8}$
F17	FEP	10,000	0.398	$1.5x10^{-7}$
	M-L	12,703	0.403297	$8.8324x10^{-3}$
	LEA	10,538	0.398	$2.652x10^{-8}$
	GACO	2,900	0.39806762	$7.939\ x10^{-5}$
F18	ALEP	3,000	3.000	0.000
	FEP	10,000	3.02	$1.1x10^{-1}$
	M-L	16,325	3.048855	$6.03749x10^{-2}$
	LEA	11,721	3.00003	$6.245x10^{-5}$
	GACO	5,300	3.0000	0
F19	FEP	400,000	$5.0x10^{-4}$	$3.2x10^{-4}$
	M-L	186,768	$1.3449x10^{-3}$	$2.9776x10^{-4}$
	LEA	55,714	$3.512x10^{-4}$	$7.361x10^{-5}$
	GACO	12,850	$6.8022\ x10^{-4}$	$3.978\ x10^{-4}$
F20	FEP	20,000	-3.27	$5.9x10^{-2}$
	M-L	92,516	-3.12696	$6.739x10^{-2}$
	LEA	28,428	-3.301	$7.832x10^{-3}$
	GACO	4,250	-3.3194523	$5.061x10^{-4}$

a. M.F.E denotes the average number of function evaluations to reach the optimum value

b. G_{min} indicates the mean of the best value found in the last generation for 50 runs

c. S.D. stands for Standard Deviation

TABLE 4. RESULTS OF GACO FOR THE FUNCTIONS WITH 100 DIMENSIONS

Func.	Algo.	M.F.E	G_{min}	S.D.
F2	GACO	20,600	0	0
F3	GACO	30,150	$1.358x10^{-13}$	$1.548x10^{-13}$
F4	GACO	12,950	0	0
F5	GACO	17,850	$4.7116x10^{-33}$	$4.7116x10^{-33}$
F6	GACO	18,550	$1.34978x10^{-32}$	$1.3497x10^{-32}$
F7	GACO	9,250	-99.26889	$6.279x10^{-1}$
F8	GACO	3,350	$1.7886x10^{-7}$	$2.384x10^{-7}$
F9	GACO	8,550	-78.332275	$9.2238x10^{-5}$
F10	GACO	15,450	0	$1.776X10^{-13}$
F11	GACO	50,800	0	0
F12	GACO	29,650	$2.8605x10^{-5}$	$1.0746x10^{-4}$
F13	GACO	42,100	$1.08x10^{-19}$	$3.14x10^{-19}$
F14	GACO	829,450	22.659	28.9189
F15	GACO	24,900	$4.417x10^{-18}$	$1.177x10^{-7}$

The terms of this Table are described under Table 3.

B. Results on Function Optimization

We have tested and compared the performance of the GACO technique with some optimization approaches and the results are shown in Table 3. Table 4 shows the optimum value achieved by the GACO method for the functions F2 – F15 when their dimensions are increased to 100.

The benchmark functions listed in Table 1 have been tested on various other optimization techniques. Among them, ALEP (Adaptive Levy Evolutionary Programming) [13] uses evolutionary programming with adaptive Levy mutation in order to generate an offspring for each new generation. It is also noted in [13] that the non-adaptive algorithm can never outperform ALEP. The OGA/Q (orthogonal genetic algorithm with quantization) [15] uses an orthogonal design to construct a crossover operator. Another evolutionary approach based method, FEP (Fast Evolutionary Programming) [14] uses evolutionary programming with Cauchy mutation to generate an offspring for each new generation. Hong and Quan [16] proposed a theoretical approach called the mean-value-level-set method (M-L method) by improving the mean of the objective function value on the level set. Wang and Dang [12] designed the LEA (level-set evolutionary algorithm) for global optimization with Latin squares. Its application leads to an effective crossover operator.

Table 3 contains the results of various optimization methods along with that of the GACO technique. We can see from Table 3 that for high dimensional functions (F1 – F15), the proposed method GACO performs better than or similar to the other optimization techniques. The GACO also reaches the global or near-global optimum faster than the other optimization approaches. For F14, the GACO algorithm reaches the global optimum solution if we set the value of u to 10 for search space reduction in multiple stages. Hence in case of F14, the proposed method takes longer compared to other techniques (except FEP) to achieve the best solution. For the remaining low dimensional functions (F16 – F20) in Table 3, it is noticed that the proposed method (GACO) can reach the global or near-global optimum. However, the GACO approach takes more time (in terms of mean function evaluation) to reach the global optimum for functions F16 and F18 compared to the optimization technique ALEP. Finally it is observed that the GACO technique performs either better or similar for most of the benchmark functions (F1 – F20) in terms of achieving the global or near-global optimum in lesser time compared to the other optimization methods.

We have further tested the functions F2 – F15 with 100 dimensions (see Table 4) to evaluate the effectiveness of the proposed method to achieve the global optimum for problems with higher dimensionality. We have not considered the other functions (except F2 – F15) since the optimum value would be different for them with higher dimensions and we had no gold standard for the others. It was noted that the GACO approach can reach the global or near-global optimum easily without spending much time compared to the timings for the functions F2 – F15 with 30 dimensions as shown in Table 4. However, for F14 with 100 dimensions, the technique cannot achieve the global or near-global optimum. Here, we have used $u = 10$ but it still cannot converge to the near-global optimum. In Table 4, we have not provided comparative results since we did not have any equivalent results for comparison (as shown in Table 3) on the set of functions with higher dimensionality (100). Finally, it is concluded that the GACO achieves the global or near-global optimum value efficiently for all functions with low and high (30 and 100) dimensionality except F14 (with 100 dimensions).

IV. CONCLUSION

The GACO method performs better than other relevant optimization techniques for the mathematical functions with 30 dimensions as observed by the results in Table 3. In case of F14, we have considered $u = 10$ so that the proposed technique converges to the global optimum. It is observed (for dimension = 30) that the GACO algorithm converges to the global optimum, but is slower than the other optimization techniques except FEP for F14. For low dimensional functions (F16 – F20), the proposed method performs better in achieving the global or near-global optimum but is slower (in terms of mean function evaluation) for F16 and F18 compared to ALEP. We have also evaluated the performance of the GACO approach for the mathematical functions F2 – F15 with 100 dimensions. It is noticed that the proposed method can reach the global or near- global optimum smoothly as it reached for the same set of functions with 30 dimensions. However, the GACO approach cannot reach the near-global optimum for F14, even if we set the value of u to 10.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their comments and valuable inputs on this paper which helped us to improve its quality. We would also like to thank Mr. Suvendu Nath Bose and Mr. Nanda Lal Sanpui of Saha Institute of Nuclear Physics and Mr. Chiranjib Patra of Calcutta Institute of Engineering and Management for their help at various stages of our work.

REFERENCES

[1] A. Colorni, M. Dorigo "Distributed optimization by ant colonies", Proc. Of European Conf. on Arificial Life, Paris, France, 1991, pp. 134-142.

[2] M. Dorigo, T. Stutzle, Ant colony optimization, MIT Press, Cambridge, 2004.

[3] M. Dorigo, V Maniezzo and A. Colorni, "Positive feedback as a search strategy" Departimento di Elettronica e Informatica, Politecnico di Milano, Milano, Italy, Tech. Rep. 91016, 1991.

[4] R. Montemami, L. M. Gambardella, A. E. Rizzoli and A. Donati, "Ant Colony systems for a dynamic vehicle routing problem", J. Combinatorial optim. , Vol. 10, no. 4, pp. 327-343, 2005.

[5] C. Blum "Beam-ACO-Hybridizing ant colony optimization with beam search: An application to open shop scheduling", Comput. Oper. Res. , vol. 32, no. 6, pp. 1565-1591, 2005.

[6] M. Socha, J. Knowles and M. Samples, "A Max-Min ant system for university timetabling problem", in Proc. 3rd Int. Workshop on Ant Algorithms, M. Dorigo, G. Di Caro and M. Samples, Sep. 2002, vol. 2463, pp. 1-13, Lecture notes in Computer Science.

[7] L. M. Gamberdella and M. Dorigo, "Ant-Q: A reinforcement learning approach to the travelling salesman problem", in Proc. 12th Int. Conf. Mach. Learn. , A. Prieditis and S. Russell, Eds. , Palo Alto, CA, 1995, pp. 252-260.

[8] S. Homdl, J. Knowles and M. Dorigo, Ant based clustering and topographic mapping, Artif. Life, vol. 12, no. 1, pp. 35-61, 2006.

[9] J. Dreo and P. Siarry, "A new ant colony algorithm using this hierarchical concept aimed at optimization of multi-minima continuous functions", in M. Dorigo et al.[Eds]. LNCS 2463, Springer-Verlag, Berlin, Germany, 2002, pp. 216-221.

[10] K. Socha and M. Dorigo, Ant colony optimization for continuous domains, European Journal for Operational Research, 185(2008), pp. 1155-1173.

[11] A. Karimi, H, Nokahari and P. Siarry, "Continuous ant colony syatem and tabu search algorithms hybridized for global minimization of continuous multi-minima functions", Computational Optimization and Applications, vol. 45, 2010, pp. 639-661.

[12] Y. Wang and C. Dang, "An evolutionary algorithm for global optimization based on Level-set evolution and latin squares" IEEE Trans. On Evol. Comput. , vol. 11, no. 5, October 2007, pp 579-595.

[13] C. Y. Lee and X. Yao, "Evolutionary programming using mutations based on the levy probability distribution," IEEE Trans. Evol. Comput. , vol. 8, no. 1, pp. 1-13, Feb. 2004.

[14] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," IEEE Trans. Evol. Comput., vol. 3, no. 2, pp. 82-102, July 1999.

[15] Y. W. Leung and Y. P. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," IEEE Trans. Evol. Comput., vol. 5, no. 1, pp. 41-53, Feb. 2001.

[16] C. S. Hong and Z Quan, 1988. "Integral global optimization: Theory, implementation and application". Springer-Verlag, Berlin, Germany.