Performance evaluation of Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) Algorithm

Ravi C. Butani, Bhavin D. Gajjar
Electronics and Communication department
L.D. College of Engineering,
Gujarat Technological University
Ahmedabad, India.
ravi_butani@yahoo.com,bhavin50@gmail.com

Rajesh A. Thakker
Electronics and Communication department
Vishwakarma College of Engineering,
Gujarat Technological University
Chandkheda, India.
rathakker2008@gmail.com

***Abstract*— Swarm intelligence is a research branch that models the population of interacting agents or swarms that are able to self-organize. An ant colony, a flock of birds or an immune system is a typical example of a swarm system. Bees' swarming around their hive is another example of swarm intelligence. Artificial Bee Colony (ABC) Algorithm is an optimization algorithm based on the intelligent behavior of honey bee swarm. In this work, ABC algorithm is used for optimize multivariable standard benchmark functions and the results of ABC algorithm are compared with results of Particle Swarm Optimization (PSO) algorithm and Memory loss Particle Swarm optimization (ML-PSO) Algorithm. The result shows that ABC outperforms the other algorithms.**

Keywords—ABC, benchmark function, heuristic search, ML-PSO, PSO, swarm optimization

I. INTRODUCTION

The particle swarm optimization (PSO) algorithm is introduced by J. Kennedy and R. Eberhart in 1995 and it has been efficiently used for many numerical optimization problems. Artificial bee colony (ABC) algorithm is one of the most recently introduced swarm optimization algorithm by Dervis Karaboga in 2007. In this paper comparison of both algorithms is carried out based on different evaluation parameters. Contents of this paper organized as follow, introduction of ABC and PSO algorithm given in section-I, details of standard benchmark functions used in this paper given in section-II, details of control parameter used in PSO and ABC and simulation results given in section-III, finally conclusion and references of the paper is given.

II. BASIC PARTICLE SWARM OPTIMIZATION

PSO is essentially a population-based algorithm. It starts with a random initialization of a swarm of particles. Each particle is modeled by its position in the search space and its corresponding velocity. In a n-dimensional search space, the position and the velocity of the i_{th} particle can be represented as $X_i = (x_i^1, x_i^2, \dots, x_i^n)$ and $V_i = (v_i^1, v_i^2, \dots, v_i^n)$ respectively. Each particle i is linked to some other particles of the swarm; these particles forms the "neighborhood" of particle i. The neighborhood of each particle can be chosen using either a fixed topology, or a time-varying topology, or a random topology. The quality of a given position is evaluated with respect to an objective function f .

Each particle i has its own best location $\overline{X}_i = (p_i^1, p_i^2, \dots, p_i^n)$, which corresponds to the best location particle i has reached until time t . The global best location is denoted by $\overline{X}_g = (g^1, g^2, \dots, g^n)$, which represents the best location reached by the neighbors of the i_{th} particle. From time t to time t +Δt, each velocity is updated using the following equation:

$$V_i(t + \Delta t) = wv_i(t) + p_1 r_1(\overline{X}_i - X_i) + p_2 r_2(\overline{X}_g - X_i) \quad (1)$$

where w is a constant, called inertia weight, p_1 and p_2 are constants called acceleration coefficients, and r_1 and r_2 are two independent random numbers uniformly distributed in [0, 1] for each dimension at each time step. w controls the influence of the previous direction of displacement. p_1 controls the influence of the particle's memory on the particle's behavior, and p_2 controls the influence of the swarm on the particle's behavior. The combination of the values of w, p_1 and p_1 may favor either intensification or diversification [1]

In the local version of the algorithm, at each time step, each particle adjusts its position and velocity as a function of its previous velocity, its best location and the location of the best particle among its neighbors. In the original version of PSO, the value of each component in V_i was clamped within the range [−Vmax,Vmax]. Velocity clamping controls excessive movement of the particles inside the search space and also prevents the particles from going out of the search space. If the computed velocity takes a particle out of the search space, many methods can be used.

The three most popular methods commonly applied are given below:

– the particle is stopped on the boundary

– the particle goes out of the search space and its fitness is not computed

– the particle is brought back into the search space on the nearest boundary.

The computation of the position at time t + Δt is given by:

$$X_i(t + \Delta t) = X_i(t) + v_i(t + \Delta t) \quad (2)$$

If the optimum is known a priori, a preset "acceptable" error can be defined as a stopping criterion. If not, it is common to stop after a maximum "reasonable" number of evaluations of the objective function or a maximum "reasonable" number of iterations. This maximum number of evaluations or iterations is a function of the complexity of the objective function and the time constraints given to the user. However, depending on the problem under consideration or the tests performed, other criteria may be used.

III. MEMORY LOSS (ML) OPERATION

The strength of the PSO algorithm is that each particle has a memory in the sense that it remembers the best position attained by it during its trajectory. However, in some cases, this feature makes the population vulnerable to the possibility of getting stuck in a local minimum. In the genetic algorithm, the mutation operation is generally used to bring the population out of a local minimum. The mutation operator has also been applied on the particle's velocity vector in the PSO algorithm [5] and was found to be useful. In this work, the mutation operator has been used to perform a "memory loss" operation, as follows. In each iteration, the particles are made to undergo the memory loss operation with a small probability. A random number is generated for each particle, and if it is less than certain specified probability (typically, 1%), a randomly selected component of its past best position (i.e., its memory) is replaced by a new random value.

The following comments may be made about the memory loss operation [5].

(1) In the PSO algorithm, each particle is attracted towards its own best position and the globally best position. In the ML operation, as the particles' own best positions are perturbed, it will be effective in changing the orientation of the search space.

(2) The memory loss operation is expected to continuously add diversity to the population at a small rate, thus possibly preventing the algorithm from getting stuck in a local minimum.

(3) From tests conducted on well-known multidimensional benchmark functions, it was observed that, when the PSO algorithm was stuck in a local minimum, only a small number of the position components were far from the solution. This observation motivated the choice made in our implementation, viz., only one of the components of the position is changed (i.e., allowed to undergo the memory loss operation). If the selected component happens to be the one with a large discrepancy with respect to the solution, the memory loss operation can be expected to be very effective in bringing the population out of a local minimum over few iterations.

(4) The ML operation allows the inertia range to be reduced, which further makes the PSO algorithm more efficient.

IV. ARTIFICIAL BEE COLONY ALGORITHM

In the ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. A bee waiting on the dance area for making decision to choose a food source is named an onlooker and a bee going to the food source visited by itself previously is named an employed bee. A bee carrying out random search is named a scout. In the ABC algorithm. First half of the colony consists of employed bees and the second half constitutes the onlookers. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source is exhausted by the employed and onlooker bees becomes a scout.

The main steps of the ABC algorithm are given below [3]:

• Initialize.

• REPEAT.

(a) Place the employed bees on the food sources in the memory;

(b) Place the onlooker bees on the food sources in the memory;

(c) Send the scouts to the search area for discovering new food sources.

• UNTIL (requirements are met).

In the ABC algorithm, each cycle of the search consists of three steps: sending the employed bees onto the food sources and then measuring their nectar amounts; selecting of the food sources by the onlookers after sharing the information of employed bees and determining the nectar amount of the foods, determining the scout bees and then sending them onto possible food sources. At the initialization stage, a set of food source positions are randomly selected by the bees and their nectar amounts are determined. Then, these bees come into the hive and share the nectar information of the sources with the bees waiting on the dance area within the hive. At the second stage, after sharing the information, every employed bee goes to the food source area visited by herself at the previous cycle since that food source exists in her memory, and then chooses a new food source by means of visual information in the neighborhood of the present one. At the third stage, an onlooker prefers a food source area depending on the nectar information distributed by the employed bees on the dance area.

As the nectar amount of a food source increases, the probability with which that food source is chosen by an onlooker increases, too. Hence, the dance of employed bees carrying higher nectar recruits the onlookers for the food source areas with higher nectar amount. After arriving at the selected area, she chooses a new food source in the neighborhood of the one in the memory depending on visual information. Visual information is based on the comparison of food source positions. When the nectar of a food source is abandoned by the bees, a new food source is randomly determined by a scout bee and replaced with the abandoned one. In our model, at each cycle at most one scout goes outside

for searching a new food source and the number of employed and onlooker bees were equal.

In the ABC algorithm, the position of a food source represents a possible solution of the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population.

At the first step, the ABC generates a randomly distributed initial population P(G = 0) of SN solutions (food source positions), where SN denotes the size of population. Each solution (food source) xi (i = 1, 2, . . . , SN) is a D-dimensional vector. Here, D is the number of optimization parameters. After initialization, the population of the positions (solutions) is subjected to repeated cycles, C = 1, 2, . . . ,Cmax, of the search processes of the employed bees, the onlooker bees and scout bees. An artificial employed or onlooker bee probabilistically produces a modification on the position (solution) in her memory for finding a new food source and tests the nectar amount (fitness value) of the new source (new solution). In case of real bees, the production of new food sources is based on a comparison process of food sources in a region depending on the information gathered, visually, by the bee. In our model, the production of a new food source position is also based on a comparison process of food source positions. However, in the model, the artificial bees do not use any information in comparison. They randomly select a food source position and produce a modification on the one existing in their memory as described in [4]. Provided that the nectar amount of the new source is higher than that of the previous one the bee memorizes the new position and forgets the old one. Otherwise she keeps the position of the previous one. After all employed bees complete the search process, they share the nectar information of the food sources (solutions) and their position information with the onlooker bees on the dance area. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount. As in the case of the employed bee, she produces a modification on the position (solution) in her memory and checks the nectar amount of the candidate source (solution). Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one.

An onlooker bee chooses a food source depending on the probability value associated with that food source, pi, calculated by the following expression

$$p = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (3)$$

where fiti is the fitness value of the solution i evaluated by its employed bee, which is proportional to the nectar amount of the food source in the position i and SN is the number of food sources which is equal to the number of employed bees (BN). In this way, the employed bees exchange their information with the onlookers.

In order to produce a candidate food position from the old one, the ABC uses the following expression (4):

$$v_{ij} = x_{ij} + \emptyset_{ij}(x_{ij} - x_{kj}) \quad (4)$$

where k ϵ{1, 2, . . . , BN} and j ϵ {1, 2, . . . ,D} are randomly chosen indexes. Although k is determined randomly, it has to be different from i. $\emptyset_{ij}$ is a random number between [−1, 1]. It controls the production of a neighbour food source position around x_{ij} and the modification represents the comparison of the neighbour food positions visually

by the bee. Equation 4 shows that as the difference between the parameters of the x_{ij} and x_{kj} decreases, the perturbation on the position x_{ij} decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is adaptively reduced. If a parameter produced by this operation exceeds its predetermined limit, the parameter can be set to an acceptable value. In this work, the value of the parameter exceeding its limit is set to its limit value.

The food source whose nectar is abandoned by the bees is replaced with a new food source by the scouts. In the ABC algorithm this is simulated by randomly producing a position and replacing it with the abandoned one. In the ABC algorithm, if a position cannot be improved further through a predetermined number of cycles called limit then that food source is assumed to be abandoned.

After each candidate source position v_{ij} is produced and then evaluated by the artificial bee, its performance is compared with that of x_{ij}. If the new food has equal or better nectar than the old source, it is replaced with the old one in the memory. Otherwise, the old one is retained. In other words, a greedy selection mechanism is employed as the selection operation between the old and the current food sources.

ABC algorithm in fact employs four different selection processes:

(1) a global selection process used by the artificial onlooker bees for discovering promising regions as described by equation [3],

(2) a local selection process carried out in a region by the artificial employed bees and the onlookers depending on local information (in case of real bees, this information includes the color, shape and fragrance of the flowers) (bees will not be able to identify the type of nectar source until they arrive at the right location and discriminate among sources growing there based on their scent) for determining a neighbor food source around the source in the memory as defined in equation [4],

(3) a local selection process called greedy selection process carried out by all bees in that if the nectar amount of the candidate source is better than that of the present one, the bee forgets the present one and memorizes the candidate source. Otherwise, the bee keeps the present one in the memory.

(4) a random selection process carried out by scouts. It is clear from the above explanation that there are three control parameters used in the basic ABC: The number of the food sources which is equal to the number of employed or onlooker bees (SN), the value of limit and the maximum cycle number (MCN).

V. BENCHMARK FUNCTIONS

A function is multimodal if it has two or more local optima. A function of variables is separable if it can be rewritten as a sum of functions of just one variable [4]. The separability is closely related to the concept of epitasis or interrelation among the variables of the function. The problem is even more difficult if the function is also multimodal. The search process must be able to avoid the regions around local minima in order to approximate, as far as possible, to the global optimum. The most complex case appears when the local optima are randomly distributed in the search space. The dimensionality of the search space is another important factor in the complexity of the problem. In order to compare the performance of the proposed ABC with PSO, and ML-PSO seven classical benchmark functions used in this work as given in table [I].

TABLE I. STANDARD BENCHMARK FUNCTIONS USED IN THIS WORK

Function, Globally optimum solution, Type
F_1, Sphere, Minimize, Origin, Unimodal $f_1(x)=\sum_{i=1}^{n} x_i^2$
F_2, Rosenbrock, Minimize, $[1, 1, \ldots, 1]^n$,Unimodal $f_2(x)=\sum_{i=1}^{n-1}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$
F_3, Rastrigin, Minimize, Origin, Multimodal $f_3=10n+\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)\right]$
F_4, Griewank, Minimize, Origin, Multimodal $f_4(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$
F_5, Ackley, Minimize, Origin, Multimodal $f_5(x)=20+\exp(1)-20\exp\left(\frac{-\|x\|_2}{5\sqrt{n}}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)$
F_6, Schaffer, Maximize, Origin, Multimodal $f_6(x)=\frac{\sin^2\left[\sqrt{x_1^2+x_2^2}\right]}{1.0+10^{-3}\left[x_1^2+x_2^2\right]^2}$
F_7, Schaffer, Maximize, Origin, Multimodal $f_7(x)=0.5-\frac{\left(\sin\left[\sqrt{x_1^2+x_2^2}\right]\right)^2-0.5}{\left(1.0+10^{-3}\left[x_1^2+x_2^2\right]\right)^2}$

TABLE II. INITIALIZATION PARAMETER AND OPTIMUM SOLUTION OF BENCHMARK FUNCTIONS

Fun.	Dim. (n)	Initialization range	Search space	Goal (f_0)
F_1	30	$(50, 100)^n$	$(-100,100)^n$	10^{-9}
F_2	30	$(15, 30)^n$	$(-100, 100)^n$	10^{-6}
F_3	30	$(2.56, 5.12)^n$	$(-10, 10)^n$	10^{-9}
F_4	30	$(300, 600)^n$	$(-600, 600)^n$	10^{-9}
F_5	30	$(15, 32)^n$	$(-32, 32)^n$	10^{-9}
F_6	02	$(15, 30)^n$	$(-100, 100)^n$	0.9940069
F_7	02	$(15, 30)^n$	$(-100, 100)^n$	0.999999

VI. CONTROL PARAMETER OF PSO AND ABC ALGORITHMS

Common control parameters of the algorithms are population size and the number of maximum generation. In the experiments, maximum number of generations were 500,750 and 1,000 for the dimensions 10, 20 and 30, respectively; and the population size was 125 as in Ref. [2]. Other control parameters of the algorithms and the schemes used in Ref. [2] are presented below; and also the values of these control parameters employed for PSO and ML-PSO in Ref. [5] are presented.

- PSO Control parameters

PSO equations are given in (1) and (2) where ω is the additional inertia weight, which varies from from 0.9 to 0.7 linearly with the iterations. The learning factors, c1 and c2 are set to be 1.49. The upper and lower bounds for v, (vmin, vmax) are set to be the maximum upper and lower bounds of x, i.e. (vmin, vmax) = (xmin, xmax). If the sum of accelerations would cause the velocity on that dimension v(t + 1), to exceed vmin or vmax, then the velocity on that dimension v(t + 1), is limited to vmin or vmax, respectively [2].

- ABC Control parameters

ABC algorithm has a few control parameters: Maximum number of cycles (MCN) equals to the maximum number of generation and the colony size equals to the population size, i.e. 125, as in the study presented in Ref. [6] . The percentage of onlooker bees was 50% of the colony, the employed bees were 50% of the colony and the number of scout bees was selected as one. The increase in the number of scouts encourages the exploration as the increase of onlookers on a food source increases the exploitation. Each of the experiments was repeated 30 times with different random seeds. The mean function values of the best solutions found by the algorithms for different dimensions have been recorded. The mean ($\bar{\epsilon}$),standard deviation (σ), success rate (S) and average no of function evaluation ($\overline{N_f}$) of the function values obtained by the ABC, PSO and ML-PSO are given in Table III.

TABLE III. SIMULATION RESULTS

Fun.		ABC	PSO[4]	ML-PSO[4]			
	w	-	0.9-0.4	0.9- 0.4	0.8-0.4	0.73-0.4	0.6 - 0.4
F_1	$\bar{\epsilon}$	0	0	0	0	0	0
	σ	0	0	0	0	0	0
	S	100	100	100	100	100	100
	$\bar{N}_f$	40119	67623	75314	43059	28004	16872
F_2	$\bar{\epsilon}$	5.91	16.45	0.95	0.016	0.09	2.5×10^{-4}
	σ	8.27	23.11	7.28	0.08	0.78	2.11×10^{-3}
	S	-	–	–	20	42	71
	$\bar{N}_f$	-	–	–	197532	178403	180080
F_3	$\bar{\epsilon}$	0	43.95	0	0	0	0
	σ	0	11.05	0	0	0	0
	S	100	–	100	100	100	100
	$\bar{N}_f$	83491	–	98560	88751	75493	64948
F_4	$\bar{\epsilon}$	0	0.014	0.012	0.012	0.012	0.021
	σ	0	0.014	0.015	0.014	0.018	0.025
	S	100	32	39	34	42	39
	$\bar{N}_f$	34603	69213	76218	50698	30754	18150
F_5	$\bar{\epsilon}$	0	0.15	0	0	0	0
	σ	0	0.45	0	0	0	0
	S	100	89	100	100	100	100
	$\bar{N}_f$	83130	82366	105064	74816	55665	41710
F_6	$\bar{\epsilon}$	0	0	0	0	0	0
	σ	0	0	0	0	0	0
	S	100	100	100	100	100	100
	$\bar{N}_f$	3150	10129	10864	3463	2426	1523
F_7	$\bar{\epsilon}$	0	9.77×10^{-5}	1.94×10^{-4}	8.74×10^{-4}	9.71×10^{-4}	2.23×10^{-3}
	σ	0	9.66×10^{-4}	1.36×10^{-3}	2.78×10^{-3}	2.91×10^{-3}	4.08×10^{-3}
	S	100	99	98	91	90	77
	$\bar{N}_f$	25012	22843	30478	20276	22819	32297

Conclusion

The performance of the ABC with PSO and ML-PSO which are also member of swarm intelligence algorithms is compared in this work. From the simulation results it was concluded that the ABC algorithm has the ability to get out of a local minimum and performs better for local search. ABC algorithm requires a few control parameters and they are independent from type or complexity of problem. Hence ABC algorithm can be efficiently used for multivariable, multimodal function optimization. There are several issues which remain as the scopes for future studies such as the investigation of the control parameter's effect on the performance of the ABC algorithm and the convergence speed of the algorithm.

ACKNOWLEDGMENT

We are thankful to Dr. R.A. Thakker and Mr. K.P. Acharya for providing valuable guidance to understand the various swarm optimization algorithms and its applications for present days optimization problems.

REFERENCES

[1] J. Kennedy and R. Eberhart, Particle swarm optimization, Proceedings of the IEEE Int. Conf. on Neural Networks, Piscataway, NJ (1995), pp. 1942–1948.

[2] Yann Cooren · Maurice Clerc · Patrick Siarry "Performance evaluation of TRIBES, an adaptive particle swarm optimization algorithm" Swarm Intell (2009) 3: 149–178

[3] Dervis Karaboga · Bahriye Basturk "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm" J Glob Optim (2007) 39:459–471

[4] Rajesh A. Thakker, M. Shojaei Baghini, and M. B. Patil "Automatic Design of Low-Power Low-Voltage Analog Circuits Using Particle Swarm Optimization with Re-Initialization" Department of Electrical Engineering, Indian Institute of Technology Bombay, 400076, India, Journal of Low Power Electronics Vol. 5, 1–12, 2009

[5] R. A. Thakker, Sanjay B. Prajapati and M. B. Patil, "Particle Swarm Optimization with Memory Loss Operation",Department of Electrical Engineering, Indian Institute of Technology Bombay, IEEE Trans. Sys.Man, and Cyber. – Part B: Cybernetics, vol. 34, no. 2, pp. 977-1004, Apr. 2009

[6] Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005

Ravi C. Butani PG student at L.D.College of Engineering,Gujarat Technological University,Ahmedabad. His active thesis work on ABC algorithem for automatic CMOS circuit design.Year-2011.

Bhavin D. Gajjar PG student at L.D.College of Engineering,Gujarat Technological University,Ahmedabad. His active thesis work on PSO algorithem for automatic CMOS circuit design.Year-2011.