A Cooperative Learning Strategy with Multiple Search Mechanisms for Improved Artificial Bee Colony Optimization

Fatima Harfouchi
Applied Automation Laboratory
University of Boumerdès
Avenue de l'indépendance, 35000
Boumerdès, Algeria
harfouchi.fatima03@hotmail.com

Hacene Habbi
Applied Automation Laboratory
University of Boumerdès
Avenue de l'indépendance, 35000
Boumerdès, Algeria
habbi_hacene@hotmail.com

***Abstract*--Artificial bee colony (ABC) optimization is a swarm based stochastic search strategy inspired by the foraging behavior of honeybees. Due to its simplicity and promising optimization capability, the ABC concept has devoted special interest with an increasing number of applications to scientific and engineering optimization problems. As an open research field, many researchers attempted to improve the performance of ABC algorithm through new algorithmic frameworks or by introducing modifications on the basic model. This paper presents an improved version of ABC algorithm based on a cooperative learning strategy with modified search mechanisms incorporated at both employed and onlooker levels. The proposed approach referred to as CLABC (Cooperative learning ABC) is tested on benchmark functions for numerical optimization. The results demonstrate the good performance and convergence of the proposed algorithm over other existing ABC variants.**

Keywords-component; Artificial Bee Colony (ABC) Algorithm, cooperative learning, search mechanism, swarm intelligence.

I. INTRODUCTION

In recent years, swarm intelligence optimization has become an attractive research field and shown a noticeable increasing number of fundamental and application studies. Nature inspired swarm techniques include various meta-heuristic approaches that have been successfully applied to solve complex optimization problems. Generally, swarm optimization concepts are derived from the collective behavior of social creatures. Social learning ability to solve complex tasks represents a key feature for the development of swarm optimization mechanisms. In fact, many swarm intelligence based algorithms are reported in the literature, such as particle swarm optimization (PSO) [1], ant colony optimization (ACO) [2], genetic algorithms (GA) [3], differential evolution (DE) [4], and so on. The ABC algorithm is one of the most popular swarm algorithms which was introduced by Karaboga in 2005 [5]. Since the ABC is simple in concept and easy to implement, it has rapidly gained the attention of researchers and applied to a variety of numerical optimization problems [6] [7] as well as engineering applications [8] [9] [10] [11]. The basic ABC model was subjected to many improvements aiming to enhance its computational performance and robustness. To this end, different modified and hybridized algorithms have been developed [12] [13] [14].

According to recent studies [15], ABC is better than or similar to other population based algorithms. However, ABC is good at exploration but poor at exploitation, and its convergence speed is also an issue in some cases. In order to achieve good performance on optimization problems, exploration and exploitation strategies should be well balanced. This is a challenging issue since so far it does not exist a specific algorithm that achieves optimal solutions for all optimization problems.

Our present contribution deals precisely with this problematic issue. The main purpose is to enhance the performance of ABC optimization methodology by introducing a new algorithmic framework based on random cooperative learning concept. In that framework, exploration and exploitation abilities are managed to balance adequately by incorporating different search strategies in both employed and onlooker phases.

The paper is organized as follows. Section 2 recalls basics of the standard ABC. The proposed cooperative learning algorithm is described in Section 3. Experiments and comparison results are shown and discussed in Section 4. Conclusions and remarks are given in Section 5.

II. ARTIFICIAL BEE COLONY ALGORITHM

The artificial bee colony (ABC) algorithm is a swarm stochastic search algorithm which imitates the foraging behavior of honeybees [5]. In ABC concept [6], the honey bee colony model consists of three kinds of bees: employed bees, onlooker bees and scouts. An employed bee searches around the current food source to find a new source position with better nectar amount. If the nectar amount of the discovered position is higher than that of the previous one, the bee saves the new position in her memory and forgets the old one. Each employed bee is associated with a food source, in other words, the number of employed bees is equal to the number of food sources.

After all employed bees complete their search processes, they share their information with onlooker bees which are waiting in the hive. An onlooker bee chooses a food source according to a probabilistic greedy selection mechanism. Therefore, food sources with better profitability will get higher probability to be selected by the onlookers. Similar to employed bees, each onlooker bee produces a modification on the position in her memory and investigates the nectar amount of the generated candidate source.

If a position cannot be improved further through a predetermined limit tolerance called "*limit*", then that food source is assumed to be abandoned. The corresponding employed bee becomes then a scout. The abandoned position will be replaced with a new food source found by the scout.

The first step in the framework of the ABC algorithm consists in generating a population of SN solutions (food source positions) randomly in the admissible search domain. The fitness of a food source will be evaluated. After initialization, the population of food sources (solutions) is subjected to repeated cycles of the search processes of employed bees, onlooker bees and scout bees.

Each employed bee generates a new candidate solution around its current position by the following equation :

$$v_i^j = x_i^j + \emptyset_i^j\left(x_i^j - x_k^j\right) \tag{1}$$

where $i = 1, 2, \ldots, SN$, $j \in \{1,2, \ldots, D\}$ and $k \in \{1,2, \ldots, SN\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. $\emptyset_i^j$ is a random number between $[-1, 1]$. Greedy selection between the old and the updated food source position is performed by the employed bee based on fitness value evaluation. This valuable information about the position and the quality of the food sources are shared with the onlooker bees.

In the next step, an onlooker bee chooses a food source with a probability that depends on its nectar value which is computed by :

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{2}$$

where fit_i is the fitness value of the solution i. It should be noted that onlooker bees also use Equation (1) to generate new solution candidates.

In the third step, the ABC algorithm deals with food sources abandonment. Any solution that cannot be improved through a predefined number of generations will be abandoned and replaced by a new position that is randomly determined by a scout bee.

III. COOPERATIVE LEARNING ARTIFICIAL BEE COLONY ALGORITHM

The proposed modified framework of ABC optimization concept involves an important aspect of natural behavior of honey bees which has no longer been considered in previous variants. Here, the highlighted important feature of swarm bees consists in the cooperative learning ability which might better characterize or model social learning between bees groups. The proposed cooperative learning artificial bee colony algorithm (CLABC) makes also use of different search strategies at both employed and onlooker levels with respect to the differences in learning ways and tools to achieve tasks that might effectively describe the natural behavior of swarm bees or groups in a bee colony. The search strategy in the proposed CLABC algorithm is based on multiple solution search equations which are integrated in different ways at the employed and onlooker stages.

The employed bees in the proposed CLABC algorithm are divided into 3 groups; each group forms a sub-population of potential solutions and uses search equations different from the other groups. The solution updating equations are given as follows :

$$v_i^j = x_i^j + \emptyset_i^j\left(x_i^j - x_k^j\right) \tag{3}$$

$$v_i^j = x_k^j + \emptyset_i^j\left(x_i^j - x_{k1}^j\right) \tag{4}$$

$$v_i^j = x_i^j + r_i^j\left(x_i^j - x_{mean}^j\right) + r1_i^j\left(x_k^j - x_{k1}^j\right) \tag{5}$$

$$v_i^j = x_{best}^j + r_i^j\left(x_i^j - x_{mean}^j\right) + r1_i^j\left(x_k^j - x_{k1}^j\right) \tag{6}$$

$$v_i^j = x_k^j + \emptyset_i^j\left(x_{k1}^j - x_{k2}^j\right) \tag{7}$$

$$v_i^j = x_{best}^j + \emptyset_i^j\left(x_k^j - x_{k1}^j\right) \tag{8}$$

where $= 1, 2, \ldots, SN, i \neq k \neq k_1 \neq k_2$, $j \in \{1,2, \ldots, D\}$, $\emptyset_i^j \in [-1, 1]$, $r_i^j, r1_i^j \in [0, 1]$.

Eq. (3) is same as the update rule of the basic ABC algorithm. In Eqs. (4), (5), (6), (7) and (8), x_k^j, x_{k1}^j, x_{k2}^j, are the jth dimensions of solutions selected randomly from the sub-populations. k, k_1, k_2, are not equal to each other and to i. x_{best}^j denotes the jth dimension of the best solution obtained by the sub-populations so far. x_{mean}^j is the average solution corresponding to the jth dimension for all solutions in the sub-populations.

In the first group, formulas (3) and (4) are used for generating the new candidate solutions. Only one of the two search equations ((3) or (4)) performs solution updating based on an increasing probability as in [16] which is defined by $(iter/MCN)^{1/2}$, where $iter$ is the current generation number and MCN is the maximum number of cycles. If the random value between 0 and 1 is smaller than or equal to the predefined probability, then Eq. (3) is used. Otherwise, search mechanism described by Eq. (4) is performed. Based on the same definition of probability given above, Eqs. (5) and (6) are used for the second group, and Eqs. (7) and (8) for the last group.

According to this strategy, global exploitation is first carried out at the beginning of the search process with neighboring and mean-value oriented updating equations (Eqs. (4), (5) and (7)). Then, all equations (Eqs. (3-8)) are called roughly with the same probability. Accordingly, both global exploration and local exploitation are balanced at this level. Later, Eqs. (3), (6) and (8) are performed to increase the exploitation capability and the convergence speed. Besides, in order to increase the convergence speed of the algorithm, employed bees are managed to start by changing one random dimension as basic ABC does. Then, when the number of iterations reaches $iter = MCN/5$, the algorithm will have to optimize randomly one or all parameters simultaneously. While reaching the end of the optimization process, i.e. between $iter = 3 * MCN/5$ and $iter = MCN$, the employed bees will update all parameters of the solution.

After completion of the employed bee phase, and based on greedy selection, the onlookers will start to change one random parameter of the solution by using Eq. (4). When $iter = MCN/5$, the onlooker bees will have to update randomly all parameters of the solution by using Eq. (3) or only one random dimension through Eq. (4). It should be noted that changing all dimensions at each iteration by the onlookers may cause premature convergence.

IV. EXPERIMENTS AND RESULTS

The performance of the proposed cooperative learning ABC algorithm is tested on nine well-known benchmark numerical functions taken from [15]. The functions are described in Table I.

In the simulation study, the population size is set to 20 divided into three sub-populations of size 6, 7 and 7 respectively. The maximum number of function evaluations (FEs) is taken as 120000. Each experiment is run 30 times. The abandon limit value is set to 200.

The mean and standard deviations of function values obtained by ABC, Gbest-guided ABC (GABC) [12], a novel ABC (NABC) presented in [13] and the proposed CLABC are given in Table II. The best results are marked in bold.

According to the results reported in Table II and convergence curves illustrated in Figs. 1-6, it can be clearly seen that the proposed CLABC algorithm outperforms all the compared ABC based algorithms on most benchmark functions. As can be noticed from Fig. 1 and Fig. 4, the best solution found by CLABC is reached after only a few function evaluations for f_1 and f_4 and solutions are significantly improved through successive evaluations for test functions f_2, f_6, f_7, and f_8. The performance of CLABC is due to improvements introduced at the employed and onlooker stages where exploration and exploitation abilities are balanced adequately through different solution updating equations performed over a cooperative learning behavior.

TABLE I. BENCHMARK FUNCTIONS

Function	**Mathematical Expression**	**Range (Dimension)**
Schaffer	$f_1 = 0.5 + \frac{sin^2\left(\sqrt{x_1^2+x_2^2}\right)-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	$[-100,\ 100]^D$ (D=2)
Colville	$f_2(x) = 100(x_1^2 - x_2)^2 + (x_1-1)^2 + (x_3-1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2-1)^2 + (x_4-1)^2) + 19.8(x_2-1)(x_4-1)$	$[-10, 10]^D$ (D=4)
Sphere	$f_3(x) = \sum_{i=1}^{D} (x_i)^2$	$[-100, 100]^D$ (D=30)
Griewank	$f_4(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^D$ (D=30)
Rastrigin	$f_5(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$ (D=30)
Rosenbrock	$f_6(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$ (D=30)
Quartic	$f_7(x) = \sum_{i=1}^{D} (ix_i)^4 + random[0,1)$	$[-1.28, 1.28]^D$ (D=30)
Powell	$f_8(x) = \sum_{i=1}^{D/4} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	$[-4, 5]^D$ (D=24)
Schwefel 1.2	$f_9(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]^D$ (D=30)

TABLE II. PERFORMANCE OF THE PROPOSED ALGORITHM ON BENCHMARK NUMERICAL FUNCTIONS

Function		ABC	GABC	NABC	CLABC
f_1	mean	**0**	3.1991E-05	3.0772E-05	**0**
	std	**0**	8.6664E-05	1.6855E-04	**0**
f_2	mean	1.9345E-01	3.2438E-02	9.3189E-03	**8.1943E-31**
	std	1.4438E-01	3.1114E-02	2.9454E-02	**2.5057E-30**
f_3	mean	1.6197E-44	1.9668E-57	**3.3877E-97**	8.2876E-62
	std	2.4209E-44	2.7656E-57	**8.8689E-97**	4.1997E-62
f_4	mean	1.2838E-09	1.4542E-03	4.9307E-04	**0**
	std	4.8108E-09	4.9732E-03	1.8764E-03	**0**
f_5	mean	**0**	**0**	**0**	**0**
	std	**0**	**0**	**0**	**0**
f_6	mean	2.2609E-01	14.8816	10.3040	**2.1489E-05**
	std	4.7061E-01	23.5410	17.6939	**7.2287E-05**
f_7	mean	1.1694E-01	5.8189E-02	2.6050E-02	**6.0184E-03**
	std	1.9945E-02	1.5579E-02	8.9058E-03	**1.5510E-03**
f_8	mean	2.3925E-02	2.4381E-02	4.4396E-02	**7.6554E-05**
	std	5.7974E-03	1.5188E-02	6.3814E-02	**5.8487E-05**
f_9	mean	5561.24	7733.54	3055.4	**3.5915E-03**
	std	1308.88	2662.2	1396.15	**1.4510E-02**

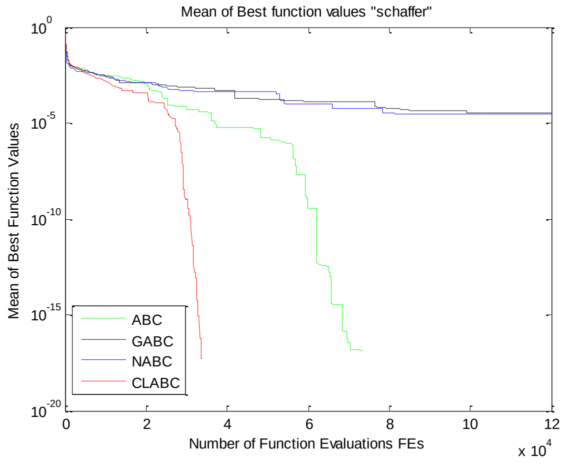

Fig. 1. Convergence curves of ABC, GABC, NABC and CLABC algorithms for f_1

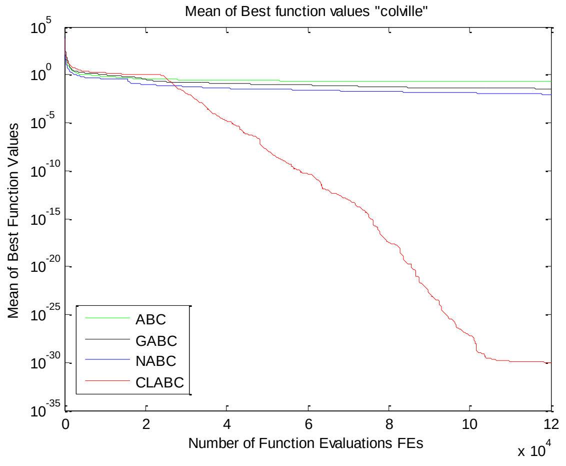

Fig. 2. Convergence curves of ABC, GABC, NABC and CLABC algorithms for f_2

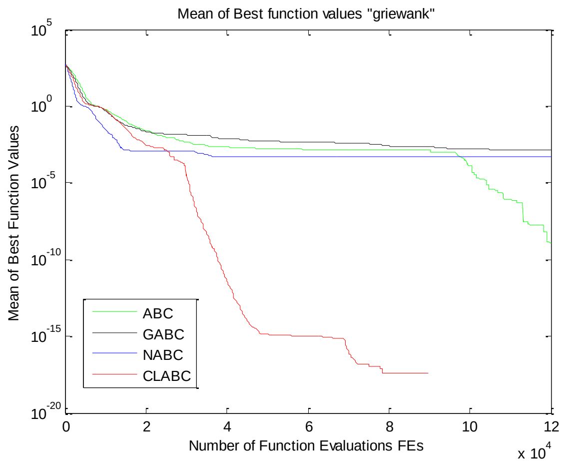

Fig. 3. Convergence curves of ABC, GABC, NABC and CLABC algorithms for f_4

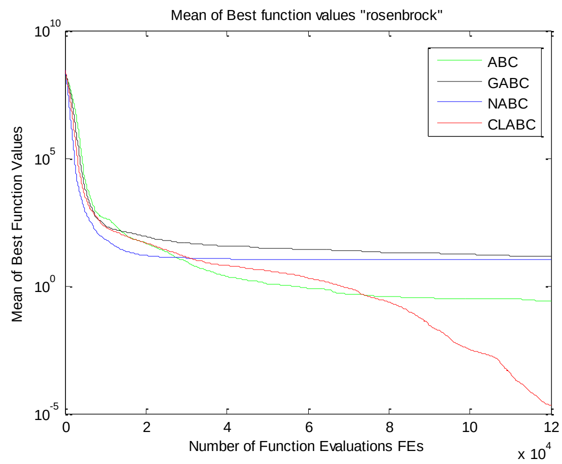

Fig. 4. Convergence curves of ABC, GABC, NABC and CLABC algorithms for f_6

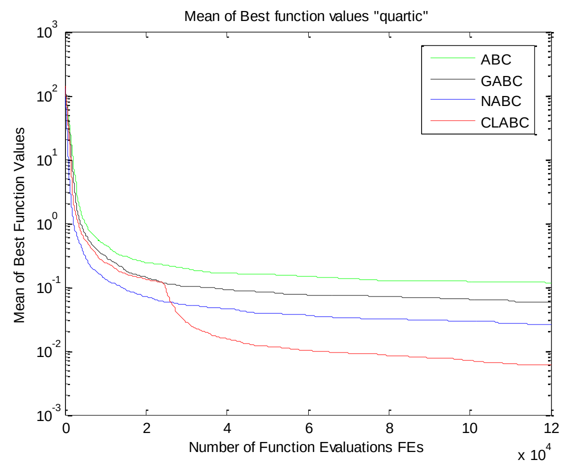

Fig. 5. Convergence curves of ABC, GABC, NABC and CLABC algorithms for f_7

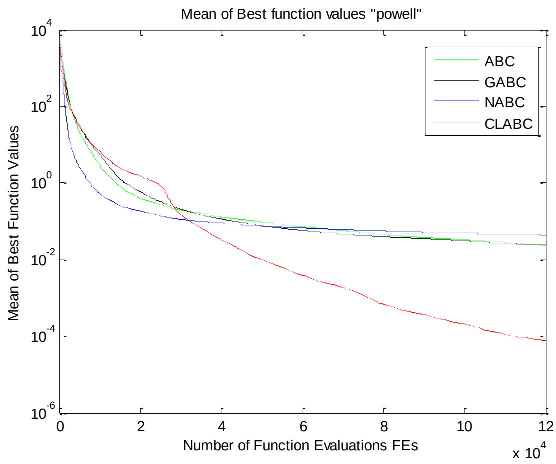

Fig. 6. Convergence curves of ABC, GABC, NABC and CLABC algorithms for f_8

V. CONCLUSION

In this paper, a modified ABC optimization algorithmic framework based on cooperative learning strategy is proposed. The cooperative learning artificial bee colony algorithm (CLABC) involves different search mechanisms at both employed and onlooker levels. The attempt is to enhance exploration and exploitation abilities of ABC optimization model so that better solutions and improved convergence are achieved. Experimental tests on benchmark numerical functions demonstrate the performance of the CLABC when compared to ABC, GABC, and NABC for most of the cases. Applications to complex numerical and engineering problems will be investigated in future works.

REFERENCES

[1] J. Kennedy, and R. C. Eberhart (1995), "Particle swarm optimization," in Proceeding of the IEEE International conference on Neural Networks, Perth, Australia, pp. 1942-1948.

[2] M. Dorigo, T. Stutzle, Ant Colony Optimization. Cambridge, MA: MIT Press, 2004.

[3] K. S. Tang, K. F. Man, S. Kwong, Q. He, "Genetic algorithms and their applications," IEEE Signal Process. Mag., vol. 13, no. 6, pp. 22-37, Nov. 1996.

[4] R. Storn, K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," J. Global Optim., vol. 11, no. 4, pp.341-359, Dec. 1997.

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

[6] D. Karaboga, B. Basturk, "A powerful and efficient algorithm for numerical function optimization : artificial bee colony (ABC) algorithm," Journal of Global Optimization, vol. 39, pp. 459-471, 2007.

[7] D. Karaboga, B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," Applied Soft computing, vol. 8, pp. 687-697, 2008.

[8] H. Habbi, Y. Boudouaoui, D. Karabogo, C. Ozturk, "Self-generated fuzzy systems design using artificial bee colony optimization," Information Sciences, vol. 295, pp.145-159, 2015.

[9] A. Baykasoglu, L. Ozbakir, P. Tapkan, "Artificial bee colony algorithm and its application to generalized assignment problem," Swarm Intelligence: Focus on Ant and Particle Optimization, I-Tech Education and Publishing, Vienna, Austria, 2007, pp. 113-144.

[10] H. Habbi, "Artificial bee colony optimization algorithm for TS-type fuzzy systems learning," in 25th International Conference of European Chapter on Combinatorial Optimization, Antalya,Turkey, 2012, Apr. 26-28.

[11] H. Habbi, Y. Boudouaoui, C. Ozturk, D. Karaboga "Fuzzy rule-based modeling of thermal heat exchanger dynamics through swarm bee colony optimization", International Conference on Advanced Technology and Sciences, ICAT'2015, August 4-7, 2015, Antalya, Turkey

[12] G. Zhu, S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," Applied Mathematics and Computation, vol. 217, pp. 3166-3173, 2010.

[13] Y. Yi, R. He, "A Novel Artificial Bee Colony Algorithm," Sixth International Conference on Intelligent Human-Machine Sytems and Cybernetics, vol.1, pp. 271-274, 2014.

[14] H. Habbi and Y. Boudouaoui, "Hybrid Artificial Bee Colony and Least Squares Method for Rule-Based Systems Learning," Waset. International Journal of Computer, Control, Quantum and Information Engineering, vol. 08, no. 12, pp. 1968-1971, 2014.

[15] D. Karaboga, B. Akay, "A comparative study of Artificial Bee Colony algorithm," Appl Math. Comput., vol. 214, pp. 108-132, Aug. 2009.

[16] A. W. Mohamed, H. Z. Sabry, T. Abd-Elaziz, "Real parameter optimization by an effective differential evolution algorithm," Egyptian Informatics Journal, vol. 14, pp. 37-53, 2013.