

A Novel Artificial Bee Colony Algorithm for Global Optimization

Donya Yazdani

Department of Electrical Engineering, Computer and
Information Technology
Science and Research Branch, Islamic Azad University
Qazvin, Iran
Dny.yazdani@gmail.com

Mohammad Reza Meybodi

Department of Computer Engineering and Information
Technology
Amirkabir University of Technology
Tehran, Iran
Mmeybodi@aut.ac.ir

Abstract— Artificial Bee Colony (ABC) algorithm is a swarm-based optimization algorithm with advantages like simplicity and proper exploration ability. However, it suffers from improper exploitation in solving complicated problems. In order to overcome this disadvantage, modifications on all three bee types are proposed. By introducing a new procedure for the scout bees and modifying the search patterns of both employed and onlooker bees, the capabilities of all three bee types are utilized properly. These modifications lead to better exploitation and exploration abilities. Experiments are conducted on 12 different benchmark functions including standard, shifted, rotated, and shifted-rotated multimodal problems. The results confirm the superiority of the proposed algorithm compared with some other well-known algorithms in this field.

Keywords- Swarm Intelligence; Artificial Bee Colony algorithm; Global Optimization

I. INTRODUCTION

In recent years, the field of natural computing has been the focus of a substantial number of research studies. These algorithms take the inspiration from nature to solve complex problems. Natural computing algorithms have been applied to a wide set of problems in a very different temporal and physical scales; ranging from a very fine scale as in bioinformatics [1, 2] to a very large scale like astronomical studies [3].

Swarm intelligence algorithms, as a subcategory of natural computing algorithms, have proved very effective in solving complicated optimization problems. Particle Swarm Optimization (PSO) [4] and Ant Colony Optimization (ACO) [5] are two of the most famous swarm intelligence algorithms.

Artificial Bee Colony (ABC) algorithm proposed by Karaboga in 2005 is a swarm intelligence algorithm which mimics the foraging behavior of honeybee swarm [6]. It is shown by many researchers that ABC is suitable for solving different computational problems [7]. Despite the advantages [6], the basic ABC suffers from some drawbacks like slow convergence speed and improper exploitation ability in solving complicated problems [11]. In the basic ABC, new solutions are created only by changing the value of one random dimension of a current solution, which leads to slow convergence. In addition, in

the process of creating a new neighbor for both employed and onlooker bees, a completely random position is chosen as the target without considering its fitness value. This insignificance in choosing targets results in wasting fitness evaluations and lowering convergence speed. This problem arises in the case of scout bees too. In the scout bee process, searching space is not adaptive to the current position of the whole swarm. Particularly in the last iterations, sending scout bees to search random places far away from other bees leads to decrease in the number of recruitments. In addition, there is a probability that the best found position becomes abandoned (since this position has no possibility of improvement). Therefore the algorithm loses the best found solution.

In order to develop an ABC based algorithm with improved performance in solving relatively complicated problems, we propose some modifications on the following areas: 1) the whole procedure of scout bees and 2) the process of creating new neighbors for both employed and onlooker bees. First, some modifications are proposed to make scout bees adaptive to the current state of the whole swarm. In the proposed approach, a new formula for calculating the position of scout is introduced which includes the whole swarm position. By using this new formula, unlike the basic scout process, scout bees can be sent for every position that got abandoned with the exception of the best position found so far. The proposed scout process is useful for the exploitation in addition to exploration. Second, instead of going toward a random position, it is proposed for employed and onlooker bees to use a selection mechanism to choose targets more significantly (according to fitness value). By doing so, these bees have more chances in finding better positions.

The rest of this paper is organized as follows. The basic ABC is described in section 2. In section 3, related works are reviewed and section 4 is dedicated to describe the proposed approach called NABC (Novel ABC). In Section 5, experiments are presented and results are discussed. Finally, conclusion is given in section 6.

II. BASIC ARTIFICIAL BEE COLONY ALGORITHM

Artificial Bee Colony (ABC) algorithm is one of the recent swarm-based algorithms. ABC simulates the intelligent foraging behavior of a honeybee swarm. In the ABC algorithm, the honey bee colony model consists of

three kinds of bee: employed, onlooker, and scout. An employed bee searches around the current food source in her memory to find a new place with better nectar value. If the nectar amount of this new position is higher than that of the previous one, the bee save this new position in her memory and forgets the old one. Note that each employed bee is assigned to one food source. After all employed bees complete their search processes, they share their obtained information with onlooker bees which are waiting in the hive. An onlooker bee chooses a food source depending on the probability value associated with that food source (more than one onlooker can be assigned to a food source). 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 the nectar is higher than previous one, the bee saves the new position in her memory and forgets the old one. When the nectar of a food source is abandoned by the bees, a new food source is determined by a scout bee randomly and replaced with the abandoned one. In the basic version of ABC, at each cycle at most one scout goes outside to search for a new food source. The number of employed bees or the onlooker bees is equal to the number of solutions in the swarm. The procedure of basic ABC is described below [6, 8].

At the beginning, ABC generates a randomly distributed initial population of SN solutions (the number of food sources or the swarm size) by (1):

$$x_i^j = x_{min}^j + rand[0,1](x_{max}^j - x_{min}^j) \quad (1)$$

Where $i = 1, \dots, SN$ and $j = 1, \dots, D$. D is dimension or the number of optimization parameters. x_{min} and x_{max} are the parameters boundaries.

After initialization, the population of the food sources (solutions) is subjected to repeat cycles of the search processes of employed bees, onlooker bees and scout bees respectively.

Each employed bee generates a new candidate solution V_i around its current position by (2):

$$v_i^j = x_i^j + \phi_i^j (x_i^j - x_k^j) \quad (2)$$

Where $k \in \{1, 2, \dots, SN\}$ is a randomly chosen index which must be different from i . j is a random integer in the range $[1, D]$, and ϕ_i^j is a uniformly distributed real number in the range $[-1, 1]$.

After employed bees completed their missions, onlooker bees start to work. An onlooker bee chooses a food source with a probability related to its nectar value which is computed by (3):

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

Where fit_i is computed by (4):

$$fit_i = \begin{cases} 1/(1 + fitness_i), & \text{if } (fitness_i \geq 0) \\ 1 + abs(fitness_i), & \text{if } (fitness_i < 0) \end{cases} \quad (4)$$

That $fitness_i$ is the nectar (fitness) value of the solutions. It should be noted that onlooker bees also use (2) to generate new solution candidates.

If a food source is not improving over a predefined number of iterations (which is called *Limit*), then the food source is considered abandoned. In this case, a scout bee is sent to discover a new food source which is going to be replaced with the abandoned one. This new place is produced using (1).

III. RALATED WORKS

Akay and karaboga proposed an ABC based algorithm called MABC to solve real parameter optimization problems [8]. A new control parameter called Modification Rate (MR) which controls the frequency of perturbation and a mechanism called Adaptive Scaling Factor (ASF) to control the magnitude of the perturbation are introduced in this paper. These authors also introduced another ABC based algorithm which uses Deb's rules as selection mechanism for solving constrained optimization problems [9]. Gao and Liu proposed an improved global ABC inspired from Differential Evolution by utilizing a new search pattern and a control parameter [10]. Their proposed algorithm showed high performance in solving complex numerical optimization problems. Zhu and Kwong, inspired from PSO, introduced a g-best guided ABC [11]. This algorithm uses the global best in the search process and shows better exploitation ability than ABC. Banharnsakun, Achalakul, and Sirinaovakul proposed to share the best feasible solutions found so far among the entire swarm [12]. They also proposed a mechanism to adjust the radius of search adaptively.

In addition, there are some research studies about applications of ABC in different areas besides numerical function optimization such as data clustering [13, 14], design of photovoltaic system [15], fuzzy clustering [16], and distribution networks reconfiguration [17].

IV. PROPOSED APPROACH

As mentioned before, the proposed algorithm called NABC includes some modifications on the following areas: 1) the whole procedure of scout bees and 2) the process of creating new neighbors for both employed and onlooker bees.

A. A new scout

The number of scout bees in NABC can be any number in $\{0, 1, \dots, SN-1\}$. It means that every position that is abandoned can be replaced with the new source found by scout bees except the best position found so far. In NABC, instead of sending scout bees to random positions within the entire search space (computed by (1)), it is proposed to send them to positions computed by (5):

$$x_i^j = x_i^j + rand[-1,1](x_{center}^j - x_i^j) \quad (5)$$

That x_{center} is the central position of swarm which is computed at the beginning of each iteration by (6):

$$x_{center}^j = \frac{1}{SN} \sum_{i=1}^{SN} x_i^j \quad (6)$$

According to (5), a scout bee is sent toward the center of swarm or away from it. This movement gives a chance to bees to find better positions, increases diversity of swarm and helps swarm to escape from local optima. By using this new scout process, exploration operation of scout bee transforms into exploitation as time passes. Therefore, the proposed scout procedure helps to set a balance between exploration and exploitation.

B. Modifying the process of creating new neighbors for employed and onlooker bees

In the ABC procedure, k in (2)—the target that a bee is going to search its neighborhood—is chosen randomly without consideration of fitness value. In this paper, it is proposed to use a roulette wheel selection for choosing k so that fitness of the positions that employed and onlooker bees are sent to search their neighborhood is considered. By doing so, these bees have higher chances to find better positions which result in increasing convergence speed and better exploitation ability.

Also, in the basic ABC, only one random dimension is changed in the process of creating neighbor by employed and onlooker bees. In order to increase the convergence speed, a random number of dimensions are chosen to change in each position in both phases of employed and onlooker bees in NABC (both number and indices are random). It should be noted that changing all dimensions may cause premature convergence.

V. EXPERIMENTAL RESULTS

A. Comparison of performance

In order to evaluate the performance of the proposed algorithm, NABC, 12 standard benchmark functions including standard, shifted, rotated, and shifted-rotated multimodal problems are used. The properties of these functions are listed in Table 1.

All experiments are conducted 30 times for each function independently and the results are presented in Table 2. The obtained results are compared with those of global PSO (GPSO) [18], CM-AFSA [19], basic ABC [6] and two new versions of ABC including MABC [8], and GABC [11]. The stop criterion for all algorithms is reaching maximum number of function evaluation which is set to 200,000.

The parameter settings for ABC based algorithms are as follows. Population size (SN) is 10, $limit$ is computed by $SN \times D$ equation (D is dimension), MR is 0.8 in MABC, and C is 1.5 in GABC. It should be noted that all comparisons are based on fitness values directly for all algorithms.

For CM-AFSA, the population size is 50, try_number is 5, $crowd_factor$ is 0.75, $visual$ is $Par[1]$, $step$ is $visual/5$, and $c_1=c_2=2$. For GPSO the population size is set to 20,

$c_1=c_2=2$, $W_{min}=0.4$, $W_{max}=0.9$, and neighborhood topology is global star.

The best results are displayed in bold in Table 2. On F1, except basic ABC, MABC, and GABC the other three algorithms including NABC find the global optimum. GPSO is the only algorithm that reaches to optimum on F2 and the proposed algorithm obtains the second best result. For F3, NABC outperforms the other algorithms. For F4 only NABC and basic ABC are able to find the optimum. NABC is the only algorithm that finds the optimum on F5 while the others are unable to find acceptable solutions. GABC and basic ABC are the only algorithms which reach the global optimum on F6. Obtained result of NABC is the second best for this test function which is significantly better than other three algorithms. For F7, NABC, basic ABC, and GABC achieve a similar result which is the best among those of other algorithms. NABC and MABC are the only algorithms which obtain the optimum on F8. On F9 and F11, NABC outperforms the other algorithms significantly. On F10, NABC, basic ABC, MABC, and GABC achieve a similar result which is the best among other results. Basic ABC is the only algorithm which reaches to the optimum on F12 while NABC and MABC together achieve the second best result. Overall, the proposed algorithm achieves the best results for nine test functions and for other three test functions it obtains the second best results. Some comparisons of performance are shown in Fig. 1 and Fig. 2.

Figure 1. Comparison of performances of all six algorithms for minimization of Weierstrass (F4).

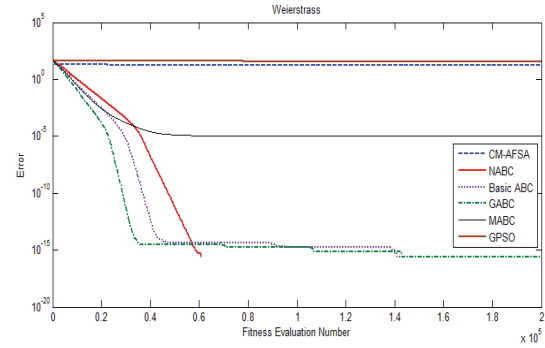
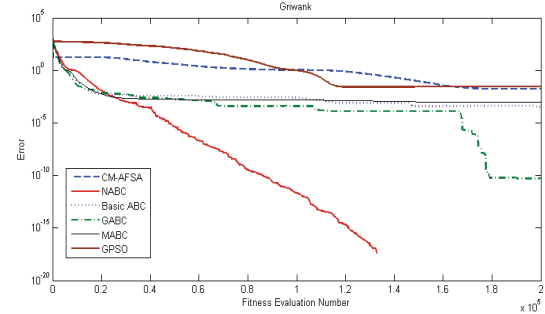


Figure 2. Comparison of performances of all six algorithms for minimization of Griewank (F5).



It should be noted that the minimum number of fitness evaluation among all iterations of NABC was SN and the maximum was $1.3 \times SN$. Compared to basic ABC which has

minimum SN number of fitness evaluation and maximum $1.05 \times SN$ number of fitness evaluation among all iterations, it can be said that the proposed algorithm uses more fitness evaluations in each iteration generally. This is due to the different number of sent scout bees in each iteration which is at most one in basic ABC and at most $SN-1$ in NABC. In addition, since the proposed algorithm uses roulette wheel selection it is more complicated and needs more time than basic ABC. However, the difference is note considerable.

B. NABC with different parameter values

In order to investigate the effect of different values of

SN and $limit$ on the performance of NABC, this algorithm is tested with different settings. Table 3 and Table 4 show the results obtained by NABC with different parameter values on all 12 benchmark functions (all experiments are conducted 30 times independently with the stop criterion of reaching 200,000 fitness evaluations). For any given SN (10, 20, 50), when $limit$ is lesser the obtained results are better for most of functions. The lesser $limit$ leads to sending more scout bees which in turn results in creating more opportunities to find better positions and escaping from local optima. In addition, for any given $limit$ ($0.5 \times SN \times D$, $SN \times D$, $2 \times SN \times D$) the obtained results for most of benchmark functions are better when SN is lesser.

TABLE I. THE STANDARD TEST FUNCTIONS USED, D INDICATES DIMENSION AND f_{\min} IS THE MINIMUM VALUE

Function	Test function	D	Search range	f_{\min}
F1, Bohachevsky	$f_1(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1 + 4\pi x_2) + 0.3$	2	$[-100, 100]^D$	0
F2, Schaffer	$f_2(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	2	$[-100, 100]^D$	0
F3, Ackley	$f_3(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i\right) + 20 + e$	30	$[-32, 32]^D$	0
F4, Weierstrass	$f_4(x) = \sum_{i=1}^D \left(\sum_{k=0}^{20} (0.5^k \cos(2\pi 3^k (x_i + 0.5))) \right) - D \sum_{k=0}^{20} (0.5^k \cos(2\pi 3^k \times 0.5))$	30	$[-0.5, 0.5]^D$	0
F5, Griwank	$f_5(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^D$	0
F6, Rastrigin	$f_6(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	30	$[-5.12, 5.12]^D$	0
F7, Penalized	$f_7(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 (1 + 10 \sin^2(\pi y_{i+1})) + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	30	$[-50, 50]^D$	0
F8, Shifted-Rotated Ackly	$f_8(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi z_i\right) + 20 + e + bias_8$ $z = (x - o^{**}) \times M^*, bias_8 = -140$	30	$[-32, 32]^D$	-140
F9, Rotated Penalized	$f_9(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (w_i - 1)^2 (1 + 10 \sin^2(\pi y_{i+1})) + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(w_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(w_i + 1)$ $u(w_i, a, k, m) = \begin{cases} k(w_i - a)^m, & w_i > a \\ 0, & -a \leq w_i \leq a \\ k(-w_i - a)^m, & w_i < -a \end{cases}$ $w = x \times M^*$	30	$[-50, 50]^D$	0
F10, Shifted Griwank	$f_{10}(x) = \frac{1}{4000} \sum_{i=1}^D z_i^2 - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + bias_{10}$ $z = x - o^{**}, bias_{10} = -180$	30	$[-600, 600]^D$	-180
F11, Shifted-Rotated Rastrigin	$f_{11}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + bias_{11}$ $z = (x - o^{**}) \times M^*, bias_{11} = -330$	30	$[-5.12, 5.12]^D$	-330
F12, Shifted-Rotated Griwank	$f_{12}(x) = \frac{1}{4000} \sum_{i=1}^D z_i^2 - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + bias_{12}$ $z = (x - o^{**}) \times M^*, bias_{12} = -180$	30	$[-600, 600]^D$	-180

* M is an orthogonal matrix, ** $o = [o_1, o_2, \dots, o_D]$ is the shifted global optimum

TABLE II. OBTAINED RESULTS OF ALL ALGORITHMS

Function	Criteria	GPSO	Basic ABC	CM-AFSA	MABC	GABC	NABC
F1	Mean (Std)	0 (0)	1.69e-006 (2.29e-006)	0 (0)	1.34e-007 (4.24e-007)	3.07e-008 (3.05e-008)	0 (0)
F2	Mean (Std)	0 (0)	0.0002 (0.0003)	9.79e-5 (0.0004)	0.0044 (0.0035)	6.48e-005 (0.0001)	2.17e-014 (1.19e-013)
F3	Mean (Std)	0.51 (2.80)	5.98e-015 (1.29e-015)	1.57 (1.06)	2.66e-015 (0)	3.61e-015 (1.59e-015)	-8.88e-016 (0)
F4	Mean (Std)	35.91 (3.37)	0 (0)	16.08 (3.98)	1.07e-005 (2.69e-005)	2.36e-016 (1.29e-015)	0 (0)
F5	Mean (Std)	0.02 (0.03)	0.0003 (0.0018)	0.07 (0.04)	0.0009 (0.0049)	4.77e-011 (2.48e-010)	0 (0)
F6	Mean (Std)	57.06 (20.93)	0 (0)	129.47 (34.45)	155.01 (18.03)	0 (0)	5.60e-014 (2.87e-013)
F7	Mean (Std)	0.09 (0.22)	1.57e-032 (5.56e-048)	2.07 (1.83)	0.17 (0.48)	1.57e-032 (5.56e-048)	1.57e-032 (5.56e-048)
F8	Mean (Std)	-119.71 (1.59)	-120.30 (0.4096)	-137.26 (4.44)	-140 (9.44e-014)	-138.23 (0.6587)	-140 (4.01e-014)
F9	Mean (Std)	236.48 (2813.94)	6.38 (1.29)	4.26 (2.83)	0.70 (0.57)	2.96 (1.0014)	1.36e-009 (7.45e-009)
F10	Mean (Std)	-54.95 (100.17)	-179.99 (0.0013)	-179.90 (0.07)	-179.99 (7.88e-005)	-179.99 (0.0028)	-179.99 (0.0013)
F11	Mean (Std)	50.08 (47.03)	-8.54 (39.95)	-160.04 (40.39)	-91.78 (19.01)	-167.04 (25.27)	-248.61 (20.41)
F12	Mean (Std)	-87.59 (78.02)	-180 (2.68e-013)	-179.91 (0.06)	-179.99 (0.0013)	-179.98 (0.06)	-179.99 (0.0030)

TABLE III. MEAN AND STD OF OBTAINED RESULTS BY NABC WITH DIFFERENT SN AND $LIMIT$ ON FUNCTIONS 1 TO 6

Parameters	F1	F2	F3	F4	F5	F6
$SN=10$ $Limit=0.5 \times SN \times D$	0 (0)	1.61e-08 (3.25e-08)	-8.88e-16 (0)	0 (0)	0 (0)	4.14e-16 (2.27e-15)
$SN=10$ $Limit=2 \times SN \times D$	0 (0)	0 (0)	-8.88e-16 (0)	0 (0)	0 (0)	7.88e-10 (3.36e-09)
$SN=20$ $Limit=0.5 \times SN \times D$	0 (0)	3.40e-008 (5.83e-008)	-8.88e-016 (0)	0 (0)	2.96e-017 (1.62e-016)	17.5853 (4.5805)
$SN=20$ $Limit=SN \times D$	0 (0)	3.59e-013 (1.40e-012)	-8.88e-016 (0)	0 (0)	2.59e-017 (1.41e-016)	13.1836 (2.1699)
$SN=20$ $Limit=2 \times SN \times D$	0 (0)	0 (0)	-8.88e-016 (0)	0 (0)	1.85e-017 (1.01e-016)	13.8140 (2.24)
$SN=50$ $Limit=0.5 \times SN \times D$	0 (0)	7.55e-10 (4.02e-09)	-7.68e-16 (6.49e-16)	2.38e-06 (1.28e-06)	9.15e-07 (1.95e-06)	38.14799 (3.8906)
$SN=50$ $Limit=SN \times D$	0 (0)	0 (0)	-8.8818e-16 (0)	2.67e-06 (1.52e-06)	8.85e-07 (1.26e-06)	37.3764 (4.2506)
$SN=50$ $Limit=2 \times SN \times D$	0 (0)	0 (0)	2.19e-015 (1.22e-015)	2.93e-006 (1.57e-006)	2.26e-006 (3.82e-006)	37.7582 (4.2445)

TABLE IV. MEAN AND STD OF OBTAINED RESULTS BY NABC WITH DIFFERENT SN AND $LIMIT$ ON FUNCTIONS 7 TO 12

Parameters	F7	F8	F9	F10	F11	F12
$SN=10$ $Limit=0.5 \times SN \times D$	1.57e-32 (5.57e-48)	-140 (9.14e-15)	4.37e-12 (1.76e-11)	-180 (7.74e-10)	-249.497 (17.32183)	-180 (0.002345)
$SN=10$ $Limit=2 \times SN \times D$	1.57e-32 (5.56e-48)	-140 (9.67e-09)	0.0119 (0.0322)	-180 (5.62e-08)	-245.5914 (17.0893)	-179.9987 (0.0034)
$SN=20$ $Limit=0.5 \times SN \times D$	3.52e-032 (2.86e-032)	-139.9999 (6.77e-009)	0.0008 (0.0016)	-180 (3.73e-014)	-232.6889 (18.8975)	-179.999 (1.81e-005)
$SN=20$ $Limit=SN \times D$	4.47e-032 (4.65e-032)	-139.9999 (6.17e-005)	0.2254 (0.2148)	-180 (3.842e-014)	-219.4263 (16.1005)	-179.9999 (0.0004)
$SN=20$ $Limit=2 \times SN \times D$	4.53e-032 (5.27e-032)	-139.9998 (0.0006)	0.6967 (0.4285)	-180 (3.53e-013)	-216.3042 (19.6395)	-179.9999 (7.69e-005)
$SN=50$ $Limit=0.5 \times SN \times D$	1.61e-11 (1.54e-11)	-139.778 (0.2115)	0.7246 (0.3079)	-180 (8.19e-06)	-187.659 (16.3368)	-179.994 (0.0031)
$SN=50$ $Limit=SN \times D$	2.21e-11 (2.01e-11)	-139.3654 (0.4616)	0.7979 (0.3723)	-179.9999 (1.94e-05)	-192.6711 (14.1426)	-179.9949 (0.0044)
$SN=50$ $Limit=2 \times SN \times D$	3.21e-011 (3.67e-011)	-138.9461 (0.7668)	0.7603 (0.3712)	-179.9999 (1.00e-005)	-193.0792 (16.0071)	-179.9934 (0.0045)

It is observed from Table 3 and Table 4 that the performance of NABC algorithm with $SN=10$ and $limit=0.5 \times SN \times D$ is superior to the other settings. Since basic ABC, MABC and GABC obtained the best results with $SN=10$ and $limit=SN \times D$ in several experiments, the same value for $limit$ is chosen for NABC in Table 2 (which is dedicated to compare the obtained results of all six algorithms) to have the same conditions as others. However, even without the optimum settings, NABC shows superiority.

VI. CONCLUSION

In this paper, a new ABC algorithm called NABC was proposed to solve complicated optimization problems. In NABC, the scout bee process is changed completely and a selection mechanism (roulette wheel) is used for choosing targets in employed and onlooker bees. In addition, the number of dimensions to be changed in the process of creating neighbors is modified. In the proposed scout bee procedure, scout bees are sent toward the center of the swarm

or away from it for every abandoned position except the best position found so far. By doing so, scout bees are able to bring balanced diversity to the swarm (unlike the standard scout bee which brings unbalanced diversity) that leads to higher chances of finding better positions and skipping local optima. Also, by making onlooker and employed bees to choose targets more significantly they have more opportunities in finding better neighbors. Moreover, a random number of dimensions are chosen to be changed in the process of creating neighbors which help the algorithm to converge faster. By these modifications, performance of ABC in solving complex problems is improved. However, due to using roulette wheel selection, NABC needs more time than basic ABC which is not significant. The proposed approach without introducing any new control parameter and with the advantage of simple implementation is able to set a proper balance between exploitation and exploration. The experimental results show that compared with some other well-known algorithms in this field, the proposed algorithm has better performance on complicated functions.

REFERENCES

- [1] A. Ghaffarizadeh, M. Eftekhari, A. Esmailizadeh, and N. Flann, "Quantitative trait loci mapping problem: An Extinction-Based Multi-Objective evolutionary algorithm approach," in *Algorithms*, vol. 6, pp. 546-564, 2013.
- [2] A. Ghaffarizadeh, K. Ahmadi, and N. S. Flann, "Sorting unsigned permutations by reversals using multi-objective evolutionary algorithms with variable size individuals," in *Evolutionary Computation*, pp. 292-295, 2011.
- [3] E. Cantu-Paz, C. Kamath, and R. Kamath, "Combining evolutionary algorithms with oblique decision trees to detect Bent-Double galaxies," in *International Symposium on Optical Science and Technology (SPIE Annual meeting)*, 2000.
- [4] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.
- [5] M. Dorigo, "Learning and natural algorithms," Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.
- [6] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Erciyes University, Engineering Faculty, Computer Engineering Department, Tech. Rep. TR06, pp. 1-10, 2005.
- [7] F. S. Abu-Mouti and M. E. El-Hawary, "Overview of artificial bee colony (ABC) algorithm and its applications," in *Systems Conference (SysCon)*, pp. 1-6, 2012.
- [8] B. Akay and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization," in *Swarm Intelligence and Its Applications*, vol. 192, pp. 120-142, 2012.
- [9] D. Karaboga and B. Akay, "A modified artificial bee colony (ABC) algorithm for constrained optimization problems," in *Applied Soft Computing*, vol. 11, pp. 3021-3031, 2011.
- [10] W. Gao and S. Liu, "Improved artificial bee colony algorithm for global optimization," in *Information Processing Letters*, vol. 111, pp. 871-882, 2011.
- [11] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," in *Applied Mathematics and Computation*, vol. 217, pp. 3166-3173, 2010.
- [12] A. Banharsakun, T. Achalakul, and B. Sirinaovakul, "The best-so-far selection in artificial bee colony algorithm," in *Applied Soft Computing Journal*, vol. 11, pp. 2888-2901, 2011.
- [13] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial bee colony (ABC) algorithm," in *Applied Soft Computing Journal*, vol. 11, pp. 652-657, 2011.
- [14] C. Zhang, D. Ouyang, and J. Ning, "An artificial bee colony approach for clustering," in *Expert Systems with Applications*, vol. 37, pp. 4761-4767, 2010.
- [15] A. F. Mohamed, M. M. Elarini, and A. M. Othman, "A new technique based on artificial bee colony algorithm for optimal sizing of stand-alone photovoltaic system," in *Journal of Advanced Research*, vol. 5, pp. 397-408, 2014.
- [16] D. Karaboga and C. Ozturk, "Fuzzy clustering with artificial bee colony algorithm," in *Scientific Research and Essays*, vol. 5, pp. 1899-1902, 2010.
- [17] N. T. Linh and N. Q. Anh, "Application of artificial bee colony algorithm (ABC) for reconfiguring distribution network," in *Computer Modeling and Simulation*, pp. 102-106, 2010.
- [18] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *IEEE World Congress on Computational Intelligence*, pp. 69-73, 1998.
- [19] H.C. Tsai and Y. H. Lin, "Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior," in *Applied Soft Computing*, vol. 11, pp. 5367-5374, 2011.