IJE TRANSACTIONS C: Aspects Vol. 28, No. 9, (September 2015) 1268-1275

International Journal of Engineering

Journal Homepage: www.ije.ir

Elite Opposition-based Artificial Bee Colony Algorithm for Global Optimization

Z. Guo*[a], S. Wang[b], X. Yue[a], D. Jiang[c], K. Li[d]

[a] *School of Science, JiangXi University of Science and Technology, Ganzhou, China*
[b] *School of Information Engineering, Shijiazhuang University of Economics, Shijiazhuang, China*
[c] *State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan, China*
[d] *School of Mathematics and Informatics, South China Agricultural University, Guangzhou, China*

PAPER INFO

Paper history:
Received 21 February 2015
Received in revised form 29 July 2015
Accepted 03 September 2015

Keywords:
Evolutionary Algorithm
Artificial Bee Colony
Opposition-based Learning
Elite Strategy

ABSTRACT

Numerous problems in engineering and science can be transformed into optimization problems. Artificial bee colony (ABC) algorithm is a newly developed stochastic optimization algorithm and has been successfully used in many areas. However, due to the stochastic characteristics of the solution search equation, the traditional ABC algorithm often suffers from poor exploitation. Aiming at this weakness of the traditional ABC algorithm, in this paper, we propose an enhanced ABC algorithm with elite opposition-based learning strategy (EOABC). In the proposed EOABC, it executes the elite opposition-based learning strategy with a preset learning probability to enhance the exploitation capacity. In the experiments, EOABC is tested on a set of numerical benchmark test functions, and is compared with some other ABC algorithms. The comparisons indicate that EOABC can obtain competitive results on the majority of the test functions.

***doi**: 10.5829/idosi.ije.2015.28.09c.03*

1. INTRODUCTION

Optimization problems widely exist in engineering applications [1]. Therefore, it is of significance to develop effective and efficient optimization algorithms for practical problems [2-4]. Evolutionary algorithm (EA) is a very promising approach for optimization algorithms, which has been successfully applied to many practical applications [5-8]. Artificial bee colony algorithm (ABC), recently developed by Karaboga and Basturk [9], is a kind of EA that mimics the foraging behavior of the honey bee swarm in nature. Like other EAs [10, 11], ABC has a very simple structure. As ABC is easy to implement and has exhibited encouraging performance in many problems from various fields, it has been quickly developed in recent years and has been successfully applied in solving diverse real-world optimization problems [12]. In practice, ABC has been compared with many other optimization techniques such as Genetic Algorithm (GA) [13], Particle Swarm Optimization (PSO) [14, 15] and Differential Evolution (DE) [16]. The comparison results reveal that the performance of ABC can outperform other optimization techniques in many optimization problems [12].

Although ABC has successfully solved a wide variety of optimization problems from various areas, it has several weaknesses when solving complex optimization problems. One of the weaknesses of the traditional ABC is that it may suffer from poor exploitation when solving complex optimization problems [12]. As known, both exploration and exploitation are very critical for EAs [17]. Therefore, in some cases, the traditional ABC cannot find satisfactory results. Accordingly, various ABC variations have been proposed to promote the search capability of the traditional ABC. In order to enhance the exploitation ability, Zhu et al. [18] proposed a gbest-guided ABC (GABC), which combines the information of the global best solution into the solution search equation. Gao and Liu [19] presented an improved ABC with two improved solution search equations which are inherited from DE. El-Abd [20] introduced an opposition-based ABC (OABC), which incorporates the opposition-based learning strategy. Banharnsakun et al. [21] proposed an

*Corresponding Author's Email: gzl@whu.edu.cn (Zhaolu Guo)

improved ABC with best-so-far selection. In the best-so-far ABC, all the solutions in the current population share the best feasible solutions found so far. Akay and Karaboga [22] presented a modification on ABC by introducing the perturbation frequency and the magnitude of the perturbation. To improve the exploitation ability, Gao and Liu [23] introduced an improved ABC, which utilizes the best solution of the previous iteration.

Karaboga et al. [24] presented a symbolic regression approach using ABC programming (ABCP). By incorporating the generalized opposition-based learning strategy, El-Abd [25] proposed an enhance ABC (GOABC) for global optimization. To improve the exploration power, Chen et al. [26] introduced the simulated annealing (SA) algorithm into ABC. Luo et al. [27] enhanced the solution search equation in the onlooker phase using the best solution of the previous generation to direct the search process. Gao et al. [28] proposed a modified ABC with improved search equation and the orthogonal learning (OL) strategy. Xiang and An [29] introduced an efficient and robust ABC (ERABC), which uses a combinatorial solution search equation to accelerate the convergence speed. Kang et al. [30] proposed a hybridized ABC, which embeds the pattern search scheme. Bansal et al. [31] presented a modified ABC with memetic search strategy. Karaboga and Gorkemli [32] proposed a quick ABC, which introduces the neighborhood radius into the search step of the onlooker bees. Gao et al. [33] presented an enhanced ABC utilizing more information-based search equations. Xiang et al. [34] introduced a hybrid ABC embedding the search operation of DE. Kıran and Fındık [35] proposed a directional ABC (dABC) which employs the directional information to improve the search ability of ABC.

In our previous work [36], we proposed an elite opposition-based learning (EOBL) strategy and employed it to enhance the exploitation ability of the traditional DE. Our previous experimental results [36] indicated that EOBL can significantly promote the exploitation ability of the traditional DE. Therefore, it is expected that the exploitation capability of the traditional ABC can also be enhanced by EOBL. Motivated by these considerations, we propose an enhanced ABC (EOABC) through the utilization of EOBL. In the proposed EOABC, at each generation, the EOBL strategy is performed with a preset learning probability to enhance the exploitation ability. In addition, EOABC has a very simple framework and thus is easy to implement.

The rest of the paper is structured as follows. Section 2 describes the traditional ABC algorithm. The elite opposition-based ABC is presented in section 3. Numerical results and comparisons are reported in section 4. Finally, section 5 concludes the paper.

2. ARTIFICIAL BEE COLONY

ABC is a newly developed meta-heuristic algorithm, which simulates the intelligent foraging behavior of the honey bee swarm in nature [9, 12]. In ABC, three kinds of bees, namely employed bees, onlooker bees and scout bees, are used to seek the global optimal solutions for a optimization problem at hand [12]. Moreover, each employed bee is associated with a solution, and the employed bees aim to exploit its associated solution and gather the information of the exploited regions to the onlooker bees. The onlooker bees focus on selecting the excellent solutions to be further exploited through the information provided by the employed bees. In the search process, if the quality of a solution has not been enhanced through a preset number of cycles, this solution is assumed to be abandoned by its employed bee, and then the associated employed bee becomes a scout bee that starts to generate a new solution by randomly sampling from the feasible search space.

Without loss of generality, we suppose in this study that ABC is for solving the minimization problem Min $f(\boldsymbol{X})$, where $\boldsymbol{X}=[x_1, x_2, \ldots, x_D]$, and the search space is:

$$\Omega = \prod_{j=1}^{D}[LB_j, UB_j] \tag{1}$$

where D is the dimension of the minimization problem, $\boldsymbol{LB}_j$ and $\boldsymbol{UB}_j$ are the lower and upper boundaries of the search space, respectively.

Like other EAs, ABC also consists of a very simple procedure. At the initialization stage, an initial population $\boldsymbol{P}(t)=\{\boldsymbol{X}^t_i\}$ is randomly generated from the domain of the minimization problem:

$$x^t_{i,j} = \boldsymbol{LB_j} + \text{rand}(0,1)\times(\boldsymbol{UB_j} - \boldsymbol{LB_j}) \tag{2}$$

where $\boldsymbol{X}^t_i=[x^t_{i,1}, x^t_{i,2}, \ldots, x^t_{i,j} \ldots, x^t_{i,D}]$, i=1, 2, …, SN; j=1, 2, …, D; t represents the generation, and SN is the number of solutions in the population, which is also equal to the size of the employed bees or onlooker bees [9], and rand(0,1) is a random real number in the range [0, 1].

Following the initialization stage, ABC executes a loop of search operations until the termination condition is satisfied. In loop of the search operations, each employed bee creates a neighborhood solution of its associated solution according to the following equation [9]:

$$v^t_{i,j} = x^t_{i,j} + \phi_{i,j} \cdot (x^t_{i,j} - x^t_{i,j}) \tag{3}$$

where j is an integer randomly selected from the set {1, 2, …, D}, index k is an integer randomly selected from the set {1, 2, …, SN}\{i}, and $\phi_{i,j}$ is a random real number uniformly distributed within [-1, 1] [12]. After creating the neighborhood solution V^t_i, its fitness value Fit^t_i is evaluated by [9]:

$$Fit_i^t = \begin{cases} \dfrac{1}{1+f(V_i^t)} & \text{if } f(V_i^t) \geq 0 \\ 1+abs(f(V_i^t)) & \text{if } f(V_i^t) < 0 \end{cases} \tag{4}$$

where $f(.)$ is the objective function of the minimization problem. Then, each employed bee selects the better one between its associated solution and the corresponding created neighborhood solution to enter the next generation [12]. Once all employed bees have finished their search operations, they will provide the information of the exploited solutions to the onlooker bees. After that, each onlooker bee randomly selects a solution for further exploitation according to the selection probability of each solution. Moreover, the selection probability of each solution is calculated by [9]:

$$p_i = \frac{Fit_i^t}{\sum_{i=1}^{SN} Fit_i^t} \tag{5}$$

Subsequently, each selected solution is further exploited by the corresponding onlooker bee using Equation (3). Each onlooker bee also utilizes the greedy selection scheme to select the better one between its selected solution and exploited solution to enter the next generation.

After all onlooker bees complete their search procedure, ABC finds the solution whose quality cannot be improved through a preset number of cycles, called limit, then the corresponding employed bee associated with the found solution becomes a scout bee which will reinitialize its associated solution by randomly sampling from the feasible search space.

3. PROPOSED ALGORITHM

In this section, we propose an enhanced ABC algorithm with elite opposition-based learning strategy (EOABC). First, we introduce the notations of the traditional opposition-based learning strategy. Then, the elite opposition-based learning strategy is presented. Finally, the detailed computation steps of EOABC are elaborated at the end of this section.

3. 1. Opposition-based Learning Opposition-based learning (OBL) strategy is a relatively new intelligent computation technique, which is firstly proposed by Tizhoosh [37]. Since its introduction, OBL has attracted many researchers in recent years and has been successfully incorporated in several EAs [38]. According to extensively reported theoretical and experimental studies, OBL is an effective approach for enhancing the performance of EAs [39]. As known, the opposition concept can be widely observed in real-life, such as opposite particles in physics, and opposition parties in politics [40]. Inspired by the opposition concept in real-life, the opposition idea is introduced into the evolutionary computation fields. The definition of opposite solution is described as follows.

Let $\boldsymbol{X}_i^t=[x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t]$ represent the ith solution in the current population at generation t. Its corresponding opposite solution $\boldsymbol{O}_i^t=[o_{i,1}^t, o_{i,2}^t, \ldots, o_{i,D}^t]$ is calculated by [38]:

$$\begin{aligned} & o_{i,j}^t = A_j^t + B_j^t - x_{i,j}^t \\ & A_j^t = \min(x_{i,j}^t), B_j^t = \max(x_{i,j}^t) \\ & o_{i,j}^t = rand(A_j^t, B_j^t), \text{ if } o_{i,j}^t < LB_i \,\|\, o_{i,j}^t > UB_i \\ & i = 1,2,\ldots,SN,\; j = 1,2,\ldots,D \end{aligned} \tag{6}$$

where $x_{i,j}^t$ is the jth component of the ith solution in the current population, $o_{i,j}^t$ is the opposite value of $x_{i,j}^t$, $\boldsymbol{A}_j^t$ and $\boldsymbol{B}_j^t$ are the minimum and maximum values of the jth dimension of the current population at generation t, respectively.

As known, due to lack of priori information about the optimum solution, many EAs utilize random guess strategies to generate solutions. However, random guess strategies are often time-consuming and inefficient. In fact, it is a potential scheme to increase the chance of finding a better solution by simultaneously considering the opposite solution. In terms of the probability theory, 50% of the time, a guess is further from the solution than its opposite guess [38]. Therefore, simultaneously considering each solution and its corresponding opposite solution can accelerate the convergence speed [38]. Based on the fundamental idea of OBL, Wang et al. have presented a generalized opposition-based learning strategy (GOBL), which introduces a random generalized coefficient to enhance the search capability of OBL [39]. Accordingly, many researchers have employed GOBL to enhance the performance of EAs [39]. The concept of GOBL is expressed as follows.

For solution $\boldsymbol{X}_i^t$, its generalized opposition-based solution $\boldsymbol{GO}_i^t=[go_{i,1}^t, go_{i,2}^t, \ldots, go_{i,D}^t]$ is defined by [39]:

$$\begin{aligned} & go_{i,j}^t = k \cdot (A_j^t + B_j^t) - x_{i,j}^t \\ & A_j^t = \min(x_{i,j}^t), B_j^t = \max(x_{i,j}^t) \\ & go_{i,j}^t = rand(A_j^t, B_j^t), \text{ if } go_{i,j}^t < LB_i \,\|\, go_{i,j}^t > UB_i \\ & i = 1,2,\ldots,SN;\; j = 1,2,\ldots,D;\; k = \mathrm{rand}(0,1) \end{aligned} \tag{7}$$

where $go_{i,j}^t$ is the generalized opposition-based value of $x_{i,j}^t$, and k is the random generalized opposition-based coefficient, which is newly generated for each i.

3. 2. Elite Opposition-based Learning Based on the idea of GOBL, in our previous work [36], we have extended the generalized opposition-based strategy, and proposed an elite opposition-based learning (EOBL) strategy. In EOBL, EN elite solutions are firstly selected from the current population, and then the beneficial information is extracted from these selected elite solutions. The elite opposition-based solutions can be obtained by taking advantage of the extracted beneficial information. By using this manner, EOBL can further enhance the search ability of GOBL.

For solution $\boldsymbol{X}^t_i$, its elite opposition-based solution $\boldsymbol{EO}^t_i=[eo^t_{i,1}, eo^t_{i,2}, \ldots, eo^t_{i,D}]$ is defined by [36]:

$$\begin{aligned}
& eo^t_{i,j} = k\cdot(A^t_j + B^t_j) - x^t_{i,j} \\
& EA^t_j = \min(ex^t_{m,j}),\ EB^t_j = \max(ex^t_{m,j}) \\
& eo^t_{i,j} = rand(EA^t_j, EB^t_j),\ \text{ if }\ eo^t_{i,j} < LB_i \,\|\, eo^t_{i,j} > UB_i \\
& i = 1,2,\ldots,SN;\ j = 1,2,\ldots,D; \\
& m = 1,2,\ldots EN;\ k = \mathrm{rand}(0,1)
\end{aligned} \tag{8}$$

where $\boldsymbol{EX}^t_m=[ex^t_{m,1}, ex^t_{m,2}, \ldots, ex^t_{m,D}]$, m=1, 2, …, EN are the selected elite solutions used to extract the beneficial information, $eo^t_{i,j}$ is the elite opposition-based value of $x^t_{i,j}$, EN is the size of the selected elite solutions, which is set to $SN\times 0.1$, as recommended in our previous work [36] and $\boldsymbol{EA}^t_j$ and $\boldsymbol{EB}^t_j$ are the minimum and maximum values of the jth dimension of the selected elite solutions, respectively.

TABLE 1. The 15 benchmark test functions

Function	Name	Initial Range	f_{min}
f1	Sphere Problem	$[-100, 100]^D$	0
f2	Schwefel's Problem 2.22	$[-10, 10]^D$	0
f3	Schwefel's Problem 1.2	$[-100, 100]^D$	0
f4	Schwefel's Problem 2.21	$[-100, 100]^D$	0
f5	Rosenbrock's Function	$[-30, 30]^D$	0
f6	Step Function	$[-100, 100]^D$	0
f7	Quartic Function with Noise	$[-1.28, 1.28]^D$	0
f8	Schwefel's Problem 2.26	$[-500, 500]^D$	0
f9	Rastrigin's Function	$[-5.12, 5.12]^D$	0
f10	Ackley's Function	$[-32, 32]^D$	0
f11	Griewank Function	$[-600, 600]^D$	0
f12	Penalized Function 1	$[-50, 50]^D$	0
f13	Penalized Function 2	$[-50, 50]^D$	0
f14	Noncontinuous Rastrigin's Function	$[-5.12, 5.12]^D$	0
f15	Alpine Function	$[-10, 10]^D$	0

TABLE 2. Experimental results of ABC, OABC, GOABC, dABC and EOABC over 30 independent runs for the 15 benchmark test functions.

Function	Mean±SD				
	ABC	OABC	GOABC	dABC	EOABC
f1	1.60E-46±9.13E-47+	2.02E-38±1.28E-38+	2.65E-61±2.79E-61+	6.14E-69±4.63E-69+	**3.95E-110±5.59E-110**
f2	5.35E-25±3.33E-25+	5.43E-21±2.40E-21+	1.31E-33±9.19E-34+	9.22E-36±6.73E-36+	**1.06E-58±1.49E-58**
f3	3.30E+03±8.26E+02+	4.10E+03±8.38E+02+	2.43E+03±5.45E+02+	2.96E+03±5.03E+02+	**1.64E-82±2.32E-82**
f4	6.93E+00±1.19E+00+	5.54E+00±4.84E-01+	6.10E-04±3.44E-04+	6.75E+00±6.15E-01+	**1.33E-48±9.62E-49**
f5	1.35E-02±8.26E-03+	9.21E-01±8.51E-01+	2.64E+01±9.18E-02+	1.28E-01±8.72E-02+	**8.43E-03±4.99E-03**
f6	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00**
f7	1.09E-01±1.28E-02+	3.34E-02±4.14E-03+	1.11E-02±3.11E-03+	1.01E-01±2.11E-02+	**4.85E-04±2.20E-04**
f8	1.82E-12±0.00E+00+	1.82E-12±0.00E+00+	1.82E-12±0.00E+00+	1.82E-12±0.00E+00+	**1.21E-12±8.57E-13**
f9	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00**
f10	3.72E-14±3.35E-15+	2.65E-14±3.35E-15+	2.77E-14±1.67E-15+	3.83E-14±4.43E-15+	**4.44E-16±0.00E+00**
f11	1.48E-16±5.23E-17+	4.44E-16±6.28E-16+	4.92E-03±6.96E-03+	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00**
f12	**1.57E-32±0.00E+00≈**	**1.57E-32±0.00E+00≈**	**1.57E-32±0.00E+00≈**	**1.57E-32±0.00E+00≈**	**1.57E-32±0.00E+00**
f13	**1.35E-32±0.00E+00≈**	**1.35E-32±0.00E+00≈**	**1.35E-32±0.00E+00≈**	**1.35E-32±0.00E+00≈**	**1.35E-32±0.00E+00**
f14	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00**
f15	1.57E-11±1.93E-11+	3.64E-12±4.37E-12+	7.73E-12±7.72E-12+	1.45E-12±8.58E-13+	**3.28E-13±3.70E-13**
-	0	0	0	0	
+	10	10	10	9	
≈	5	5	5	6	

Algorithm 1. The algorithmic description of EOABC

$t = 0$;
$FEs = 0$;
Initialize the population;
while $FEs < MAX\_FEs$ **do**
  Pr = rand(0, 1);
  if $Pr < Pe$ **then**
    Choose EN elite solutions from the current population;
    Calculate the lower and upper boundaries of the chosen elite solutions;
    $\boldsymbol{EOP} = \{\}$;
    for $i = 1$ to SN **do**
      k = rand(0, 1);
      Create the elite opposition-based solution $\boldsymbol{EO}^t_i$ for the ith solution $\boldsymbol{X}^t_i$;
      Evaluate solution $\boldsymbol{EO}^t_i$;
      $EOP = EOP \cup \{EO^t_i\}$;
      $FEs = FEs + 1$;
    end for
    Choose the top best SN solutions from $\{\boldsymbol{P}, \boldsymbol{EOP}\}$ for the next generation population;
  else
    Execute the computation procedure of the traditional ABC;
  end if
  $t = t + 1$;
end while

3. 3. EOABC The basic ABC is good at exploration but poor at exploitation, which often results in slow convergence when solving complicated practical problems [12]. Aiming at this weakness of the basic ABC, the EOBL strategy is utilized to improve the exploitation ability. In the search process, the distribution information of the elite solutions in the current population is used to create the opposition-based solution of each solution. Therefore, the EOBL strategy can guide the search towards the promising area and thus improve the exploitation ability.

Like EOBL embedded in DE [36], EOABC has the similar framework. EOABC starts with a random initial population. After initialization, it executes a loop of search process. In the loop of search process, a random real number Pr is generated. If Pr is less than the EOBL probability Pe, EOABC performs the steps of EOBL; Otherwise, it executes the computation procedure of the traditional ABC. In the steps of EOBL, EN elite solutions are firstly chosen from the current population. Then, the lower and upper boundaries of these chosen elite solutions are calculated. Subsequently, the elite opposition-based solution of each solution in the current population is created to constitute an elite opposition-based population. Finally, the elite opposition-based population is competed with the current population to choose the top best SN solutions for the next generation. The algorithmic description of EOABC is shown in Algorithm 1, where FEs is the number of fitness evaluations, Max_FEs is the maximum number of evaluations, Pe denotes the EOBL probability, $\boldsymbol{P}(t)=\{\boldsymbol{X}^t_i\}$ is the current population, and $\boldsymbol{EOP}(t)=\{\boldsymbol{EO}^t_i\}$ is the elite opposition-based population.

4. NUMERICAL EXPERIMENTS

4. 1. Experimental Settings In order to evaluate the effectiveness of the proposed EOABC, 15 classical benchmark test functions widely used in the evolutionary computation community are employed in the experiments [41, 42], which are described in Table 1. The dimension of these 15 benchmark test functions are set to D=30. In the experiments, the proposed EOABC is compared with ABC [9], OABC [20], GOABC [25], and dABC [35]. For a fair comparison, the parameter settings related to ABC are set as the same as reported values [25], and the learning probabilities of OBL, GOBL and EOBL are set to 0.3, as recommended [25]. Due to the stochastic characteristics of EAs, 30 independent runs for each algorithm and each test function are executed with 300,000 function evaluations (FEs) as the stopping criterion. Moreover, the average and standard deviation of the function error values are recorded for measuring the performance of the ABC algorithms. In order to obtain statistically sound conclusions, two-tailed t-test at a significance level of 0.05 is done on the experimental results [39].

4. 2. Results and Discussions The mean and standard deviation of the function error values achieved by each algorithm for f1-f15 are presented in Table 2. For convenient analysis, the best results among the algorithms are shown in **bold**. The summary of the comparison results are shown in the last three rows of Table 2. "Mean" and "SD" indicate the mean and standard deviation of the function error values obtained by 30 independent runs, respectively.

TABLE 3. Average rankings of the five ABC algorithms for the 15 benchmark test functions obtained by the Friedman test

Algorithm	Ranking
EOABC	**1.70**
dABC	3.00
GOABC	3.17
OABC	3.50
ABC	3.63

The symbols "+", "-", and "≈" denote that EOABC obtains better, worse, and similar results than the corresponding algorithms in terms of the two-tailed *t*-test, respectively.

From Table 2, it can be known that EOABC exhibits better performance than all the other four ABC algorithms on the majority of the 15 benchmark test functions. Specifically, EOABC is significantly better than ABC, OABC, GOABC, and dABC on test functions $f1$, $f2$, f 3, $f4$, $f5$, $f7$, $f8$, $f10$, and $f15$ according to the two-tailed *t*-test. In contrast, ABC, OABC, GOABC, and dABC can not outperform EOABC on any test function. In addition, on test functions $f6$, $f9$, $f12$, $f13$, $f14$, ABC, OABC, GOABC, dABC, and EOABC all exhibit the similar performance. On test functions $f11$, both EOABC and dABC yield similar results, while they are significantly better than ABC, OABC, and GOABC on this test function. Overall, EOABC performs better than ABC, OABC, GOABC, and dABC on 10, 10, 10, and 9 out of 15 test functions, respectively.

In order to compare the total performance of the five ABC algorithms on the all 15 benchmark test functions, the average ranking of Friedman test is performed on the experimental results following the suggestions in [39, 43]. Table 3 presents the average ranking of the five ABC algorithms on the all 15 benchmark test functions. These five ABC algorithms can be sorted by the average ranking into the following order: EOABC, dABC, GOABC, OABC, and ABC. Therefore, EOABC achieves the best average ranking, which indicates that the total performance of EOABC is better than that of the other four ABC algorithms on the all 15 benchmark test functions. This can be because the EOBL strategy can significantly enhance the exploitation capacity of the basic ABC, and EOBL is more efficient than OBL and GOBL for improving the performance of the basic ABC.

5. CONCLUSIONS

EOBL is an effective strategy to enhance the performance of EAs. In this study, we employ the EOBL strategy to promote the performance of the traditional ABC, and thus propose a modified ABC, called EOABC. In the experiments, EOABC is compared with ABC, OABC, GOABC, and dABc on 15 benchmark test functions. The experimental results show that EOABC can significantly surpass ABC, OABC, GOABC, and dABC on the majority of the benchmark test functions. The comparison results also reveal that EOBL is more efficient than OBL and GOBL for promoting the search ability of the traditional ABC. In the future, we will apply EOABC to other complex optimization problems, such as multi-objective and dynamic optimization problems.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (Nos. 61402481, 61462036, and 41261093), by Natural Science Foundation of Jiangxi, China (Nos. 20151BAB217010, 20151BAB201015) and by the Natural Science Youth Foundation of Hebei Educational Committee (No. QN20131053).

7. REFERENCES

1. Alfi, A. and Modares, H., "System identification and control using adaptive particle swarm optimization", ***Applied Mathematical Modelling***, Vol. 35, No. 3, (2011), 1210-1221.
2. Alfi, A. and Khosravi, A., "Constrained nonlinear optimal control via a hybrid ba-sd", ***International Journal of Engineering***, Vol. 25, No. 3, (2012), 197-204.
3. Khooban, M.H., Alfi, A. and Abadi, D.N.M., "Teaching–learning-based optimal interval type-2 fuzzy pid controller design: A nonholonomic wheeled mobile robots", ***Robotica***, Vol. 31, No. 07, (2013), 1059-1071.
4. Arab, A. and Alfi, A., "An adaptive gradient descent-based local search in memetic algorithm applied to optimal controller design", ***Information Sciences***, Vol. 299, (2015), 117-142.
5. Alireza, A., "Pso with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems", ***Acta Automatica Sinica***, Vol. 37, No. 5, (2011), 541-549.
6. Alfi, A., Khosravi, A. and Lari, A., "Swarm-based structure-specified controller design for bilateral transparent teleoperation systems via μ synthesis", ***IMA Journal of Mathematical Control and Information***, (2013), dnt005.
7. Shokri-Ghaleh, H. and Alfi, A., "A comparison between optimization algorithms applied to synchronization of bilateral teleoperation systems against time delay and modeling uncertainties", ***Applied Soft Computing***, Vol. 24, (2014), 447-456.
8. Alfi, A., Kalat, A.A. and Khooban, M.H., "Adaptive fuzzy sliding mode control for synchronization of uncertain non-identical chaotic systems using bacterial foraging optimization", ***Journal of Intelligent and Fuzzy Systems***, Vol. 26, No. 5, (2014), 2567-2576.
9. Karaboga, D. and Basturk, B., "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm", ***Journal of global optimization***, Vol. 39, No. 3, (2007), 459-471.
10. Alfi, A. and Fateh, M.-M., "Intelligent identification and control using improved fuzzy particle swarm optimization", ***Expert Systems with Applications***, Vol. 38, No. 10, (2011), 12312-12317.
11. Shokri-Ghaleh, H. and Alfi, A., "Optimal synchronization of teleoperation systems via cuckoo optimization algorithm", ***Nonlinear Dynamics***, Vol. 78, No. 4, (2014), 2359-2376.
12. Karaboga, D., Gorkemli, B., Ozturk, C. and Karaboga, N., "A comprehensive survey: Artificial bee colony (abc) algorithm and applications", ***Artificial Intelligence Review***, Vol. 42, No. 1, (2014), 21-57.
13. Guilania, P.P., Sharifia, M., Niaki, S. and Zaretalaba, A., "Redundancy allocation problem of a system with three-state components: A genetic algorithm", . Vol. 27, No. 11, (2014), 1663-1672.

14. Sedighizadeh, M. and Kashani, M.F., "A tribe particle swarm optimization for parameter identification of proton exchange membrane fuel cell (technical note)", ***International Journal of Engineering-Transactions A: Basics***, Vol. 28, No. 1, (2014), 16-25.
15. Deepa, S., Babu, S.R. and Ranjani, M., "A robust statcom controller using particle swarm optimization", ***International Journal of Engineering-Transactions B: Applications***, Vol. 27, No. 5, (2013), 731-738.
16. Storn, R. and Price, K., "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces", ***Journal of global optimization***, Vol. 11, No. 4, (1997), 341-359.
17. Crepinsek, M., Liu, S.-H. and Mernik, M., "Exploration and exploitation in evolutionary algorithms: A survey", ***ACM Computing Surveys (CSUR)***, Vol. 45, No. 3, (2013).
18. Zhu, G. and Kwong, S., "Gbest-guided artificial bee colony algorithm for numerical function optimization", ***Applied Mathematics and Computation***, Vol. 217, No. 7, (2010), 3166-3173.
19. Gao, W. and Liu, S., "Improved artificial bee colony algorithm for global optimization", ***Information Processing Letters***, Vol. 111, No. 17, (2011), 871-882.
20. El-Abd, M., "Opposition-based artificial bee colony algorithm", in Proceedings of the 13th annual conference on Genetic and evolutionary computation, ACM., (2011), 109-116.
21. Banharnsakun, A., Achalakul, T. and Sirinaovakul, B., "The best-so-far selection in artificial bee colony algorithm", ***Applied Soft Computing***, Vol. 11, No. 2, (2011), 2888-2901.
22. Akay, B. and Karaboga, D., "A modified artificial bee colony algorithm for real-parameter optimization", ***Information Sciences***, Vol. 192, (2012), 120-142.
23. Gao, W.-f. and Liu, S.-y., "A modified artificial bee colony algorithm", ***Computers & Operations Research***, Vol. 39, No. 3, (2012), 687-697.
24. Karaboga, D., Ozturk, C., Karaboga, N. and Gorkemli, B., "Artificial bee colony programming for symbolic regression", ***Information Sciences***, Vol. 209, (2012), 1-15.
25. El-Abd, M., "Generalized opposition-based artificial bee colony algorithm", in Evolutionary Computation (CEC), 2012 IEEE Congress on, IEEE., (2012), 1-4.
26. Chen, S.-M., Sarosh, A. and Dong, Y.-F., "Simulated annealing based artificial bee colony algorithm for global numerical optimization", ***Applied Mathematics and Computation***, Vol. 219, No. 8, (2012), 3575-3589.
27. Luo, J., Wang, Q. and Xiao, X., "A modified artificial bee colony algorithm based on converge-onlookers approach for global optimization", ***Applied Mathematics and Computation***, Vol. 219, No. 20, (2013), 10253-10262.
28. Gao, W.-f., Liu, S.-y. and Huang, L.-l., "A novel artificial bee colony algorithm based on modified search equation and orthogonal learning", ***Cybernetics, IEEE Transactions on***, Vol. 43, No. 3, (2013), 1011-1024.
29. Xiang, W.-L. and An, M.-Q., "An efficient and robust artificial bee colony algorithm for numerical optimization", ***Computers & Operations Research***, Vol. 40, No. 5, (2013), 1256-1265.
30. Kang, F., Li, J. and Li, H., "Artificial bee colony algorithm and pattern search hybridized for global optimization", ***Applied Soft Computing***, Vol. 13, No. 4, (2013), 1781-1791.
31. Bansal, J.C., Sharma, H., Arya, K. and Nagar, A., "Memetic search in artificial bee colony algorithm", ***Soft Computing***, Vol. 17, No. 10, (2013), 1911-1928.
32. Karaboga, D. and Gorkemli, B., "A quick artificial bee colony (qabc) algorithm and its performance on optimization problems", ***Applied Soft Computing***, Vol. 23, (2014), 227-238.
33. Gao, W.-f., Liu, S.-y. and Huang, L.-l., "Enhancing artificial bee colony algorithm using more information-based search equations", ***Information Sciences***, Vol. 270, (2014), 112-133.
34. Xiang, W., Ma, S. and An, M., "Habcde: A hybrid evolutionary algorithm based on artificial bee colony algorithm and differential evolution", ***Applied Mathematics and Computation***, Vol. 238, (2014), 370-386.
35. Kıran, M.S. and Fındık, O., "A directed artificial bee colony algorithm", ***Applied Soft Computing***, Vol. 26, (2015), 454-462.
36. Wang, S., Ding, L., Xie, C., Guo, Z. and Hu, Y., "A hybrid differential evolution with elite opposition-based learning", ***Journal of Wuhan University(Natural Science Edition)***, Vol. 59, No. 2, (2013), 111-116.
37. Tizhoosh, H.R., "Opposition-based learning: A new scheme for machine intelligence", in null, IEEE., (2005), 695-701.
38. Rahnamayan, S., Tizhoosh, H.R. and Salama, M., "Opposition-based differential evolution", ***Evolutionary Computation, IEEE Transactions on***, Vol. 12, No. 1, (2008), 64-79.
39. Wang, H., Wu, Z., Rahnamayan, S., Liu, Y. and Ventresca, M., "Enhancing particle swarm optimization using generalized opposition-based learning", ***Information Sciences***, Vol. 181, No. 20, (2011), 4699-4714.
40. Rahnamayan, S., Wang, G.G. and Ventresca, M., "An intuitive distance-based explanation of opposition-based sampling", ***Applied Soft Computing***, Vol. 12, No. 9, (2012), 2828-2839.
41. Yao, X., Liu, Y. and Lin, G., "Evolutionary programming made faster", ***Evolutionary Computation, IEEE Transactions on***, Vol. 3, No. 2, (1999), 82-102.
42. Ali, M.M., Khompatraporn, C. and Zabinsky, Z.B., "A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems", ***Journal of global optimization***, Vol. 31, No. 4, (2005), 635-672.
43. Wang, H., Wu, Z., Rahnamayan, S., Sun, H., Liu, Y. and Pan, J.-s., "Multi-strategy ensemble artificial bee colony algorithm", ***Information Sciences***, Vol. 279, (2014), 587-603.

Elite Opposition-based Artificial Bee Colony Algorithm for Global Optimization

Z. Guo[a], S. Wang[b], X. Yue[a], D. Jiang[c], K. Li[d]

[a] *School of Science, JiangXi University of Science and Technology, Ganzhou, China*
[b] *School of Information Engineering, Shijiazhuang University of Economics, Shijiazhuang, China*
[c] *State Key Laboratory of Software Engineering, School of Computer, Wuhan University, Wuhan, China*
[d] *School of Mathematics and Informatics, South China Agricultural University, Guangzhou, China*

P A P E R I N F O

Paper history:
Received 21 February 2015
Received in revised form 29 July 2015
Accepted 03 September 2015

Keywords:
Evolutionary Algorithm
Artificial Bee Colony
Opposition-Based Learning
Elite Strategy

چكيده

مشکلات متعدد در مهندسی و علوم را می توان به مسائل بهینه سازی تبدیل کرد. الگوریتم کلونی زنبور عسل مصنوعی(ABC) یک الگوریتم تصادفی بهینه سازی به تازگی توسعه یافته است و با موفقیت در زمینه های مختلف استفاده می شود. با این حال، با توجه به ویژگی تصادفی معادله‌ی جستجوی راه حل، الگوریتم ABC قدیمی عموما کمتر مورد استفاده قرار می گیرد. با توجه به این ضعف الگوریتم ABC سنتی، در این مقاله، ما یک الگوریتم ABC پیشرفته با استراتژی یادگیری مبتنی بر مخالفت برگزیدگان (EOABC) ارائه می دهیم. در EOABC پیشنهاد شده، استراتژی یادگیری مبتنی بر مخالفت برگزیدگان همراه با احتمال آموزش موجود اجرا می شود تا ظرفیت بهره برداری افزایش یابد. در آزمایشات، EOABC روی مجموعه ای از توابع آزمون عددی معیار تست شده و با برخی از الگوریتم های ABC دیگر مقایسه شده است. مقایسه نشان می دهد که EOABC می تواند نتایج رقابتی در اکثر توابع آزمون به دست آورد.

***doi**: 10.5829/idosi.ije.2015.28.09c.03*