Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN ARTIFICIAL BEE COLONY ALGORITHM WITH MODIFIED SEARCH STRATEGIES FOR GLOBAL NUMERICAL OPTIMIZATION

[1]JIANFENG QIU, [2]JIWEN WANG, [3]DAN YANG, [4]JUAN XIE
[1, 2, 3] School of Computer Science and Technology, Anhui University, Hefei 230039, Anhui, China
[4] Department of Mathematics & Physics， Anhui University of Architecture, Hefei 230022, Anhui, China

ABSTRACT

The Artificial Bee Colony (ABC) algorithm based on swarm intelligence is a more competitive algorithm than other Evolution Algorithm (EA). The results of recent studies indicate that the ABC algorithm has many advantages but it has two major weaknesses: one is slower convergence speed; the other is getting trapped in local optimal value early. Inspired by differential evolution (DE), different with other improved ABC algorithm based Differential Evolution (DE), we propose a modified ABC algorithm, named it ABC/current-to-best/1, by introducing the best food source (the best solution) and randomly choosing food source (the random solution). Experiments are conducted on a group of 24 benchmark functions. The results testify the performance of ABC/current-to-best/1 algorithm better than original ABC and some pre-existing improved ABC algorithm.

Keywords: *Artificial Bee Colony, Global Numerical Optimization, Search Strategy, Differential Evolution*

1. INTRODUCTION

Population-based algorithm can be mainly classified into two types: Evolutionary Algorithm (EA) and Swarm Intelligence Algorithm (SIA). The two population-based algorithms have a common feature: all possible solutions in population can be moved toward the optimized solution by applying some operators based on the fitness value. In EA, The popular algorithms include Genetic Algorithm(GA)[1],Genetic Programming(GP)[2],Evolution Strategy(ES)[3] and Evolution Programming(EP)[4]. Since the late 1990s, Differential Evolution (DE) has emerged as a competitive EA algorithm [5-6]. From then on, DE has been applied in tackling multimodal, multiobjective, constrained and dynamic optimization problems extensively and has got better experimental results than other EAs. As for swarm intelligence-based algorithm, Bonabeau has defined the swarm intelligence as "...any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies..." [7].The classical examples of swarm intelligence algorithms include Ant Colony Optimization(ACO) algorithm which simulates foraging behavior of ants[9], Particle Swarm Optimization (PSO) algorithm which is composed of birds and simulates the social behavior of bird flocking[9], Virtual bee algorithm(VBA)[10],Artificial bee colony (ABC) algorithm[11] and so on. Moreover some hybrid methods based EA and SIA have been proposed to compensate some drawbacks in using EA or SIA alone [12-15].

In this paper, we will pay more attention to how to make use of ABC algorithm to solve global numerical optimization problem effectively by modifying the searching strategies. As we all know, exploration and exploitation should be carefully balanced in all population-based algorithms in order to achieve better solution. Exploration means independent searching for search space while exploitation means searching process according to collected information move toward objective. In fact, the two above contradicts to each other sometimes. To achieve better performance, we should make balance between exploration and exploitation. There are similar problems in the ABC algorithm, as well. The major problems facing the ABC algorithm and all population-based algorithms include slower convergence speed in solving unimodal problems and easier getting trapped in local optima in solving multimodal problems [16]. In ABC algorithm, the searching strategies are more stressing exploration than exploitation and thereby some important variants about ABC algorithm have been proposed to achieve better global optimization ability by balancing exploration and exploitation [13, 14, 15, 17]. These improved algorithms which are using either EA or PSO to ABC's searching strategies enhance exploitation in some sense but getting into local optimized solution easier.

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In order to balance exploration and exploitation well, inspired by DE [4], we propose a new modifie d searching strategy which is not only improving ex ploitation but also avoiding getting trapped into loc al optimization early.

The rest of this paper is organized as follows. In section 2, we outline classical ABC algorithm and s ome important variants of ABC algorithm. Section 3 proposes modified ABC algorithm, explaining im proved searching strategies in detail. The experime ntal parameters settings and results are described in section 4. Section 5 concludes this paper with a disc ussion.

2. CLASSICAL ABC ALGORITHM AND IMPORTANT VARIANTS

Artificial Bee Colony(ABC) algorithm introduce d by karaboga was first used to find an optimal solu tion in numerical optimization by simulating the be havior of foraging selection[11].The collective intel ligence in ABC is composed of three components: f ood sources, employed bees, unemployed bees(onlo oker bees, scout bees) and two behavior models: re cruitment and abandonment. The whole algorithm s tructure is described as follows:

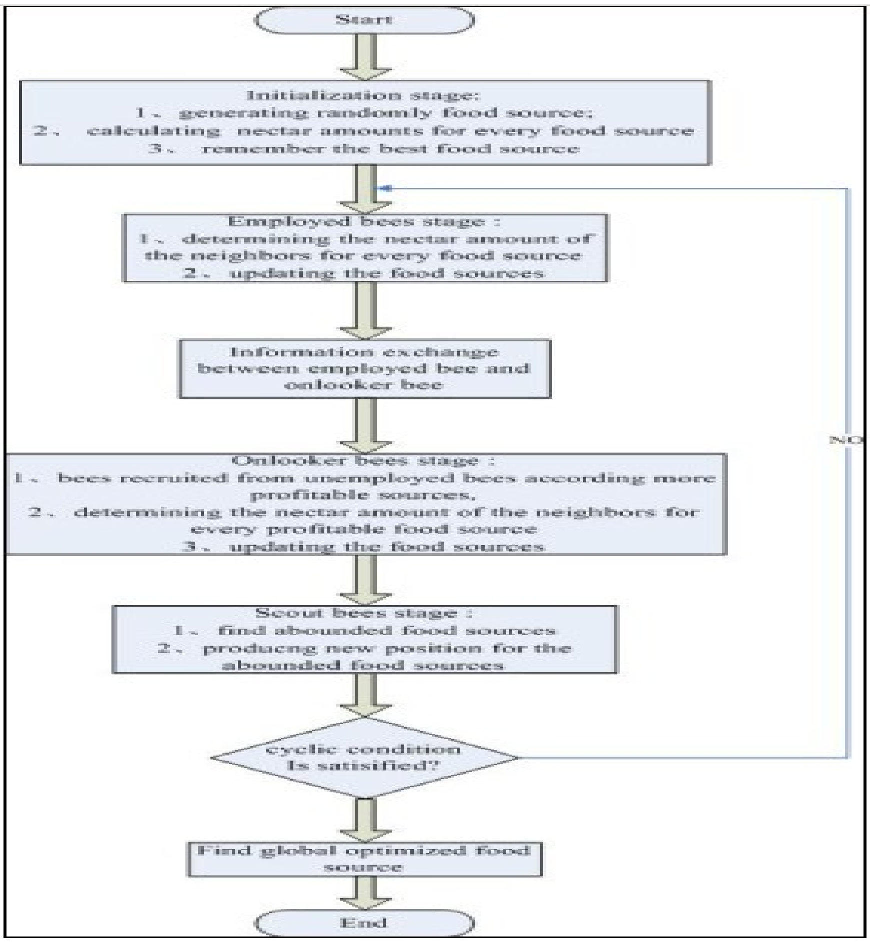

Figure 1: Flowchart Of The Classical ABC Algorithm

In reference [11], a detailed explanation was mad e for classical ABC algorithm. In this section, we o utline the major idea of the algorithm. As we can se e from Figure 1, three major stages are included int o ABC algorithm, employed bees stages, onlooker bees stages and scout bees stages respectively.

2.1 Initialization Stage

In ABC algorithm, every food source position $X_i = (x_{i,1}, x_{i,2} ... x_{i,D})$, where D is the number of problem dimension, representing a possible solution. The ini tial population should contain the range as much as possible by uniformly randomizing individuals with in the search space. This operation is defined as in [16]:

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

Where, x_i^j is the jth component of the ith food so urce position (vector), $x_{\min}^j$ and $x_{\max}^j$ are the mini mum and the maximum of the jth dimension res

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

pectively. In ABC algorithm, the amount of nectar in a food source represents the quality of the solution. In practice, the amount of nectar can be expressed with the optimal value of the objective function. So, the final step in initialization stage is to calculate the fitness value according to the objective functions optimized.

2.2 Employed Bees Stage

In this stage, every employed bee which is corresponding to a food source is in charge of two things. One is going on exploiting the food source which has already been there, the other is being responsible for exchanging information with onlooker bees. In ABC algorithm, exploiting means that employed bees produce a modification on the food source (solution) according to her memory of finding new food source and evaluate it. The ABC algorithm uses (2) for producing a candidate solution:

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) \qquad (2)$$

Where $k \in \{1, 2, 3...SN\}$, SN is the size of population, $j \in \{1, 2, 3...D\}$, k is produced randomly which is different from *i*. Φ_{ij} is restricted in [-1, 1] and determined randomly. v_{ij} is a modification to the food source x_{ij} which is remembered by employed bees. The perturbation on x_{ij} is determined by $\Phi_{ij}(x_{ij} - x_{kj})$. The amplitude of the perturbation is controlled by Φ_{ij}. The employed bee will remember the new food source and forget the old one according to the amount of nectar. Besides exploiting, employed bees carry some information (e.g. nectar amount) which is shared as onlooker bees. An unemployed bee can be recruited to an onlooker bee according to the probability value p_i, associated with that food source.

$$p_i = \frac{fit_i}{\sum_{k=1}^{SN} fit_k} \qquad (3)$$

Where fit_i is the fitness value of the objective functions.

2.3 Onlooker Bees Stage

Onlooker bees are recruited to those food sources which is abundant in nectar amount depending on the information carried by employed bees. According to the probability formula (3), the onlooker bees can select a food source (position) which is rich in nectar amount (the best fitness value). When exchanging onlooker bees fly to the food source, exchanging they will also produce a modification just like employed bees doing so using (2) and then check the nectar amount of the new food source. By comparing the amount of nectar of the new position with the old one, the onlooker bees remember the better food source and forget the other.

2.4 Scout Bees Stage

When a food source (position) can not be improved through predefined number which is the parameter called it "limit" in ABC algorithm, the old food source (position) will be abandoned and replaced by a new food source (position) by the scout bees according to (1).

So, from the explanation above, we can see that there are many advantages in ABC algorithm, such as fewer control parameters, including population size (NP), the value Limit, the max cycle number (MCN) and robust searching process. In robust search progress, exploitation and exploration must be carried out together. Employed and Onlooker bees carry out exploitation while Scout bees are in charge of exploration.

However, like the other EA, there are some problems in ABC algorithm. As indicated in reference [13], the two major problems are slowing convergence speed in handling unimodal and easier getting in local optima in solving multimodal. So, a number of variants have been proposed to improve original ABC by balancing exploration and exploitation and to enhance the searching ability in solving complex global optimized questions [13, 15, 17]. For example, GABC, inspired by PSO [13] improves the exploitation ability by introducing the information of global best solution to searching space. CABC which is taking advantage of chaotic idea to improve ABC [15] and ABC/best, inspired by DE [17], and the like.

To achieve the two above goals, exploration and exploitation, inspired by DE, we propose a new modified searching strategy to improve ABC algorithm, named it ABC/rand-to-best/1 which is different from reference [17]. In the following section, the proposed improved ABC will be explained in detail.

3. ABC/CURRENT-TO-BEST/1 ALGORITHM

Differential Evolution [5] is a simple and very effective EA. The performance of DE is dependent on the mutation strategies, crossover scheme and selection. The family of DE mainly contains some different mutation strategies, such as, DE/rand/1, DE/rand/2, DE/current-to-best/1, DE/best/1, and so on [18]. Different search strategies result in different mutation scheme. The following mutation strategy is used in the literature about DE:

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$DE/current-to-best/1: \vec{V}_{i,G} = \vec{X}_{i,G} + F\cdot(\vec{X}_{best,G} - \vec{X}_{i,G}) + F\cdot(\vec{X}_{r1,G} - \vec{X}_{r2,G}) \quad (4)$$

Where the indices *r1*, *r2* are mutually exclusively integers randomly selected from $[1\cdots NP]$ and different with *i*, *NP* is the size of the population. The factor *F* is a controlling parameter which is used to scale the difference vector. $\vec{X}_{i,G}$ and $\vec{V}_{i,G}$ are known as target vector and donor vector. $\vec{X}_{best,G}$ is the best individual at the generation G which is the minimum objective function value in the minimum problem. The best solution is introduced to the current generation in (4), which guides the searching progress. It is beneficial to discover rapidly the best solution [6].

Inspired by the above mutation in DE and based on the characteristic of ABC algorithm, we modify the strategies in searching new food source by employed and onlooker bees, named it ABC/current-to-best/1. The expression is as follows:

$$ABC/current-to-best/1: \vec{v}_{j,i,G} = \vec{x}_{j,i,G} + F_1\cdot(\vec{x}_{j,best,G} - \vec{x}_{j,i,G}) + F_2\cdot(\vec{x}_{j,r1,G} - \vec{x}_{j,r2,G}) \quad (5)$$

Where, $x_{j,i,G}$ is the jth component in the ith food source at the generation G. similarly, $x_{j,best,G}$ is the jth component of the best food source in the current generation. The definition *r1*, *r2* is the same as the above in *DE/current-to-best/1*. The combination of $\vec{x}_{j,best,G} - \vec{x}_{j,i,G}$ and $\vec{x}_{j,r1,G} - \vec{x}_{j,r2,G}$ to perturb the target vector $x_{j,i,G}$. The one difference $\vec{x}_{j,best,G} - \vec{x}_{j,i,G}$ indicates the distance between the current food source and the best food source in the current generate, which helps to fast discover optimized food source fast but exists some risk in getting trapped local optima; the other difference $\vec{x}_{j,r1,G} - \vec{x}_{j,r2,G}$ reflects some random exploration in the neighbor of the old food source. So, the improved method we propose is not only keeping the search guide to the optimized solution rapidly but also keeps a certain stochastic exploration. The scaling factors *F1* and *F2* are controlling parameters for the two above differences and are generated as uniform random numbers in [*0*, *f1*] and [*0*,*f2*] respectively. It is noted that parameters *f1* and *f2* play an important role in producing candidate solution. By adjusting different group of (*f1*, *f2*), we can achieve optimum searching process.

In this paper, we modify the search strategies at the employed and onlooker bees stage in the original ABC algorithm, inspired by DE. Although some modifications about ABC based on DE are also made in reference [17], it is different from our modification. The modification in [17] increases the exploitation of ABC algorithm but gets trapped in local optimization early sometimes.

4. EXPERIMENTS

Some experiments are designed to illustrate our modification to ABC algorithm. We used 24 benchmark problems [16] including unimodal, multimodal, separable and non-separable in order to test the performance of ABC/current-to-best/1 algorithm. The 24 benchmark functions are listed in Table 1. Because the parameters (f1, f2) play an important role in the algorithm we propose, we will test how to choose f1 and f2 to achieve better searching optimization ability in the first experiment. The second one is comparing between original ABC and ABC/current-to-best/1 algorithm we propose and the last group is done between pre-existing ABC algorithm (e.g. those improved) and that we propose

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table1. Benchmark Function Used In Experiments
D: Dimension, C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable

No	Function	Search Range	C	D	Formulation	Min
1	Step	[-100,100]	US	30/60	$f(x)=\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2$	0
2	Sphere	[-100,100]	US	30/60	$f(x)=\sum_{i=1}^{n}x_i^2$	0
3	SumSquares	[-10,10]	US	30/60	$f(x)=\sum_{i=1}^{n}ix_i^2$	0
4	Quartic	[-1.28,1.28]	US	30/60	$f(x)=\sum_{i=1}^{n}ix_i^4+random[0,1)$	0
5	Beale	[-4.5,4.5]	UN	5	$f(x)=(1.5-x_1+x_1x_2)^2+(2.25-x_1+x_1x_2^2)^2+(2.625-x_1+x_1x_2^3)^2$	0
6	Easom	[-100,100]	UN	2	$f(x)=-\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	-1
7	Matyas	[-10,10]	UN	2	$f(x)=0.26(x_1^2+x_2^2)-0.48x_1x_2$	0
8	Colville	[-10,10]	UN	4	$f(x)=100(x_1^2-x_2)^2+(x_1-1)^2+(x_3-1)^2+90(x_3^2-x_4)^2+10.1((x_2-1)^2+(x_4-1)^2)+19.8(x_2-1)(x_4-1)$	0
9	Trid10	[-D²,D²]	UN	10	$f(x)=\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n}x_ix_{i-1}$	-210
10	Schwefel2.22	[-10,10]	UN	30/60	$f(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	0
11	Rosenbrock	[-30,30]	UN	3/4	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	0
12	Dixon-Price	[-10,10]	UN	30/60	$f(x)=(x_1-1)^2+\sum_{i=2}^{n}i(2x_i^2-x_{i-1})^2$	0
13	Bohachevsky1	[-100,100]	MS	2	$f(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$	0
14	Booth	[-10,10]	MS	2	$f(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	0
15	Rastrigin	[-5.12,5.12]	MS	30/60	$f(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	0
16	Schwefel	[-500,500]	MS	30/60	$f(x)=n*418.982887-\sum_{i=1}^{n}(x_i\sin(\sqrt{\lvert x_i\rvert}))$	0
17	Schaffer	[-100,100]	MN	2	$f(x)=0.5+\frac{\sin^2(\sqrt{x_1^2+x_2^2})-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	0
18	Six Hump Camel Back	[-5,5]	MN	2	$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	-1.03163
19	Bohachevsky2	[-100,100]	MN	2	$f(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)\cos(4\pi x_2)+0.3$	0
20	Bohachevsky3	[-100,100]	MN	2	$f(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1+4\pi x_2)+0.3$	0
21	GoldStein-Price	[-2,2]	MN	2	$f(x)=[\frac{1+(x_1+x_2+1)^2}{(19-14x_1+13x_1^2-14x_2+6x_1x_2+3x_2^2)}]*[\frac{30+(2x_1-3x_2)^2}{(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)}]$	3
22	Griewank	[-600,600]	MN	30/60	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	0
23	Ackley	[-32,32]	MN	30/60	$f(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	0
24	Penalized2	[-50,50]	MN	30/60	$f(x)=0.1\{[\frac{\sin^2(\pi x_1)+}{\sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})}]+(x_n-1)^2[1+\sin^2(2\pi x_n)]\}+\sum_{i=1}^{n}u(x_i,5,100,4)$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, x_i>a\\0,\quad -a\le x_i\le a\\k(-x_i-a)^m, x_i<-a\end{cases}$	0

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.1 Influence of Parameters in ABC/Current-to-Best/1

Different from original ABC and other improved ABC algorithms, the parameter (*f1, f2*) plays an im portant role in ABC/current-to/best/1 algorithm. So, we make use of several typical benchmark function s to see how the parameters (*f1, f2*) influence the ab ility of ABC/current-to-best/1 algorithm. The result is listed in Table 2. The arrangement of the paramet ers is the same as GABC [13].

From Table 2, as a whole, ABC/current-to-best/1 displays excellent performance when parameters (*f 1, f2*) are (1.6, 0.4) rough. Here, the result on the Gr iewank and Rastrigin is better than the two others.

Table 2 . The Analysis Of The Performance Of Parameters In ABC/Current-To-Best/1

(C1,C2) / Fun	(0.0, 2.0)	(0.4, 1.6)	(0.6, 1.4)	(0.8, 1.2)	(1.0, 1.0)
Sphere	1.3043e-005(mean) （5.6479e-005）(std)	1.3208e-58 (1.1728e-58)	1.1729e-78 (8.2620e-79)	3.3911e-97 (3.0445e-97)	2.3591e-116 (3.2080e-116)
Griewank	0.0121 (0.0124)	7.5658e-07 (4.1393e-06)	**0** **(0)**	**0** **(0)**	**0** **(0)**
Rastrigin	0.0097 (0.0162)	6.0583e-010 (3.3181e-009)	**0** **(0)**	**0** **(0)**	**0** **(0)**
Ackley	1.2475e-004 (2.0417e-004)	3.5468e-14 (3.9510e-15)	3.4284e-14 (4.6275e-15)	3.1442e-14 (3.3118e-15)	3.0731e-14 (2.0010e-15)
	(1.2, 0.8)	(1.4, 0.6)	**(1.6, 0.4)**	(1.8, 0.2)	(2.0, 0)
Sphere	1.4067e-130 (1.4790e-130)	2.1038e-137 (4.2126e-137)	**4.5502e-140** **(9.0476e-140)**	6.2978e-140 (1.1991e-139)	7.7552e-134 (1.1756e-133)
Griewank	**0** **(0)**	**0** **(0)**	**0** **(0)**	1.4297e-08 (7.8310e-08)	2.4898e-04 (0.0013)
Rastrigin	**0** **(0)**	**0** **(0)**	**0** **(0)**	**0** **(0)**	**0** **(0)**
Ackley	2.9665e-14 (2.3511e-15)	2.9428e-14 (2.7174e-15)	**2.9428e-15** **(1.9755e-15)**	3.0020e-14 (2.1681e-15)	3.0139e-14 (1.7906e-15)

4.2 Comparison Between ABC/Current-To-Best/1 And Original ABC

In the second experiment, parameters are arrange d below: population size SN is 100, limit is SN/2*D. All benchmark functions listed in Table 1 are condu cted for different dimensions. Each of the experime nt was run 30 times independently. The means and standard deviations of the experiments comparing o riginal ABC with that of we propose are reported in Table 3.

From the Table 3, we can see clearly that ABC/c urrent-to-best/1 displays preferable ability of search ing optimization in most cases through the searchin g space. In order to explain convergence speed mor e visually, we choose several benchmark functions t o illustrate it in Figure 2 - 7.

4.3 ABC/Current-To-Best/1 Vs. Other Pre-Existi ng Improved ABC

In this part, we will assess the performance of wh at we propose with GABC [13], ABC/best/1, ABC/ best/2[17], EABC [19]. The setting of parameters is the same as [13]. The result of the experiment is lis ted in Table 4 below.

It is distinct that there is better result on Rosebro ck and Sphere test functions and the others keep sli mily consequence. As a whole, ABC/current-to-bes t/1 has been demonstrated as good performance co mpared with original ABC algorithm in Table 4.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3. The Performance Comparison Of ABC And ABC/Current-To-Best/1

Fun	**D**	**G**	***ABC* (*mean***	***std)***	***ABC/current-to-best/1* (*mean***	***std)***
F1	30	1000	0	0	0	0
	60	2000	0	0	0	0
F2	30	1000	6.99e-10	5.91e-10	**1.5150e-030**	**1.1045e-27**
	60	2000	1.94e-09	8.33e-10	**4.8462e-025**	**2.4576e-24**
F3	30	1000	5.1976e-13	4.3798e-13	**1.4495e-025**	**9.6960e-026**
	60	2000	4.4281e-12	3.3042e-12	**1.2956e-023**	**8.0970e-024**
F4	30	1000	1.01e-01	2.44e-02	**0.0402**	**0.0081101**
	60	2000	2.58e-01	2.92e-02	**0.1179**	**0.012609**
F5	5	2000	1.5026e-008	2.3103e-008	**1.1409e-012**	**4.1795e-012**
F6	2	2000	-1.0000	5.1531e-008	-1	5.1531e-09
F7	2	2000	2.9632e-016	2.0788e-016	**3.3994e-048**	**1.4509e-047**
F8	4	2000	1.17e-01	6.94e-02	**0.1e-04**	**9.9096e-04**
F9	10	2000	-209.9540	0.0372	**-209.8984**	**0.0070**
F10	30	1000	2.36e-06	8.32e-07	**5.0520e-017**	**1.6848e-15**
	60	2000	8.30e-06	8.93e-07	**3.4634e-015**	**7.1907e-014**
F11	3	1000	3.93e-02	3.11e-02	**2.3661e-008**	**4.5290e-06**
	4	2000	3.21e-02	3.26e-02	**1.2443e-007**	**1.7276e-05**
F12	30	1000	0.0153	0.0089	**0.0103**	**0.14299**
	60	2000	0.0335	0.0203	**0.0214**	**0.32399**
F13	2	2000	0	0	0	0
F14	2	2000	4.7386e-018	4.6731e-018	**0**	**0**
F15	30	1000	6.63e-03	1.71e-02	**1.3074e-013**	**1.5380e-13**
	60	2000	3.03e-01	4.53e-01	**2.6441e-010**	**1.4324e-09**
F16	30	1000	2.05e+02	1.63e+02	**0**	**0**
	60	2000	6.93e+02	1.39e+02	**1.2164e-12**	**1.3436e-11**
F17	2	2000	3.7007e-018	1.4084e-017	**0**	**0**
F18	2	2000	-1.0316	6.7122e-016	-1.0316	5.4546e-16
F19	2	2000	0	0	0	0
F20	2	2000	3.6822e-016	2.7793e-016	**0**	**0**
F21	2	2000	3.0000	1.5472e-015	**2.6950**	**1.1662e-015**
F22	30	1000	8.73e-09	1.47e-08	**2.3700e-013**	**9.5669e-09**
	60	2000	4.46e-09	6.68e-09	**1.9114e-12**	**6.9137e-12**
F23	30	1000	1.02e-05	4.15e-06	**2.1521e-15**	**7.2099e-13**
	60	2000	2.05e-05	5.54e-06	**1.0075e-15**	**2.1603e-14**
F24	30	1000	3.72e-09	1.79e-09	**2.9044e-032**	**1.5136e-29**
	60	2000	1.06e-08	6.25e-09	**8.8286e-029**	**4.0155e-28**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

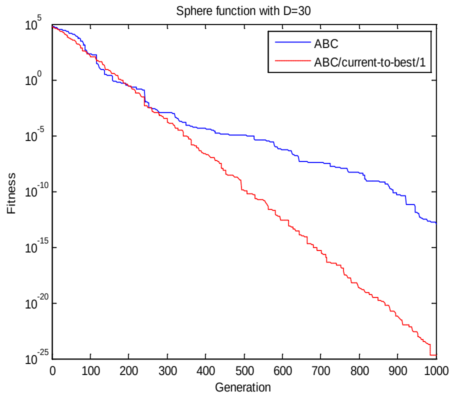

Figure 2: Convergence Curves For Sphere Function

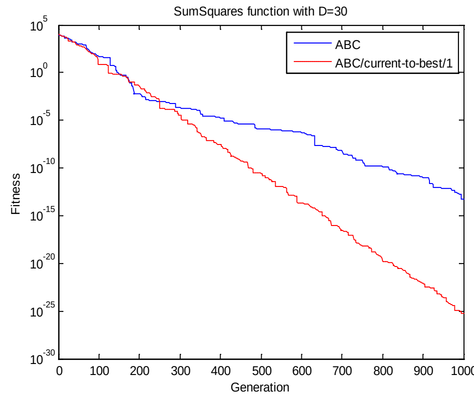

Figure 3: Convergence Curves For Sumsquares Function

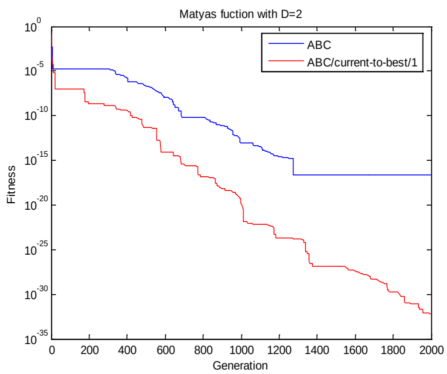

Figure 4: Convergence Curves For Matyas Function

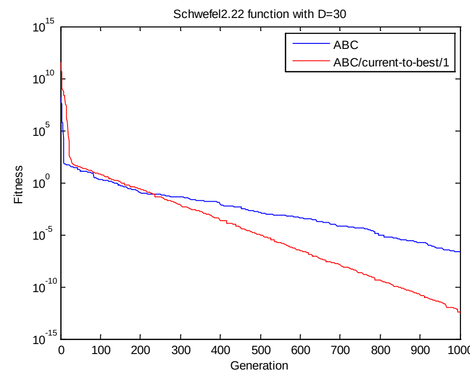

Figure 5: Convergence Curves For Schwefel2.22 Function

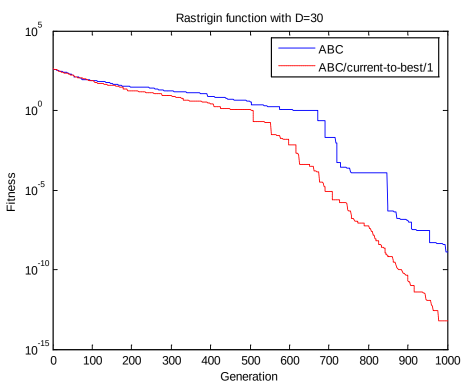

Figure 6: Convergence Curves For Rastrigin Function

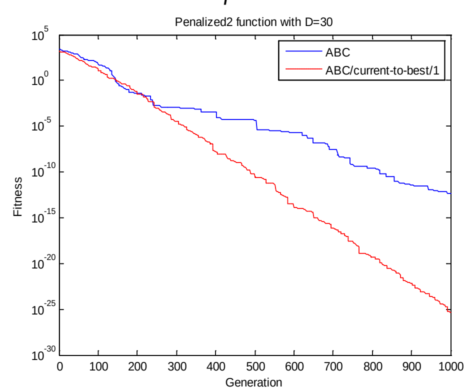

Figure 7: Convergence Curves For Penalized2 Function

Journal of Theoretical and Applied Information Technology
10th February 2013. Vol. 48 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4. Performance Comparison Between ABC/Current-To-Best/1 And Pre-Existing Improved ABC

Algorithm	***Schaffer***				***Rosebrock***			
	D=2		***D=3***		***D=2***		***D=3***	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
GABC	0	0	1.85e-18	1.01e-17	1.68e-04	4.42e-04	2.65e-03	2.22e-03
EABC	0	0	2.79e-07	2.24e-07	4.63e-04	4.57e-04	1.20e-02	7.06e-03
ABC/best/1	0	0	0	0	4.99e-06	8.22e-06	5.52e-06	3.03e-06
ABC/best/2	0	0	3.56e-06	1.27e-06	4.42e-04	2.39e-04	9.90e-04	6.92e-04
ABC/current-to-best/1	0	0	0	0	**3.50e-16**	**1.53e-15**	**3.34e-11**	**1.62e-10**
	Sphere				***Griewank***			
	D=30		***D=60***		***D=30***		***D=60***	
	Mean	***Std***	***Mean***	***Std***	***Mean***	***Std***	***Mean***	***Std***
GABC	4.17e-16	7.36e-17	1.43e-15	1.37e-16	2.96e-17	4.99e-17	7.54e-16	4.12e-16
EABC	1.67e-16	2.70e-16	1.41e-15	1.82e-15	4.90e-14	7.31e-03	4.19e-14	9.05e-03
ABC/best/1	1.1e-150	1.4e-150	4.40e-69	2.56e-69	0	0	0	0
ABC/best/2	1.7e-126	2.7e-126	3.72e-58	2.67e-58	0	0	0	0
ABC/current-to-best/1	**4.55e-160**	**9.05e-157**	**3.92e-72**	**3.13e-70**	0	0	0	0
	Rastrigin				***Ackley***			
	D=30		***D=60***		***D=30***		***D=60***	
	Mean	***Std***	***Mean***	***Std***	***Mean***	***Std***	***Mean***	***Std***
GABC	1.32e-14	2.44e-14	3.52e-13	1.24e-13	3.21e-14	3.25e-15	1.66e-13	2.21e-14
EABC	9.97e-15	3.87e-15	7.51e-13	6.15e-13	1.22e-10	4.86e-11	1.55e-07	2.84e-08
ABC/best/1	0	0	0	0	1.72e-14	2.84e-15	6.62e-14	1.74e-15
ABC/best/2	0	0	0	0	2.50e-14	3.48e-15	7.12e-14	4.14e-15
ABC/current-to-best/1	0	0	0	0	2.94e-14	1.98e-15	7.34e-14	4.44e-15

5. CONCLUSION

In this paper, we modify the searching strategies of Artificial Bee Colony algorithm at the employed and onlooker bees' stage. The modification is inspired by DE and introduces not only the best solution at the current generation but also stochastic perturbation. We can get better balance between exploration and exploitation by adjusting the amplitude of the perturbation *f2* and *f2*.It is clear that the improved method we propose with suitable parameters can enhance the ability of searching optimization effectively by testing a group of 24 benchmark functions.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to the anonymous reviewers for their valuable comments and suggestions on this paper. This work is supported by National Nature Science Foundation of China (Grant Nos. 61075049), the Excellent Young Talents Foundation Project of Anhui Province (Grant Nos.2011SQRL018), the Youth Foundation of Anhui University (Grant No. KJQN1015）, the University Natural Science Research Project of Anhui Province(Grant Nos.KJ2012B038).

REFERENCES:

[1] Eiben, A. E. et al, "Genetic algorithms with multi-parent recombination", Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature, October 9-14, 1994, pp.78–87.

[2] J.R. Koza, "Genetic programming: a paradigm for genetically breeding populations of computer programs to solve problems", Technical Report STAN-CS-90-1314, Stanford University Computer Science Department, 1990.

[3] H.P. Schwefel, Kybernetische, "evolution als strategie der experimentellen forschung in der stromungstechnik", Master's Thesis, Technical University of Berlin, Germany, 1965.

[4] Fogel, D.B. Fogel, L.J. Atmar, J.W, "Meta-evolutionary programming", Proceedings of the twenty-fifth Asilomar Conference on Signals, systems&computers, *IEEE Conference Publishing Services,* November 4-6, 1991, pp. 540-545.

[5] Storn. R, Price. K, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces", *Journal of global optimization,* Vol. 11, No. 4, 1997, pp. 341-359.

[6] Mezura-Montes.E, Velázquez-Reyes. J, Coello Coello. C.A, "A comparative study of differential evolution variants for global optimization", Proceedings of the 8th annual conference on G

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

enetic and evolutionary computation, July 08 – 12, 2006, pp. 485-492.

[7] E. Bonabeau, M. Dorigo, G. Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems", Oxford University Press, NY, 1999.

[8] Dorigo. M, Maniezzo. V, Colorni. A, "Ant system: optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man, and Cybernetics, *Part B: Cybernetics*, Vol. 26, No. 1, 1996, pp. 29-41.

[9] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, October 4-6, 1995,pp.39-43.

[10] X.S. Yang, "Engineering optimizations via nature-inspired virtual bee algorithms", *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, vol. 3562, No. 2, 2005, pp. 317–323.

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

[12] Srinivasan. D, Seow. T.H, "Particle swarm inspired evolutionary algorithm (PS-EA) for multiobjective optimization problems", Proceedings of International Conference on Evolutionary Computation, *IEEE Conference Publishing Services*, Dec 8–12,2003,pp. 2292–2297.

[13] G.P. Zhu, S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization", *Applied Mathematics and Computation*, Vol. 217, No.7, 2010, pp. 3166–3173.

[14] B. Akay, D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization", *Information Sciences*, Vol. 192, 2010, pp. 120-142.

[15] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization", *Expert Systems with Applications*, Vol. 37, 2010, pp. 5682–5687.

[16] D. Karaboga, B. Basturk, "A comparative study of artificial bee colony algorithm", *Applied Mathematics and Computation*, Vol. 214, 2009, pp.108–132.

[17] Weifeng Gao, Sanyang Liu, Lingling Huang, "A global best artificial bee colony algorithm for global optimization", *Journal of Computational and Applied Mathematics*, Vol. 236, 2012, pp.2741-2753.

[18] S. Das and P. N. Suganthan, "Differential evolution—A survey of the state-of-the-art," *IEEE Trans. Evol. Comput*, Vol.15, No.1, 2011, pp. 4–31.

[19] E.M. Montes, et al, "Elitist artificial bee colony for constrained real-parameter optimization", Proceedings of International Conference on Evolutionary Computation, *IEEE Conference Publishing Services*, July 18-23, 2010, pp.4244-6909.