A Parameter Adaptive Artificial Bee Colony Algorithm for Real-Parameter Optimization

http://dx.doi.org/10.3991/ijoe.v9iS4.2609

Jian feng Qiu[1], Ji wen Wang[1], Dan Yang[1], Juan Xie[2], Nan zhen Yao[3]
[1] Anhui University, Hefei, Anhui, China
[2] Anhui University of Architecture, Hefei, Anhui, China
[3] Library of University of Science & Technology of China, Hefei, Anhui, China

***Abstract*—A new adaptive variant of Artificial Bee Colony algorithm, PAABC, is proposed to improve optimization performance and enhance the robustness of ABC algorithm by incorporating $current-to-\overline{pbest}$ into searching strategy and updating control parameter of searching equation adaptively . The incorporation of the $\overline{pbest}$ information is helpful to accelerate convergence while avoiding prematurity especially multimodal problem. According to the characteristic of optimization problem, the control parameter will be update adaptively. The better parameter value associated with the mean of the p% best individual will survive into the next generation. Experiment results show that PAABC is better or equal to evolutionary algorithm according to a set of basic test function and the CEC'13 test suite.**

***Index Terms*—PAABC; Artificial Bee Colony; adaptive parameter; best; Differential evolution; optimization**

I. INTRODUCTION

The Artificial Bee Colony (ABC) algorithm was first introduced by Karaboga [1]. ABC algorithm is an optimization algorithm by simulating the intelligent behavior of honey bee swarm. It has been shown to be one of competitive optimization algorithms for many real-world applications [2]-[5]. The performance for optimizing a set of numerical test functions has been proved to achieve better results compared with genetic algorithm, particle swarm optimization (PSO) algorithm and differential evolution (DE) [6]. In recent years, a series of schemes on improving the Artificial Bee Colony algorithm are discussed, and the new applications are also provided. References [7] give a modified ABC algorithm by bringing "Frequency of the perturbation" and "Magnitude of the perturbation" into ABC algorithm for efficiently solving real-parameter optimization. Inspired by PSO, GABC incorporated the information of global best solution improved the exploitation of ABC algorithm [8]. Using the different mutation strategies in DE algorithm for reference, some improved ABC algorithms based on DE were presented [9] [10].

In optimization algorithms, exploration and exploitation processes should be well balanced. In ABC algorithm which includes Employed bees, Onlooker bees and Scout bees' stage, employed bees and onlooker bees carry out the exploitation while scout bees can be responsible for exploration in the search process. One advantage of ABC algorithm is fewer control parameters, that is, NP (population size), Limit (predefined number of cycles) and MCN (maximum cycle number). How to determine the above parameters will directly affect the optimal purpose. There is no a uniform mode to choice ABC's parameters for different optimization problems. In present work, except for "limit", the other parameters are adjusted through trial-and-error method. The suggested choices about the parameter "limit" by Karaboga in [6], [7] are:

1. limit=$NP*D$;
2. limit=200;

Recall that D is the dimensionality of the optimized problem.

Some improved ABC algorithms which are focused on the improvement of mutation strategies in employed and onlooker stage also usually adopt trial-and-error method for different problems[8][11]. Although there are already some proposed improvements, it is difficult to choose a suitable parameter setting for different optimized problems. Because the complexity of the optimized problem (non-convex, multi-peak, multi- dimension), we can not fully understand the relationship between optimized algorithm and problems. In order to achieve the best parameter setting, the trial-and-error method is adopted by tuning the parameters. Recently, some dynamic update parameters mechanisms without understanding the optimized problems which is called parameter adaptation mechanism have been brought up. According to the explanation of Angeline [12] and Eiben et al [13], [14], the adaptation can be classified three types:

1. Deterministic parameter control which is used to update the parameters according to deterministic rules unrelated to the search process.
2. Adaptive parameter control can change the parameters according to the feedback of search process. It can be used in the algorithm proposed in this paper.
3. Self-adaptive parameter control is the idea that incorporating the parameters into the evolution process and make better parameters associated with better individual survive during the evolution process.

Adaptation and self-adaptation were introduced in the ABC algorithm in recent years. ABC-SAM, one self-adaptive mutation method used in ABC algorithm, was introduced by Alam [15]. It is a variant of ABC algorithm by executing an 'adaptation cycle' to adapt the scaling factor *SF*. For every individual x_i , we compare the three different offspring producing by $SF_i, SF_i = 2^{\mu_1}, SF_i = 2^{\mu_2}$ and choose the best fitness value into the next generation. The SF_i corresponding to the best fitness value will be retained. The experimental results shows that ABC-SAM which adapt the mutation step size suitably for better exploitations or explorations. In [16], the self adaptive artificial bee colony algorithm is proposed by employing the self adaptive iteration according the relative maximum iterations.

The search strategies used in ABC algorithm which is similar to the mutation in DE is good at exploration but poor at i exploitation [11]. Inspired by DE, in order to achieve balance between exploration and exploitation, an improved ABC algorithm, ABC/current-tobest/1, has been proposed [10]. The algorithm considers not only fast convergence using the best solution but also avoids plunging into local minimum early by introducing two random solutions as perturbation.

As we all know, a good search strategy and well designed parameters adaptation are quite crucial to the performance of ABC algorithm and the robustness. In this paper, we adjust the parameters of the solution search equation of ABC algorithm by applying one parameter adaptation method not trial-and-error method. We name the algorithm using adaptive parameters setting as Parameter Adaptive Artificial Bee Colony algorithm (PAABC).

The rest of paper is organized as follows. Section II outlines the basic ABC algorithm. The proposed Parameter Adaptive ABC algorithm will be elaborated in Section III. Experiments and parameters setting are presented in Section IV. Finally, some conclusion is concluded in Section V.

II. OUTLINE OF ARTIFICIAL BEE COLONY ALGORITHM

As a simulation for foraging behavior of real honey bee swarm, the ABC algorithm is composed of three essential components: employed bees, onlooker bees and scout bees. The collective intelligence is formed by the division of labor and mutual cooperation among them. In ABC, the position of nectar source represents a possible solution in optimized problem. It is usually expressed as $X_i = (x_{1,i}, x_{2,i}, \cdots x_{D,i})$, where D is the number of dimension. The nectar amount of food source corresponds to the quality of the possible solution which is normally specified as a function of position vector. Employed bees search in the neighbor of every food source after initialization. The equation of search strategy is described as follows:

$$v_{i,j} = x_{i,j} + \Phi_{i,j}(x_{i,j} - x_{k,j}) \tag{1}$$

where $k \in \{1, 2, \cdots SN\}, k \neq i$ and $j \in \{1, 2, \cdots D\}$ are random generated indexes. $\Phi_{i,j}$ is a random number between -1 and +1. After that, the information exchange is occurred between employed bees and onlooker bees. The onlooker bees are recruited to the food sources which contain rich nectar depending on the probability value P_i associated with every food source, usually described as follows:

$$P_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \tag{2}$$

where $fitness_i$ is the fitness value for every possible solution i in optimized problem which represent the assessment for nectar source in honey bee swarms; *SN* is the number of swarm size that is the number of food source, which equals the number of employed bees plus onlooker bees. In onlooker bees' stage, the bee which is recruited according to (2) flies to the food source which is better solution corresponding to rich amount nectar. Then, the bee became onlooker bee and exploit further as (1). If a solution (i.e. position of food source) can not be improved within the specified number of cycles, the solution will be abandoned. The specified number of cycles is the parameter "limit" in ABC algorithm. After the solution has been abandoned, a new food source will be produced and displace the abandoned food source. The position of new food source can be expressed as follows:

$$v_{i,j} = x_{\min,j} + rand(0,1)(x_{\max,j} - x_{\min,j}) \tag{3}$$

where $x_{\min,j}$ and $x_{\max,j}$ are the upper and lower bounds of the position in jth dimension. The flowchart of the basic ABC algorithm is described as follows:

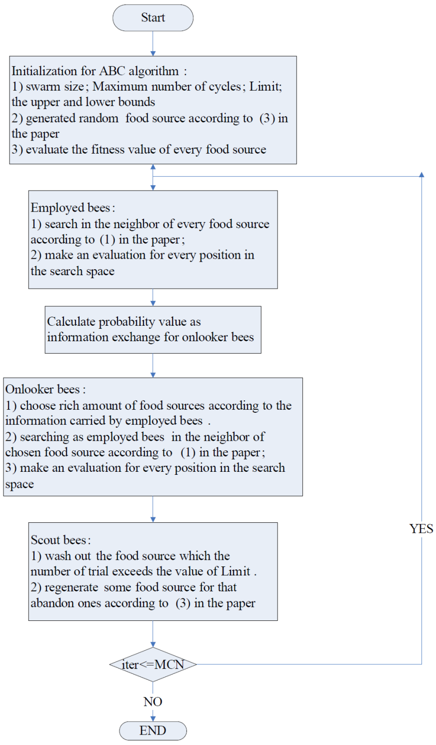
Start
Initialization for ABC algorithm :
1) swarm size; Maximum number of cycles; Limit; the upper and lower bounds
2) generated random food source according to (3) in the paper
3) evaluate the fitness value of every food source
Employed bees :
1) search in the neighbor of every food source according to (1) in the paper;
2) make an evaluation for every position in the search space
Calculate probability value as information exchange for onlooker bees
Onlooker bees :
1) choose rich amount of food sources according to the information carried by employed bees .
2) searching as employed bees in the neighbor of chosen food source according to (1) in the paper;
3) make an evaluation for every position in the search space
Scout bees :
1) wash out the food source which the number of trial exceeds the value of Limit .
2) regenerate some food source for that abandon ones according to (3) in the paper
YES
iter<=MCN
NO
END

Figure 1. Flowchart of basic ABC algorithm

III. PARAMETER ADAPTIVE ABC ALGORITHM

In this section, a new adaptive variant of ABC algorithm is proposed based on ABC/current-to-best [10] which is different from the basic ABC algorithm. We name it as "Parameter adaptive ABC algorithm". At the same time, we control the parameter $\Phi_{i,j}$ in an adaptive manner in order to enhance robust search process. In next section, the parameter $\Phi_{i,j}$ will be replaced by F_i.

A. $ABC / current - to - \overline{pbest}$

In ABC algorithm, the search equation of employed bees' and onlooker bees' stage is similar to the mutation in

DE algorithm. Inspired by DE, in recent years, some improvements have been proposed based on the mutation strategies of DE, for example, ABC/rand, ABC/best, ABC/current-to-best[10][17]. It is helpful for enhancing the convergence speed of ABC algorithm by integrating the best solution. However, the integration of the greedy strategies may lead to trap in local optimum especially multimodal problem. How to make best use of the advantages and bypass the disadvantages of greedy strategies, inspired by DE algorithm, a new variant of ABC, named " $ABC/current-to-\overline{pbest}$ " proposed in this paper. The new search equation used in Employed bees and onlooker bees' stage is expressed as follows:

$$v_{i,G}=x_{i,G}+F_i(\overline{x_{pbest,G}}-x_{i,G})+F_i(x_{r1,G}-x_{r2,G}) \tag{4}$$

where $\overline{x_{pbest,G}}$ is the mean of the first 100p% individuals in the current population with $p\in(0,1]$, F_i is the scaling factor related to every individual and updated by adaptation process in every generation. The algorithm proposed in this paper makes use of the best information to accelerate convergence velocity while avoiding premature in solving multimodal problem.

B. Parameter Adaptation in search strategy

At every generation, the scaling factor of the search equation used in employed bees' and onlooker bees' stage influence the effectiveness of the algorithm we proposed. Every F_i associated with individual x_i is generated according to Cauchy distribution function. The generated scaling factors can avoid premature more effectively through Cauchy distribution than traditional norm distribution.

$$F_i=randcuchy(\mu_F,0.1) \tag{5}$$

where μ_F is location parameter and 0.1 is the scale parameter. After initialization, the F_i will be updated according to (6):

$$\mu_F=(1-k)\cdot\mu_F+k\cdot mean_A(F) \tag{6}$$

where $mean_A()$ is arithmetic mean operation and F is a set consisted of the F_i which is associated with food source i survived at every generation.

C. parameter settings

Compared with basic ABC algorithm, two new parameters *k* and *p* have been added in the algorithm proposed in this paper. The parameter k controls the amplitude of searching process. The parameter c decides the degree of greedy avoiding premature in searching for optimal solution. As we will see in the Section IV, PAABC we proposed can achieve best performance in most cases with $k\in[0.25,0.5]$ and $p\in[0.25,0.5]$.

IV. EXPERIMENT AND CONCLUSION

In this section, two groups of experiments will be arranged to test the performance of ABC algorithm. Group one is a set of tests based on the basic test functions which is used in [8] and given in table I. In order to make comprehensive evaluation of the algorithm we proposed, all types of benchmark functions suggested in CEC 2013[18] were tested in group two.

A. Testing on basic function

The first experiment consists of some multimodal or unimodal functions. In order to assess the performance of algorithm proposed in this paper and compare with other existed optimization algorithm, some values of parameters were be set according to [8]. The population size was 10, the parameter limit was 200, and the maximum number of function evaluation was 30,000.All experiments were conducted 30 times.

The results of group one was given in table II.

TABLE I.
BASIC FUNCTION

Function	Global min	Search range	Formula
Shpere(UM)	0	[-100,100]^D	$f(x)=\sum_{i=1}^{n}x_i^2$
Rosenbrock(UM)	0	[-2.048,2.048]^D	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$
Ackley(MM)	0	[-32.768,32.768]^D	$f(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})$ $-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$
Griewank(MM)	0	[-600,600]^D	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$
Weierstrass(MM)	0	[-0.5,0.5]^D	$f(x)=\sum_{i=1}^{D}(\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k(x_i+0.5))])$ $-D\sum_{k=0}^{k\max}[a^k\cos(a^k\cos(2\pi b^k 0.5))],a=0.5,b=3,k\max=20$
Rastrigin(MM)	0	[-5.12,5.12]^D	$f(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$
Schwefel(MM)	0	[-500,500]^D	$f(x)=n*418.982887-\sum_{i=1}^{n}(x_i\sin(\sqrt{\lvert x_i\rvert}))$

The scalability of the algorithm is important for the algorithm. Table III summarize how the performance of the PAABC algorithm increasing with dimension.

From table II, we can see that PAABC algorithm we proposed has obvious advantages in testing basic function especial for multimodal.

Remark: As we can see form Table III , there is no obvious influence on the performance of the PAABC algorithm when the dimension increasing from 10D to 100D. The results show that the PAABC algorithm is not sensitive for the increment of dimension. The algorithm we proposed has a preferable scalability.

TABLE II.
D:10, MAXIUM EVALUATION:30,000,RUNS:30,MM:MULTIMODAL,UM:UNIMODAL

mean *(std)*	*Sphere* *UM*	*Rosebrock* *UM*	*Ackley* *MM*	*Griewank* *MM*	*Weierstrass* *MM*	*Rastrigin* *MM*	*Schwefel* *MM*
PSO-w	7.96E-051 (3.56 E-050)	3.08E+000 (7.69E-001)	1.58E-014 (1.60E-0 14)	9.69E-002 (5.01 E-002)	2.28E-003 (7.04E-003)	5.82E+000 (2.96E+ 000)	3.20E+002 (1.85E +002)
PSO-cf	9.84E-105 (4.21E-104)	6.98E-001 (1.46E+000)	9.18E-001 (1.01E+000)	1.19E-001 (7.11E-002)	6.69E-001 (7.17E-001)	1.25E+001 (5.17E+ 000)	9.87E+002 (2.76E +002)
PSO-w-local	2.13E-035 (6.1 7E-035)	3.92E+000 (1.19 E+000)	6.04E-015 (1.67E-015)	7.80E-002 (3.79 E-002)	1.41E-006 (6.31E-0 06)	3.88E+000 (2.30E+ 000)	3.26E+002 (1.32E +002)
PSO-cf-local	1.37E-079 (5.60 E-079)	8.60E-001 (1.56E +000)	5.78E-002 (2.58E-0 01)	2.80E-002 (6.34E-002)	7.85E-002 (5.16E-0 02)	9.05E+000 (3.48E+ 000)	8.78E+002 (2.93E+ 002)
UPSO	9.84E-118 (3.56 E-117)	1.40E+000 (1.88E +000)	1.33E+000 (1.48E+ 000)	1.04E-001 (7.10 E-002)	1.14E+000 (1.17E+ 000)	1.17E+001 (6.11E+ 000)	1.08E+003 (2.68E+ 002)
FDR	2.21E-090 (9.88 E-090)	8.67E-001 (1.63E +000)	3.18E-014 (6.40E-0 14)	9.24E002 (5.61E-002)	3.01E-003 (7.20E-0 03)	7.51E+000 (3.05E+ 000)	8.51E+002 (2.76E+ 002)
FIPS	3.15E-030 (4.5 6E-030)	2.78E+000 (2.26 E-001)	3.75E-015 (2.13E-0 14)	1.31E-001 (9.32 E-002)	2.02E-003 (6.40E-0 03)	2.12E+000 (1.33E+ 000)	7.10E+001 (1.50E+ 002)
CPSO-H	4.98E-045 (1.00 E-044)	1.53E+000 (1.70E +000)	1.49E-014 (6.97E-0 15)	4.07E-002 (2.80 E-002)	1.07E-015 (1.67E-0 15)	0.00E+000 (0.00E+000)	2.13E+002 (1.41E+ 002)
M-ABC[8]	7.04E-017 (4.5 5-017)	4.42E-001 (8.67E-001)	3.32E-016 (1.84E-016)	1.52E-002 (1.2 8E-002)	1.18E-016 (6.38E-016)	1.61E-016 (5.20E-016)	3.96E+000 (2.13E+00)
PAABC	**0.00E+000 (0.00E+000)**	**1.38E-001 (1.51E-001)**	**4.44E-017 (0.00E+000)**	**6.98E-003 (9.25E-003)**	**0.00E+000 (0.00E+000)**	**0.00E+000 (0.00E+000)**	**1.27E-004 (0.00E+000)**

PSO-w, PSO-cf, PSO-w-local, PSO-cf-local, UPSO FDR, FIPS, CPSO-H were given in [19]

TABLE III.
MEAN AND STANDARD DEVIATION OF ERROR VALUES ACHIEVED FROM PAABC ALGORITHM

CS	**Function**	**D**		
		10	***50***	***100***
10	Sphere	0.00E+000±0.00E+000	1.76E-301±0.00E+000	2.69E-309±0.00E+000
	Rosebrock	1.56E+000±1.96E+000	5.39E+001±2.92E+001	1.08E+002±2.98E+001
	Ackley	4.44E-015±0.00E+000	5.36E-014±4.58E-015	1.42E-013 ±6.82E-015
	Griewank	5.30E-003±6.45E-003	0.00E+000±0.00E+000	0.00E+000±0.00E+000
	Weierstrass	0.00E+000±0.00E+000	0.00E+000±0.00E+000	0.00E+000±0.00E+000
	Rastrigin	0.00E+000±0.00E+000	0.00E+000±0.00E+000	0.00E+000±0.00E+000
	Schwefel	1.27E-004±0.00E+000	6.36E-004±0.00E+000	1.27E-003±3.39E-012
20	Sphere	0.00E+000±0.00E+000	0.00E+000±0.00E+000	0.00E+000±0.00E+000
	Rosebrock	6.37E-001±1.56E+000	2.52E+001±1.47E+001	9.85E+001±3.33E+001
	Ackley	4.44E-015±0.00E+000	5.25E-014±4.91E-015	1.36E-013±6.72E-015
	Griewank	5.51E-003±7.47E-003	0.00E+000±0.00E+000	0.00E+000±0.00E+000
	Weierstrass	0.00E+000±0.00E+000	0.00E+000±0.00E+000	9.47E-016±5.19E-015
	Rastrigin	0.00E+000±0.00E+000	0.00E+000±0.00E+000	7.58E-015±2.88E-014
	Schwefel	1.27E-004±0.00E+000	6.36E-004±0.00E+000	1.27E-003±3.49E-012
50	Sphere	0.00E+000±0.00E+000	0.00E+000±0.00E+000	0.00E+000±0.00E+000
	Rosebrock	1.18E-002±6.32E-002	1.97E+001±4.03E+000	7.34E+001±1.80E+001
	Ackley	4.44E-015±0.00E+000	5.37E-014±3.46E-015	1.37E-013±4.03E-015
	Griewank	4.11E-004±2.25E-003	7.40E-018±4.05E-017	1.48E-017±4.82E-017
	Weierstrass	0.00E+000±0.00E+000	0.00E+000±0.00E+000	2.84E-015±8.67E-015
	Rastrigin	0.00E+000±0.00E+000	0.00E+000±0.00E+000	4.93E-014±5.73E-014
	Schwefel	1.27E-004±0.00E+000	6.36E-004±0.00E+000	1.27E-003±1.33E-012

B. Benchmark function in CEC 2013

The second experiment for PAABC algorithm was built on a group of benchmark function in CEC 2013[18]. The CEC'13 test suite which includes 28 benchmark functions is composed of unimodal, multimodal, separable, non-separable, rotated and hybrid composition functions. The detailed description about the benchmark functions can be seen in [18]. The parameter settings were arranged as follows:

The population size and Limit are 50 and 200 respectively for the all 28 functions. The dimensions should be specified as 10 and 30. The algorithm was terminated when the number of cycles reach the MaxFES or the value of error function was less than or equal to1e-8. The 'Max.FES' in [18] is equal to 10000*D for different dimension. In the cause of fairness, the algorithm proposed in this paper was run 51 times for every function. Sorting the error values achieved after MaxFES in 51 runs from the smallest (best) to the largest (worst), the best, worst, mean, median and standard were presented in table IV – table V.

TABLE IV.
D:10, POPULATION SIZE:50, "LIMIT":200, MAX.FES: 10000*D SORTED ERROR VALUES OF 51 RUNS

Prob	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Best	0.00E+000	3.69E+005	3.79E+005	5.07E+003	0.00E+000	7.85E-003	8.32E+000
Worst	0.00E+000	8.59E+006	1.35E+008	1.69E+004	0.00E+000	9.83E+000	4.31E+001
Median	0.00E+000	3.45E+006	1.68E+007	9.49E+003	0.00E+000	5.13E-001	2.10E+001
Mean	0.00E+000	3.29E+006	7.91E+006	9.29E+003	0.00E+000	6.39E-002	2.01E+001
Std	0.00E+000	2.04E+006	2.52E+007	2.49E+003	0.00E+000	1.92E+000	7.66E+000
	8	9	10	11	12	13	14
Best	0.00E+000	2.54E+000	6.78E-001	0.00E+000	4.59E+000	5.77E+000	0.00E+000
Worst	0.00E+000	7.05E+000	2.01E+000	0.00E+000	2.31E+001	3.04E+001	1.87E-001
Median	0.00E+000	5.05E+000	1.27E+000	0.00E+000	1.49E+001	2.05E+001	6.86E-002
Mean	0.00E+000	5.16E+000	1.24E+000	0.00E+000	1.46E+001	2.06E+001	6.25E-002
Std	0.00E+000	9.30E-001	2.83E-001	0.00E+000	3.88E+000	5.70E+000	6.02E-002
	15	16	17	18	19	20	21
Best	2.91E+002	3.84E-001	0.00E+000	2.08E+001	3.16E-007	2.15E+000	1.00E+002
Worst	1.19E+003	1.39E+000	1.01E+001	4.10E+001	3.16E-002	3.53E+000	4.00E+002
Median	8.25E+002	1.03E+000	9.14E+000	3.37E+001	7.29E-003	3.13E+000	2.57E+002
Mean	8.46E+002	1.08E+000	1.01E+001	3.41E+001	1.78E-003	3.18E+000	2.00E+002
Std	1.76E+002	2.21E-001	3.01E+000	4.63E+000	8.46E-003	2.84E-001	1.02E+002
	22	23	24	25	26	27	28
Best	0.00E+000	6.01E+002	1.19E+002	1.13E+002	1.12E+002	2.15E+002	1.00E+002
Worst	2.39E+001	1.49E+003	1.43E+002	2.19E+002	1.38E+002	4.00E+002	3.00E+002
Median	7.21E+000	1.07E+003	1.29E+002	1.50E+002	1.23E+002	3.67E+002	2.41E+002
Mean	6.52E+000	1.07E+003	1.29E+002	1.45E+002	1.22E+002	3.67E+002	3.00E+002
Std	5.65E+000	1.85E+002	5.25E+000	2.68E+001	6.42E+000	3.73E+001	9.20E+001

TABLE V.
D:30, POPULATION SIZE:50, "LIMIT":200, MAX.FES: 10000*D SORTED ERROR VALUES OF 51 RUNS

Prob	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Best	0.00E+000	9.23E+006	1.59E+008	5.03E+004	0.00E+000	9.03E+000	7.06E+001
Worst	0.00E+000	2.84E+007	2.75E+009	8.74E+004	0.00E+000	2.79E+001	1.32E+002
Median	0.00E+000	1.80E+007	8.22E+008	6.96E+004	0.00E+000	1.67E+001	9.82E+001
Mean	0.00E+000	1.77E+007	6.82E+008	7.05E+004	0.00E+000	1.64E+001	9.76E+001
Std	0.00E+000	4.68E+006	5.54E+008	9.32E+003	0.00E+000	3.78E+000	1.51E+001
	8	9	10	11	12	13	14
Best	0.00E+000	2.46E+001	1.09E+000	0.00E+000	1.13E+002	1.65E+002	4.16E-002
Worst	0.00E+000	3.15E+001	2.88E+000	0.00E+000	2.08E+002	2.59E+002	1.16E+000
Median	0.00E+000	2.86E+001	1.56E+000	0.00E+000	1.64E+002	2.08E+002	1.61E-001
Mean	0.00E+000	2.88E+001	1.55E+000	0.00E+000	1.67E+002	2.11E+002	1.04E-001
Std	0.00E+000	1.62E+000	2.73E-001	0.00E+000	2.35E+001	2.16E+001	2.08E-001
	15	16	17	18	19	20	21
Best	3.24E+003	1.49E+000	3.04E+001	2.00E+002	5.48E-002	1.30E+001	2.00E+002
Worst	5.35E+003	2.53E+000	3.04E+001	2.93E+002	5.35E-001	1.48E+001	3.00E+002
Median	4.68E+003	2.05E+000	3.04E+001	2.57E+002	1.71E-001	1.39E+001	2.51E+002
Mean	4.75E+003	2.05E+000	3.04E+001	2.59E+002	1.51E-001	1.39E+001	3.00E+002
Std	3.40E+002	2.24E-001	9.43E-007	2.19E+001	9.67E-002	4.32E-001	5.00E+001
	22	23	24	25	26	27	28
Best	6.98E+000	4.47E+003	2.69E+002	2.78E+002	2.01E+002	4.00E+002	1.00E+002
Worst	1.12E+002	6.18E+003	2.92E+002	3.02E+002	2.02E+002	4.02E+002	3.00E+002
Median	4.65E+001	5.46E+003	2.83E+002	2.94E+002	2.01E+002	4.00E+002	2.96E+002
Mean	1.43E+001	5.45E+003	2.83E+002	2.95E+002	2.01E+002	4.00E+002	3.00E+002
Std	4.70E+001	4.33E+002	4.62E+000	4.95E+000	2.46E-001	2.88E-001	2.80E+001

From the results in table IV and table V, the PAABC algorithm achieve the accuracy values on function 1, 5, 8, 11 in the case of D =10 and 30. Increasing with the dimension, the global error is no big change. The algorithm proposed in this paper has better robustness.

ACKNOWLEDGMENT

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 An-hui University (Grant No. KJQN1015), the University Natural Science Research Project of Anhui Province (Grant Nos.KJ2012B038).

REFERENCES

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

[2] Shah H, Ghazali R, Nawi NM "Using artificial bee colony algorithm for mlp training on earthquake time series data prediction", J Comput vol. 3, No. 6, pp:135–142, 2011.

[3] Sharma TK, Pant M "Differential operators embedded artificial bee colony algorithm". Int J Appl Evol Comput vol 2, No.3, pp:1–14, 2011. http://dx.doi.org/10.4018/jaec.2011070101

[4] Shukran MAM, Chung YY, YehWC,Wahid N, Zaidi AMA "Artificial bee colony based data mining algorithms for classification tasks". Mod Appl Sci vol. 5,No. 4, pp:217–231, 2011.

[5] Tahooneh A, Ziarati K "Using artificial bee colony to solve stochastic resource constrained project scheduling problem". In: Tan Y, Shi Y, Chai Y, Wang G (eds) Advances in swarm intelligence. Lecture notes in computer science, vol. 6728. Springer, Berlin, pp: 293–302,2011. http://dx.doi.org/10.1007/978-3-642-21515-5_35

[6] D. Karaboga, B. Basturk, "A comparative study of artificial bee colony algorithm", Applied Mathematics and Computation, Vol. 214, pp.108–132, 2009. http://dx.doi.org/10.1016/j.amc.2009.03.090

[7] B. Akay, D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization", Information Sciences, Vol. 192, pp. 120-142, 2010 http://dx.doi.org/10.1016/j.ins.2010.07.015

[8] B. Akay, D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization", Information Sciences, Vol. 192, 2010, pp. 120-142. http://dx.doi.org/10.1016/j.ins.2010.07.015

[9] Weifeng Gao, Sanyang Liu, Lingling Huang, "A global best artificial bee colony algorithm for global optimization", Journal of Computational and Applied Mathematics, Vol. 236, pp: 2741-2753, 2012. http://dx.doi.org/10.1016/j.cam.2012.01.013

[10] Jianfeng Qiu, Jiwen Wang, Dan Yang, Juan Xie," An Artificial Bee Colony Algorithm With Modified Search Strategies for Global Numerical Optimization", Journal of Theoretical and Applied Information Technology, Vol, 48, No. 1, 2013.

[11] G.P. Zhu, S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization", Applied Mathematics and Computation, Vol. 217, No.7, pp. 3166–3173, 2010. http://dx.doi.org/10.1016/j.amc.2010.08.049

[12] P. J. Angeline, "Adaptive and self-adaptive evolutionary computations," in *Computational Intelligence: A Dynamic Systems Perspective*. 1995, pp. 152–163.

[13] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 124–141, Jul. 1999. http://dx.doi.org/10.1109/4235.771166

[14] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing. Natural Computing*. New York: Springer, 2003. http://dx.doi.org/10.1007/978-3-662-05094-1

[15] Alam, Mohammad Shafiul Islam, MM, "Artificial Bee Colony algorithm with Self-Adaptive Mutation: A novel approach for numeric optimization ", TENCON 2011 IEEE Region 10 Conference, pp: 49-53

[16] Wenxiang Gu, Minghao Yina, Chunying Wang, "Self Adaptive Artificial Bee Colony for Global Numerical Optimization", 2012 2nd International Conference on Mechanical, Industrial, and Manufacturing Engineering, Vol 1, pp: 59-65,2012.

[17] Gao, W. F, Liu, S. Y, "Improved artificial bee colony algorithm for global optimization", Information Processing Letters, vol. 111, NO. 17,pp: 871-882, 2011.

[18] J. J. Liang, B. Y. Qu, P. N. Suganthan, Alfredo G. Hernández-Díaz," Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization" Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China And Technical Report, Nanyang Technological University, Singapore, December, 2012.

[19]] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE

AUTHORS

Jianfeng Qiu is with the School of Computer Science and Technology, Anhui University, 230039 Chian (e-mail: qiujianf@ ahu.edu.cn).

JIWEN WANG, was with he School of Computer Science and Technology, Anhui University, 230039 Chian (e-mail: qiujianf@ ahu.edu.cn) (e-mail: wjw@ahu.edu.cn).

This article is an extended and modified version of a paper presented at the International Conference on Mechanical Engineering, Automation and Material Science (MEAMS2012), held 22-23 December 2012, Wuhan, China. Received 20 March 2013. Published as resubmitted by the authors 01 May 2013.