Differential Artificial Bee Colony Algorithm for Global Numerical Optimization

Bin Wu and Cun hua Qian
College of Management Science and Engineering, Nanjing University of Technology, Nanjing, China
Email: seasimen@yahoo.com.cn

***Abstract*—artificial bee colony (ABC) is the one of the newest nature inspired heuristics for optimization problem. In order to improve the convergence characteristics and to prevent the ABC to get stuck on local solutions, a differential ABC (DABC) is proposed. The differential operator obeys uniform distribution and creates candidate food position that can fully represent the solution space. So the diversity of populations and capability of global search will be enhanced. To show the performance of our proposed DABC, a number of experiments are carried out on a set of well-known benchmark continuous optimization problems. Simulation results and comparisons with the standard ABC and several meta-heuristics show that the DABC can effectively enhance the searching efficiency and greatly improve the searching quality.**

***Index Terms*—Artificial Bee Colony Algorithm, Global Numerical Optimization, Differential Evolution**

I. Introduction

The Artificial Bee Colony (ABC) algorithm is a new swarm intelligence technique inspired by intelligent foraging behavior of honey bees. The first framework of ABC algorithm mimicking the foraging behavior of honey bee swarm in finding good solutions to optimize multi-variable and multi-modal continuous functions was presented by D.Karaboga. Numerical comparisons demonstrated that the performance of the ABC algorithm is competitive to other population-based algorithm with an advantage of employing fewer control parameters [1], [2], [3]. The basic version of the Artificial Bee Colony algorithm has only one control parameter ‘‘limit” apart from the common control parameters of the population-based algorithms such as population size or colony size (SN) and maximum generation number or maximum cycle number (MCN).Due to its simplicity and ease of implementation, the ABC algorithm has captured much attention and has been applied to solve many practical optimization problems. Singh [4] used the ABC algorithm for the leaf-constrained minimum spanning tree problem and compared the approach against genetic algorithm (GA), ant colony optimization (ACO) and tabu search (TS). Pan et.al [5] proposed a discrete ABC algorithm in combinational optimization for flow shop scheduling problem. Feng et.al [6] proposed a hybrid simplex artificial bee colony algorithm which combines Nelder-Mead simplex method to solve inverse analysis problems. Rao et al. [7] applied the ABC algorithm to network reconfiguration problem in a radial distribution system in order to minimize the real power loss, improve voltage profile and balance feeder load subject to the radial network structure in which all loads must be energized. The ABC algorithm also was extended for constrained optimization problems in [8] and was applied to train neural networks [9], to medical pattern classification and clustering problems [10,11], and to solve TSP problems [12].

Similar to other swarm based optimization algorithms, it is important to establish a proper balance between exploration and exploitation in bee swarm optimization approaches. A poor balance between exploration and exploitation may result a weak optimization method which may suffer from premature convergence, trapping in a local optima, and stagnation. In order to enhance the global convergence and to prevent to stick on a local solution of the ABC, some improvement algorithms were proposed. Haijun and Qingxian [13] proposed a modification in the initialization scheme by making the initial group symmetrical, and the Boltzmann selection mechanism was employed instead of roulette wheel selection for improving the convergence ability of the ABC algorithm Quan and Shi [14] integrated a search iteration operator based on the fixed point theorem of contractive mapping in Banach spaces with the ABC algorithm in order to improve convergence rate. In order to maximize the exploitation capacity of the onlooker stage, Tsai et al.[15]introduced the Newtonian law of universal gravitation in the onlooker phase of the basic ABC algorithm in which onlookers are selected based on a roulette wheel. Baykasoglu et al [16].was incorporated the ABC algorithm with shift neighborhood searches and greedy randomized adaptive search heuristic and applied it to the generalized assignment problem. Bilal Alatas[17] used different chaotic maps for parameter adaptation in order to improve the convergence characteristics and to prevent the ABC to get stuck on local solutions.

In the paper, differential operator incorporated into the ABC algorithm is proposed to efficiently control the global search and convergence to the global best solution. Simulation results and comparisons demonstrate the effectiveness and efficiency of the proposed DABC. The remaining of this paper is organized as follows: the basic ABC algorithm is introduced in Section 2. Section 3 describes the proposed methods, Differential Artificial Bee Colony Algorithms, shortly DBCAs. In Section 4, the benchmark problems used for comparisons are

doi:10.4304/jcp.6.5.841-848

described and the simulation results are compared. Finally, the conclusions are presented in Section 5.

II. ARTIFICIAL BEE COLONY ALGORITHM

In ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. A bee waiting on the dance area for making a decision to choose a food source is called onlooker and one going to the food source visited by it before is named employed bee. The other kind of bee is scout bee that carries out random search for discovering new sources. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. In the algorithm, the first half of the colony consists of employed artificial bees and the second half constitutes the onlookers. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population.

At the first step, the ABC generates a randomly distributed initial population of *SN* solutions (food source positions), where *SN* denotes the size of population. Each solution x_i where $i = 1, 2, \cdots SN$ is a *D*-dimensional vector. Here, *D* is the number of optimization parameters. After initialization, the population of the positions (solutions) is subjected to repeated cycles, $C = 1, 2 \cdots MCN$ of the search processes of the employed bees, the onlooker bees and scout bees. An employed bee produces a modification on the position (solution) in her memory depending on the local information (visual information) and tests the nectar amount (fitness value) of the new source (new solution). Provided that the nectar amount of the new one is higher than that of the previous one, the bee memorizes the new position and forgets the old one. Otherwise she keeps the position of the previous one in her memory. After all employed bees complete the search process; they share the nectar information of the food sources and their position information with the onlooker bees on the dance area. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount. As in the case of the employed bee, she produces a modification on the position in her memory and checks the nectar amount of the candidate source. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one. An artificial onlooker bee chooses a food source depending on the probability value associated with that food source p_i, calculated by the following expression (1):

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

Where fit_i is the fitness value of solution *i*. In order to produce a candidate food position from the old one in memory, the ABC uses the following expression (2):

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

Where $k \in \{1, 2, \cdots SN\}$ and $j \in \{1, 2, \cdots D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. ϕ_{ij} is a random number between [−1, 1]. It controls the production of neighbor food sources around x_{ij} and represents the comparison of two food positions visible to a bee. As can be seen from (2), as the difference between the parameters of the x_{ij} and x_{kj} decreases, the perturbation on the position x_{ij} decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is adaptively reduced.

After each candidate source position is produced and evaluated by the artificial bee, its performance is compared with that of its old one. If the new food source has equal or better quality than the old source, the old one is replaced by the new one. Otherwise, the old one is retained. If a position cannot be improved further through a predetermined named "*limit*", then that food source is assumed to be abandoned. The corresponding employed bee becomes a scout. The abandoned position will be replaced with a new food source found by the scout. Assume that the abandoned source x_i, then the scout discovers a new food source to be replaced with x_i. This operation can be defined as in (3):

$$x_i^j = x_{\min}^j + rand()(x_{\max}^j - x_{\min}^j) \qquad (3)$$

Where $x_{\min}^j$ and $x_{\max}^j$ are lower and upper bounds of parameter j, respectively.

Detailed pseudo-code of the ABC algorithm is given below:

Step1: Initialize the population of solutions $x_{ij}(i = 1, 2 \cdots SN, j = 1, 2, \cdots D)$. Evaluate the population, and cycle=1.
Step2: Repeat
Step3: Produce new solutions v_{ij} for the employed bees by using (2) and evaluate them, then apply the greedy selection process.
Step4: Calculate the probability values p_i for the solutions x_i by (1),
Step5: Produce the new solutions v_{ij} for the onlookers from the solutions x_{ij} selected depending on p_i and evaluate them, then apply the greedy selection process
Step6: Determine the abandoned solution for the scout, if exists, and replace it with a new randomly produced solution x_{ij} by (3).And memorize the best solution achieved so far.
Step7: cycle=cycle+1, until cycle=MCN.

Fig.1. The pseudo-code of the ABC algorithm

III. Differential Artificial Bee Colony Algorithm (DABC)

Exploration, which is the ability to search the solution space to find promising new solutions, and exploitation, which is the ability to find the optimum solution in the neighborhood of a good solution, is two important aspects in evolutionary computing paradigms. However, different algorithms in evolutionary computing employ different operators for exploration and exploitation. In a bee swarm, different behaviors of the bees provide this possibility to establish powerful balancing mechanism between exploration and exploitation. This property provides the opportunity to design more efficient algorithms in comparison with other population based algorithms such as PSO and GA. The performance of ABC is very good in terms of the local and the global optimization due to the selection schemes employed and the neighboring production mechanism used. But in ABC, the employed bees and onlooker bees carry out exploration and exploitation use the same formula (2). Obviously, the performance of ABC greatly depends on formula (2). To enrich the searching behavior and to avoid being trapped into local optimum, differential operator is incorporated into the ABC.

Differential Evolutionary (DE) algorithm is a population-based evolutionary computation technique, which uses simple differential operator to create new candidate solutions and one-to-one competition scheme to greedily select new candidate. In DE, it starts with the random initialization of a population of individuals in the search space and works on the cooperative behaviors of the individuals in the population. Therefore, it finds the global best solution by utilizing the distance and direction information according to the differentiations among population. However, the searching behavior of each individual in the search space is adjusted by dynamically altering the differentiation's direction and step length in which this differentiation performs.

In DE, the *i*th individual in the *D*-dimensional search space can be represented as $x_i = \{x_{i1}, x_{i2}, \cdots x_{iD}\}$, the differential mutation is described by the following equation:

$$v_i = x_a + F(x_b - x_c) \tag{4}$$

Where $a,b,c \in SN$ are randomly chosen and mutually different and also different from the current index *i*. $F \in [0,2]$ is constant called scaling factor which controls amplification of the differential variation of $x_{bj} - x_{cj}$.

The differential mutation can produce uniform distribution of the solution. This property can be proved as follows:

General, suppose individuals obey uniform distribution in (0,1). $x_1, x_2, \cdots x_n$ is independent random variables in (0,1), $c_1, c_2, \cdots c_n$ is constant value.

Lemma.1.[18] $X = \sum_{i=1}^{n} c_i x_i$, $X \sim U(0,1)$ if and only if $x_i \sim U(0,1)$ and c_1 is integer.

Theorem.1. the candidate individuals obey uniform distribution after differential mutation.

Proof: for individual $x_i (i = 1, 2, \cdots SN)$, the candidate individual v_i is calculated by (4).Then,

$$v_i = x_a + Fx_b - Fx_c$$

Since $a,b,c \in SN$ are different from *i*, and F is constant, $x_a, x_b, x_c \sim U(0,1)$, $c_1 = 1, c_2 = F, c_3 = -F$. According to Lemma.1., $v_i \sim U(0,1)$.

Theorem.2. the candidate food positions that employed bees or onlooker bees find don't obey uniform distribution.

The proof is similar to the above. Since, $c_1 = 1 + \phi_{ij}$ isn't an integer, it doesn't obey Lemma.1.

The neighboring solution production mechanism used in ABC is similar to the self-adapting mutation process of DE. From this point of view, in DE and ABC algorithms, the solutions in the population directly affect the mutation operation since the operation is based on the difference between them. However, in DE, the difference is weighted by a constant scaling factor while in ABC; it is weighted by a random step size. Unlike DE, in ABC, there is no explicit crossover and there is only one dimensional position adjusted for each individual in every cycle. In DE, the *D*-dimensional positions are adjusted in every generation and there is no operation as in the scout bee's phase of ABC to insert a random solution into the population during a search. Therefore, although the local convergence speed of a standard DE is quite good and the ability of keeping uniform distribution of the solution is very well, it might result in the premature convergence in optimizing multimodal problems. In [19] proposed that the uniform distribution can fully express the solution space characteristics and enhance the diversity of the population. For ABC, the neighboring solution production mechanism that doesn't produce uniform distribution of the solution can result premature convergence, trapping in a local optima, and stagnation. In order to enhance the global convergence and to prevent to stick on a local solution of the ABC, the improvement differential operator is incorporated into ABC. This operation can be defined as in (5):

$$v_{ij} = x_{aj} + F(x_{bj} - x_{cj}) \tag{5}$$

The employed bees and onlooker bees produce candidate food position according (2) and (5). Then, there are two version of DABC. The employed bees carry out the differential operator, called EDABC or the onlooker bees carry out the differential, called LDABC. Which is more efficiently, it will be discussed in the next section. But, if they all apply the differential operator, the algorithm becomes DE.

Ⅳ.EXPERIMENTS

In this section, the experiments that have been done to evaluate the performance of the proposed DABC algorithm and its variants for a number of analytical benchmark functions are described. The DABC is coded in Visual C++ 6.0 and experiments are executed on a Pentium E2200 CPU PC with 2G RAM. Each benchmark is independently run with every algorithm 30 times for comparison. The mean value (Mean), minimum value (Min), the standard deviation (Dev) and the succeed ratio (SR) in 30 runs are calculated as the statistics for the performance measures.

A Benchmark Functions

Well-defined benchmark functions which are based on mathematical functions can be used as objective functions to measure and test the performance of optimization methods. The nature, complexity and other properties of these benchmark functions can be easily obtained from their definitions. The difficulty levels of most benchmark functions are adjustable by setting their parameters. To test the performance of DARC, six well known benchmark functions are presented in Table 1. Initial range, formulation, properties and global optimum values are listed in table. The first two functions are unimodal, while others are multimodal. A function is called unimodal, if it has only one optimum position. The multimodal functions have two or more local optima.

The Sphere function is a continuous, convex and unimodal function. The Rosenbrock function has smooth slope around its' global optimum position, its global optimum lays inside a long, narrow, and parabolic shaped flat valley, its variables are strongly dependent, and the gradient do not point towards its optimum position. All of these cause that the convergence toward the global optimum in Rosenbrock be relatively difficult. The variables of Griewank function have interdependence since the function has a product term. The multimodality is removed by the increment in dimensionality (D > 30) and the problem seems unimodal. Rastrigin function is based on the Sphere function with the addition of cosine modulation to produce many local minima. The surface of Schwefel function is composed of numerous peaks and valleys. The Ackley function has a surface with many local optima due to its exponential term.

B Experiment1: Parameters Discussion

As we know, the performance of population based meta-heurist greatly depends on the control parameters. So, we investigate the effect of parameters on the performance of DABC by manually trying some different values. DABC algorithm has a few control parameters: the colony size (*SN*), the maximum number of cycles (*MCN*), "*limit*" and scaling factor (*F*). The dimensionality of the search space is an important issue for the performance of the algorithm. In order to analyze the robustness of DABC, we investigated the performance of DABC with respect to growing dimensions. The parameters are set in table 2.

The LDABC was tested in the experiments. Firstly, we investigate the performance of LDABC with different population size *SN* for Rastrigin function. The effects with different *SN* on the mean values, the standard deviation and succeed ratio are illustrated in Fig.2. As shown in Fig. 2, when *SN* is too small, the average searching quality is poor because the solution space may not be covered sufficiently during the evolution process. As *SN* increases, the results become better at a cost of more fitness evaluations, but there is a threshold beyond which the results will not be affected in a significant manner. Therefore, considering both the searching quality

TABLE I. NUMERICAL BENCHMARK FUNCTIONS

Name	Formulation	Property	Search range	Opt
Sphere	$f1=\sum_{i=1}^{n}x_i^2$	Unimodal	[-5.12,5.12]	0
Rosenbrock	$f2=\sum_{i=1}^{n-1}(100(x_{i+1}-x_i^2)+(x_i-1)^2)$	Unimodal	[-30,30]	0
Griewank	$f3=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	Multimodal	[-600,600]	0
Rastrigin	$f4=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$	Multimodal	[-5.12,5.12]	0
Schwefel	$f5=418.9829n+\sum_{i=1}^{n}-x_i\sin(\sqrt{\lvert x_i\rvert})$	Multimodal	[-500,500]	0
Ackley	$f6=20+e-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))$	Multimodal	[-32,32]	0

TABLE II. THE PARAMETERS VALUE

Name	Value				
SN	10	20	30	40	50
	60	70	80	90	100
MCN	500	1000	2000	3000	4000
	5000	6000	7000	8000	9000
	10000				
limit	20	50	80	100	150
	200	300			
F	0.01	0.05	0.1	0.2	0.3
	0.4	0.5	0.6	0.7	0.8
	0.9	1	1.5	2	
D	10	20	30	40	50
	60	70	80	90	100

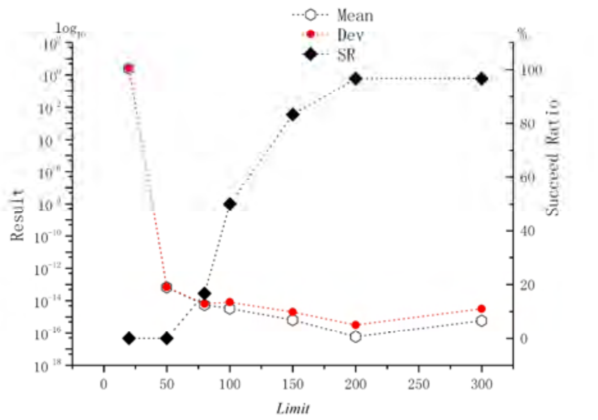

Fig.4. Result of LDABC with different limit for Rastrigin

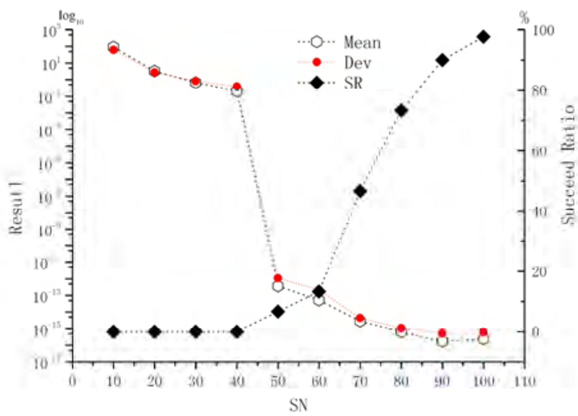

Fig.2. Result of LDABC with different *SN* for Rastrigin

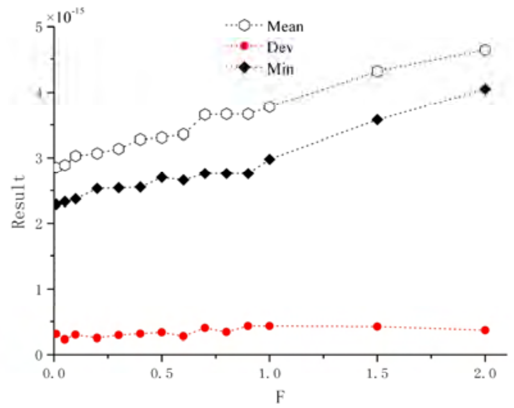

Fig.5. Result of LDABC with different *F* for Sphere

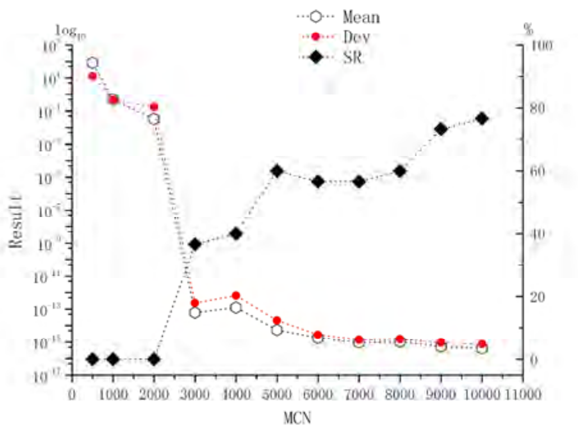

Fig.3 Result of LDABC with different *MCN* for Rastrigin

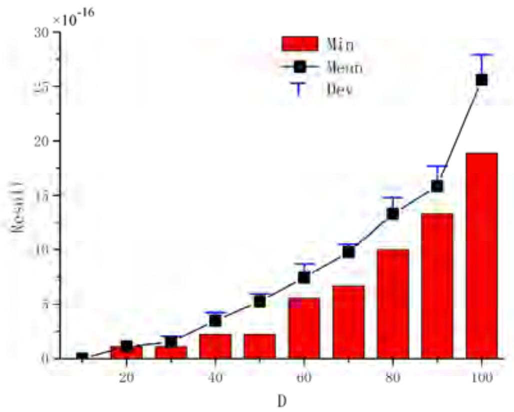

Fig.6. Result of LDABC with different *D* for Griewank

and computational efforts, it is recommended to choose *SN* between 70 and 100. Similar to *SN*, the effects with different *MCN* are illustrated in Fig.3. However, the effects of *MCN* are weaker than the effects of *SN*. The suitable scope of *MCN* is 5000 to 10000.The scout bee production is controlled by the control parameter "*limit*"; there is an inverse proportionality between the value of "*limit*" and the scout production frequency .As the value of "*limit*" approaches to infinity, the total number of the scouts produced goes to zero. So, the "*limit*" isn't very big. But, if the "*limit*" is too small, the scout production frequency is very high; the employed and onlooker bees haven't enough time carry out exploration and exploitation. The effects with different "*limit*" are illustrated in Fig.4. It is recommended to choose "*limit*" between 150 and 200.

Fig.5 shows the effect with different *F* for Sphere function. The mean and min values become worse with *F* increasing, but the standard deviation keeps stably. The reason is that *F* controls amplification of the differential variation, if it too big, it will affect the exploitation of the employed or onlooker bee. So, the proper range of *F* is 0.01 to 0.2.

The performance of LDABC with different dimensions for Griewank is illustrated in Fig.6.As can be seen from the graph, when the problem dimension was increased from 10 to 100, the performances of the DABC algorithm were influenced from this change as expected. However, an increase in problem dimension did not lead to exponential increment in mean value and minimum value. Therefore, it can be stated that the DABC algorithm is not very sensitive to increments in problem dimensions and has a good robustness.

So, in the follow experiments, all the benchmark functions were used in 100 dimensions, and the parameters were set as follow: *SN*=70, *MCN*=5000, "*limit*"=200, *F*=0.01.

C Experiment 2: EDABC vs LDABC

In this experiment, we compare ABC with EDABC and LDABC. The statistical performances are listed in Table 3. The results show that EDABC and LDABC can obtain competitive or better results than ABC for all benchmarks. Especially, it is clear from the result that for Rastrigin function, DABCs search the global best value and achieves 100% success rate. DABCs aren't only higher quality solution, but also more stable. This is occurred, due to using differential operator which maintain the diversity of the algorithm and express the solution space characteristics fully regardless of the type of the considered function. Furthermore, the Mean, the Min values obtained by EDABC are better than those obtained by LDABC for almost all benchmarks, which demonstrates the employed bees' behaviors have an impact on the ability of global search for ABC. Fig.7 and Fig.8 show that EDABC converge faster then LDABC algorithm for Rosenbrock and Ackley functions. Fig. 9 and Fig.10 show that EDABC has the ability of escaping the local optima after a long stagnation process for Sphere and Griewank functions. As a consequence, the EDABC algorithm produces better results than LDABC and ABC.

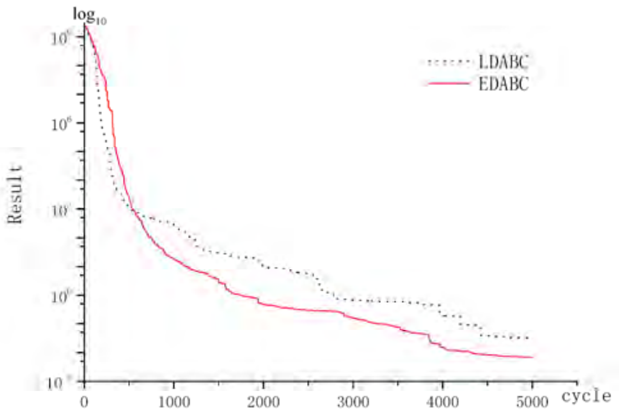

Fig.7. Evolution of min values for Rosenbrock

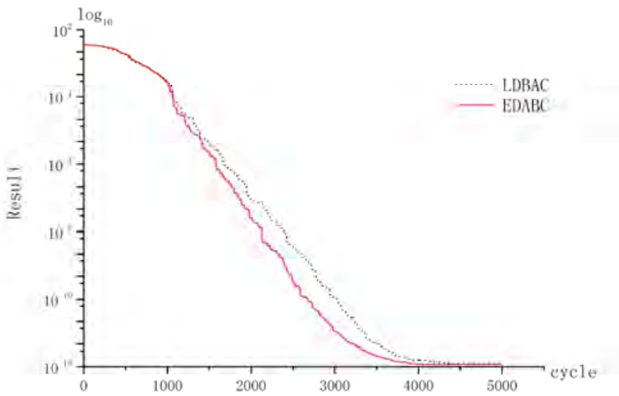

Fig.8. Evolution of min values for Ackley

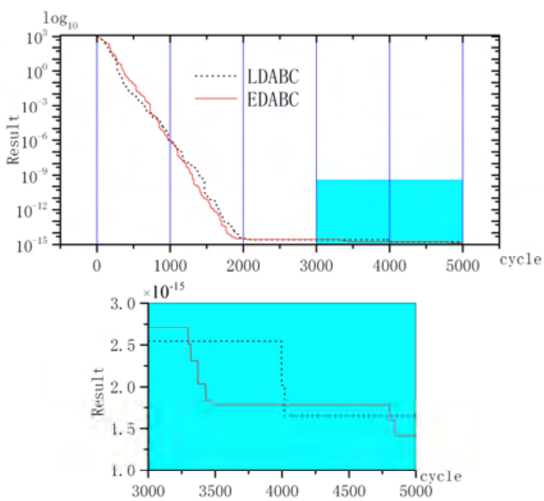

Fig.9. Evolution of min values for Sphere

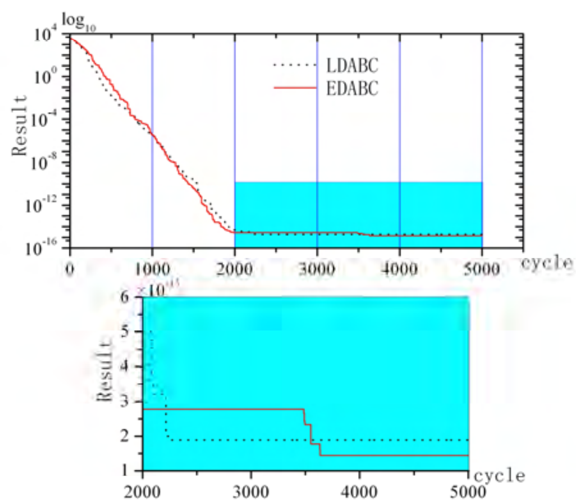

Fig.10. Evolution of min values for Griewank

TABLE III. THE RESULTS OBTAINED BY ABC ,LDABC AND EDABC ALGORITHMS

		Sphere	Rosenbrock	Griewank	Rastrigin	Schwefel	Ackley
ABC	Min	2.538E-15	1.138E-02	2.553E-15	1.776E-15	1.272E-03	2.956E-10
	Mean	3.018E-15	2.048E-01	3.271E-15	3.912E-13	3.339E+02	8.706E-10
	Dev	2.194E-16	2.914E-01	3.979E-16	7.876E-13	1.808E+02	6.524E-10
LDABC	Min	1.646E-15	0.236E-02	1.887E-15	0	1.27E-03	1.237E-13
	Mean	2.371E-15	6.248E-02	2.560E-15	0	1.27E-03	1.509E-13
	Dev	2.496E-16	2.517E-02	2.292E-16	0	0	9.722E-15
EDABC	Min	1.414E-15	6.822E-03	1.443E-15	0	1.27E-03	1.273E-13
	Mean	1.873E-15	4.421E-02	1.891E-15	0	1.27E-03	1.461E-13
	Dev	1.815E-16	1.532E-02	2.275E-16	0	0	8.865E-15

TABLE IV. THE RESULTS OBTAINED BY DABC AND ABC ALGORITHMS

		EDABC	GA[3]	PSO[3]	DE[3]	EDA[20]
Sphere	mean	0	1.11E+03	0	0	0
	dev	0	76.561450	0	0	0
Rosenbrock	mean	4.421E-02	1.96E+05	15.088617	18.203938	-------
	dev	1.532E-02	3.85E+04	24.170196	5.036187	-------
Griewank	mean	0	10.63346	0.01739118	0.0014792	0
	dev	0	1.161455	0.020808	0.002958	0
Rastrigin	mean	0	52.92259	43.9771369	11.716728	19.7405
	dev	0	4.564860	11.728676	2.538172	2.4246
Schwefel	mean	1.272E-03	976.1	5560.3641	2303.5	1964.9
	dev	0	93.254240	457.957783	521.849292	120.2
Ackley	mean	0	14.67178	0.16462236	0	6.089E-12
	dev	0	0.178141	0.493867	0	0

D. Experiment 3: Comparison with Other Algorithms

Results of EDABC algorithm have been compared with the results presented by D. Karaboga et al. [3] of Differential Evolution (DE), Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Cheng Yu-hu et.al [20] of Estimation of Distribution Algorithm (EDA). In [3], the dimensions were set 30 and values less than E-12 were reported as 0. The mean and the standard deviations of the function values found are given in Table4. The results show that all the algorithms provide good performance for Sphere function except GA; On Griewank, while EDABC and EDA showed equal performance and found the optimum, PSO, DE, GA demonstrated worse performance than them. On Ackley function, EDABC and DE have better performance than other algorithm. However EDABC strongly produce better results than other algorithm on Rosenbrock, Rastrigin and Schwefel. It is clear from the result that for Rastrigin function, the global optimum value was found by EDABC. For all the benchmark functions, the EDABC algorithm outperforms other methods. To sum up, EDABC is a very efficient and effective algorithm with excellent quality and robustness for unimodal, multimodal functions.

Ⅴ.CONCLUSION

In this work, a modified version of the Artificial Bee Colony algorithm was proposed, which has been embedded to enhance the global searching capability by differential operator. In order to verify the feasibility and the performance of the proposed algorithm, six high dimensional numerical benchmark functions were tested. Comparing the performances with other meta-heurists presented in the literature. From the simulation results it was concluded that the proposed algorithm is superior to ABC in term of searching quality and efficiency. Besides, the searching quality of DABC is better than other meta-heuristics, such as GA, PSO, DE, and EDA. The future work is to theoretically investigate the effect of DE incorporating into ABC and apply the DABC for some real engineering optimization problems.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (Grant No. 70801036) and Natural Science Foundation of Jiangsu Province (Grant No. BK2010555).

REFERENCES

[1] D. Karaboga, B. Basturk, On the performance of artificial bee colony algorithm, Applied Soft Computing , Vol.8,No.1, pp687–697,2008.

[2] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm, Journal of Global Optimization, Vol.39,No.3,pp459-471,2007.

[3] D.Karaboga, Bahriye Akay, A comparative study of Artificial Bee Colony algorithm, Applied Mathematics and Computation, Vol.214,No.1 ,pp108–132,2009.
[4] A. Singh, An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem, Applied Soft Computing Journal, Vol.9, No.2, pp625-631,2009.
[5] Quan-Ke Pan, M. Fatih Tasgetiren,et.al A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem, Information Sciences, online, in press
[6] F. Kang, J. Li and Q. Xu,Structural inverse analysis by hybrid simplex artificial bee colony algorithms, Computers and Structures,Vol.87,No.13,pp861-870,2009.
[7] .R.S. Rao, S.V.L.Narasimham, M.Ramalingaraju, Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm, International Journal of Electrical Power and Energy Systems Engineering, Vol.1, No.2 ,pp 116–122,2008.
[8] D. Karaboga, B. Basturk, Advances in soft computing: foundations of fuzzy logic and soft computing, in: Artificial Bee Colony Optimization Algorithm for Solving Constrained Optimization Problems, LNCS, 4529, Springer-Verlag, 2007, pp. 789–798.
[9] Karaboga, D; Ozturk, C, Neural networks training by artificial bee colony algorithm on pattern classification , Neural Network World, Vol.19,No.3,pp279-292,2009.
[10] T. Kurban and E. Besdok, A comparison of RBF neural network training algorithms for inertial sensor based terrain classification, Sensors, Vol.9, No.8, pp6312-6329, 2009.
[11] Zhang, CS; Ouyang, DT; Ning, JX, An artificial bee colony approach for clustering,, Expert Systems with Application,Vol.37,No.7, pp4761-4767,2010.
[12] Z.-H. Hu and M. Zhao, Simulation on traveling salesman problem (TSP) based on artificial bees colony algorithm, Transaction of Beijing Institute of Technology, Vol.29, No.11, pp978-982, 2009.
[13] D. Haijun, F. Qingxian, Artificial bee colony algorithm based on boltzmann selection strategy, Computer Engineering and Applications,Vol.45,No.32, pp53-55, 2009.
[14] H. Quan, X. Shi, On the analysis of performance of the improved artificial-bee-colony algorithm, in: Fourth IEEE International Conference on Natural Computation, ICNC (Jinan, China), August2008,pp654-658.
[15] P.-W. Tsai, J.-S. Pan et.al, Enhanced artificial bee colony optimization, International Journal of Innovative Computing, Information and Control, Vol.5, No.12, pp5081-5092, 2009.
[16] A. Baykasoglu, L. Ozbakir, P. Tapkan, Swarm intelligence focus on ant and particle swarm optimization, Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem, I-Tech Education and Publishing, Vienna, Austria, 2007, pp. 113–144.
[17] Bilal Alatas, Chaotic bee colony algorithms for global numerical optimization, Expert Systems with Applications, Vol.37, pp5682–5687, 2010.
[18] Deng L Y,Geogre E O. Generation of uniform variate from several nearly uniformly distributed variables.Communications in Statistics: Simulation and Computation,Vol.19,No.1,pp145–154,1990.
[19] Cai Xiaofen, Zhong Shounan. Evolutionary uniform optimization algorithm, Journal of Mathematics, Vol.25,No.3,pp349–354,2005.
[20] Cheng Yu-hu, Wang Xue-song, Hao Ming-lin, An Estimation of Distribution Algorithm with Diversity Preservation(In Chinese), Acat Electronica Sinica, Vol.38,No.3,pp591-596,2010.

Bin Wu is a Assistant Professor at College of Management Science and Engineering, Nanjing University of Technology, China. He received his BS degree in Automation from Henan University, Kaifeng, China, in 2002, MS degree in Mechanical Manufacturing and Automation in 2004 and Ph.D. degree in Control Theory and Engineering in 2008, both from Zhejiang University of Technology, Hangzhou, China.

He is currently interested in Combinatorial Optimization, Evolutionary Computation and Supply Chain Management. His papers have been cited by SCI, EI and ISTP more than 10 articles and have been published in Systems Engineering – Theory& Practice, Control and Decision and CIMS.

Cunhua Qian is a Professor at College of Management Science and Engineering, Nanjing University of Technology, China. He received his BS degree in Mathematics from Yangzhou University, Yangzhou, China, in 1984, MS degree in Industrial Engineering in 1998 and Ph.D. degree in reliability in 2001, both from Aichi Institute of Technology, Toyota, Japan. He is currently interested in stochastic processes and reliability. His papers have been published in Journal of the Operations Research Society of Japan, RAIRO Operations Research, Computers and Mathematics with Applications, Journal of Quality in Maintenance Engineering and other journals. Prof. Qian is a member of ORSJ.