Improved Artificial Bee Colony Algorithm Guided by Experience

Chunfeng Wang, Pengpeng Shang, and Lixia Liu

***Abstract*—At present, artificial bee colony algorithm (ABC) is one of the hot issues in swarm intelligence algorithm. Since it was proposed, people have done a lot of improvement work for ABC algorithm. To address the shortcomings of ABC, an improved ABC guided by experience (named as EABC) is proposed in this paper. In EABC, it collects the experience of individual improvement caused by dimension change in the iterative process, and selects the dimensions to change according to a ratio when a new position needs to be generated. In this way, the individual can choose a good direction to improve its quality. Numerical experiments show that EABC has a better performance.**

***Index Terms*—Artificial bee colony; Optimization; Swarm intelligence; Direction information**

I. INTRODUCTION

BY simulating animal behavior, many swarm intelligence algorithms have been proposed, including Simulated Annealing (SA) [1], Differential Evolution (DE) [2], Flower Pollination Algorithm (FPA) [3], Genetic Algorithm (GA) [4,5], Particle Swarm Optimization (PSO) [6], Firework Algorithm (FA) [7], Ant Colony Optimization (ACO) [8], and Artificial Bee Colony (ABC) [9] etc.

According to the real behavior of bees in nature, Karaboga [9] presented ABC and used it to deal with numerical optimization problems. To find nectar sources, in ABC, each type of bee performs its own task, and different types of bees work together through cooperation. After that, ABC has been widely used in different fields, such as data clustering [10], network planning [11], image segmentation [12], and numerical optimization [13,14]. However, researches showed ABC has powerful exploration ability, but its exploitation capability is weak, thus many improved versions of ABC have been proposed. For example, a modified version of ABC was introduced by Akay and Karaboga in 2012 [15]. In this algorithm, to improve the performance of ABC, two control parameters SF and MR were proposed. For obtaining better results for the optimization problem, Zhang et al. presented three modified versions of ABC [16]. To avoid ABC getting stuck local minimums, Alatas [17] proposed an ABC model based on chaotic map. By combining the information of the global best solution, Zhu and Kwong [18] designed an improved ABC algorithm (named GABC). By using opposition-based learning and chaotic strategies, Gao and Liu [19] proposed MABC by modifying search equation of the basic ABC. Based on the best-so-far selection for onlooker bees, Banharnsakun et al. [20] enhanced the convergence speed of ABC and verified the performance of their method on some benchmark problems. By using the Debs Rules, Karaboga and Akay [21] adopted the basic ABC to address constrained optimization problems. In onlooker bee phase, Kiran and Gunduz [22] presented a crossover technique to select neighbors. By using mutual learning, Liu et al. [23] proposed a variant of ABC algorithm which improved the performance of basic ABC. Based on two update techniques adopted from DE, Gao et al. [24] designed two ABC-based methods, which are named as ABC/best/1 and ABC/best/2. Later, for ABC/best/1, to avoid premature and fall into local optimal solution, Gao and Liu [25] suggested that the update rule of ABC/best/1 and the update rule of basic ABC should be used for employed and onlooker bees, respectively. In addition, Gao et al. [26] developed a new update rule for ABC, which looks like crossover operator in GA. By selecting some top best solutions from the whole population, Bajer and Zoric [27] designed a novel search technique. In their method, some new candidates are generated in the scout bee phase. Wang et al. [28] designed a new ABC on the basis of knowledge fusion (KFABC).

In this paper, we develop a new ABC variant named EABC with previous experience. As we know, experience is of great help to guide our next work, so each bee is endowed with the function of memory with search value dimension, which can enhance the exploitation ability of EABC. To very its performance, some numerical experiments are carried on.

The structure of this paper is designed as follows: Section I briefly introduces the research status of ABC. The process of basic ABC is explained in Section II, and the details of EABC is given in Section III. Experiment and related results are discussed in Section IV, and finally the conclusion and the next work are given in Section V.

II. BASIC ABC ALGORITHM

In 2005, to find food sources, Karaboga [9] first proposed ABC algorithm by simulating behavior of bees. In ABC, three types of bees: employed bees, onlooker bees and scout bees are included. Different kinds of bees carry out different search tasks, and they share experience with each other.

Initialization Assume that the number of employed bees and onlooker bees is equal to the number of food sources. At the beginning, each food source $x_i (i = 1, \cdots, N)$ is

Manuscript received May 16, 2021, revised December 15, 2021. This work was supported by the National Natural Science Foundation of China(11801430); the Key cultivation project of Xianyang Normal University (XSYK21044); the Key Project of Henan Educational Committee (20A110021).

Chunfeng Wang is a professor of the School of Mathematics and Statistics, Xianyang Normal University, Xianxiang, 712000, PR China; College of Mathematics and Information, Henan Normal University, Xinxiang, 453007, PR China, Email:wangchunfeng09@126.com

Pengpeng Shang is a graduate student of the College of Mathematics and Information Science, Henan Normal University, Xinxiang, 453007, PR China. Email: 783977427@qq.com

Lixia Liu is an associate professor of the School of Mathematics and Statistics, Xidian University, Xi'an 710126, PR China. Email: 15319789939@163.com

randomly produced as follows:

$$x_{ij} = l_j + r \times (u_j - l_j),\ j = 1, \ldots, D, \quad (1)$$

where x_{ij} represents the jth dimension of x_i, l_j and u_j represent the lower and upper bounds of the jth dimension, respectively, $r \in [0, 1]$ is a random number, N is size of the swarm, and D is the size of dimension. The quality of x_i is measured by using the following equation:

$$fit_i = \begin{cases} \frac{1}{1+f_i} & \text{if } f_i > 0 \\ 1 + |f_i| & \text{else,} \end{cases} \quad (2)$$

where fit_i and f_i are the fitness and the objective function value of the ith food source x_i, respectively.

Employed bees The task of the employed bees is to search around the current solution and try to find a better one. For ith employed bee, its new position v_i around x_i is generated below:

$$\begin{array}{c} v_{ij} = x_{ij} + \phi \times (x_{ij} - x_{kj}), \\ j \in \{1, \cdots, D\}, k \in \{1, \cdots, N\}, \end{array} \quad (3)$$

where $x_k (x_k \neq x_i)$ is selected randomly from the whole population, $j \in [1, D]$ is a random integer, and $\phi \in [-1, 1]$ is a random number.

According to (2), compute $fit(v_i)$. If $fit(v_i) > fit(x_i)$, it implies that v_i is better than the old one x_i, then we set

$$x_i = v_i, \text{and } trail_i = 0;$$

else, set

$$trail_i = trail_i + 1,$$

where $trail_i$ is a counter, which records the number of x_i that has not been improved, and its initial value is 0.

Onlooker bees When the employed bees go back to the hive, the employed bees will share information with the onlooker bees. A selection mechanism based on roulette wheel is designed as follows:

$$p_i = \frac{fit_i}{\sum_{j=1}^{N} fit_j}, \quad (4)$$

where p_i denotes the probability that the ith food source is selected by an onlooker bee. From (4), we can see that the food source with good quality has higher probability of being selected. Thereafter, by using (3), the onlooker bee searches near the food source selected. Similarly, by comparing the quality of the new position and the old one, the food source and the counter $trail_i$ will be updated.

Scout bees After onlookers search, for food source x_i, if $trail_i > limit$, where $limit$ is a preset value, then, a new position x_i will be produced by using (1), and the trial counter $trail_i$ is set to 0. It should be pointed out that only one scout bee occur at each iteration.

From the above introduction, the framework of the basic ABC is described in Algorithm 1.

Algorithom 1. Basic ABC

01. Given the population size N, the maximum number of iteration $maxcycle$, and $limit$.
02. Generate the initial population $\{x_i | i = 1, \cdots, N\}$. Compute the function values of the population $\{f_i | i = 1, \cdots, N\}$, the fitness of the population $\{fit_i | i = 1, \cdots, N\}$, and determine the best solution x_{best}.
03. When *the stopping criterion is not met* do
04. For $i = 1$ to N do
05. Employed bees use (3) to obtain the new candidate v_i.
06. If $fit(v_i) > fit(x_i)$
07. set $x_i = v_i, fit_i = fit(v_i)$
08. End if
09. End for
10. Calculate p_i according to (4).
11. For $i = 1$ to N do
12. Select the food source by the probability p_i, and search the candidate near the food source for each onlooker bee according to (3).
13. Identify new food sources in a greedy way.
14. End for
15. If $trail_i > limit$
Generate a new candidate according to (1).
16. End if
17. Set $iter = iter + 1$.
18. End when

III. Improved artificial bee colony algorithm with experience (EABC)

According to basic ABC, we can see that only one dimension of x_i changes, which will undoubtedly affect the convergence rate of the algorithm. In addition, if a certain dimension of food source changes and the new position is improved, it is a good experience to guide individual evolution, but it is not well used in the basic ABC. Considering these two points, we present a new strategy to improve the basic ABC by using previous experience.

A. Improvement strategy based on experience

In the employed bee phase, for ith food source x_i, jth dimension is selected to generate a new candidate. Let v_i be the new candidate; $T(i, j)$ is used to record the times of jth dimension is selected, and $t(i, j)$ is used to record the times of the candidate solution becomes better after jth dimension is selected. Initially, both $T(i, j)$ and $t(i, j)$ are set to 0. After obtaining v_i, set

$$T(i, j) = T(i, j) + 1, \quad (5)$$

and if $fit(v_i) > fit(x_i)$, then it implies that the change of jth dimension is helpful for the improvement of the individual, set

$$t(i, j) = t(i, j) + 1. \quad (6)$$

Based on (5) and (6), we define a matrix M as follows:

$$M = (M_{ij})_{N \times D}, i = 1, \cdots, N,\ j = 1, \cdots, D, \quad (7)$$

where

$$M_{ij} = \frac{t(i, j)}{T(i, j)}.$$

To avoid the situation that (7) is meaningless when the denominator is zero, M_{ij} can be modified as

$$M_{ij} = \frac{t(i, j)}{T(i, j) + 0.01}, i = 1, \cdots, N,\ j = 1, \cdots, D. \quad (8)$$

The larger the M_{ij} value is, the more likely it is to choose the jth dimension to improve the quality of the solution. So, we have reason to believe that after selecting a food source x_i, we can select the corresponding dimensions to change according to the value M_{ij}.

According to the above analysis, once x_i is selected, we will generate a new position v_i according to the following formula:

$$v_{ij} = \begin{cases} x_{ij} + r_1 \times (x_{kj} - x_{ij}), & \text{if } rand < M_{ij}, \\ x_{ij}, & \text{else}, \end{cases} \tag{9}$$

where $r_1 \in [-1, 1]$ and $r_1 \in [0, 1]$ are two random numbers.

In order to make better use of the existing information, we modify (9) and add the direction guidance information as follows: if $fit(x_k) > fit(x_i)$, then v_i is generated by (10) below,

$$v_{ij} = \begin{cases} x_{ij} + r_2 \times (x_{kj} - x_{ij}), & \text{if } rand < M_{ij}, \\ x_{ij}, & \text{else}, \end{cases} \tag{10}$$

else v_i is generated by (9). Here, $r_2 \in [0, 1]$ is a random number.

From (9) and (10), it is not difficult to see that our method can not only make full use of historical information to guide individual evolution, but also overcome the problem of slow convergence speed caused by only one-dimensional change in basic ABC.

The pseudo-code of the proposed algorithm EABC is shown as follows:

Algorithom 2. EABC

01. Given the population size N, the maximum number of iteration $maxcycle$, and $limit$.
02. Generate the initial population $\{x_i | i = 1, \cdots, N\}$. Compute the function value of the population $\{f_i | i = 1, \cdots, N\}$, the fitness of the population $\{fit_i | i = 1, \cdots, N\}$, and determine the best solution x_{best}.
03. When *the stopping criterion is not met* do
04. For $i = 1$ to N do
05. Employed bees use (3) to obtain the new candidate v_i.
06. If $fit(v_i) > fit(x_i)$
07. set $x_i = v_i, fit_i = fit(v_i)$
08. End if
09. End for
10. Calculate p_i according to (4).
11. For $i = 1$ to N do
12. Select the food source by the probability p_i, and search the candidate near the food source for each onlooker bee according to (9) and (10).
13. Identify new food sources in a greedy way.
14. End for
15. If $trail_i > limit$
Generate a new candidate according to (1).
16. End if
17. Set $iter = iter + 1$.
18. End when

IV. Numerical experiments

To invest the performance of EABC, EABC and three ABC variants: ABC [9], COABC [29], and GABC [18] are tested on 12 benchmark functions. The experiments are executed on a computer with an Intel (R) Core (TM) i7-6500U CPU @ 2.50 GHz, 8 GB memory, Windows 10 system, and the experiments are written in Matlab 2017a.

A. Benchmark functions

Table I shows the 12 benchmark functions. In Table I, D, Range and $f(x^*)$ are used to represent dimensions, bounds of the search space and global minimum values of these functions, respectively. Among these benchmark functions, f_1-f_7 are continuous unimodal functions, f_8 is discontinuous step function, f_9-f_{12} are multi-modal functions.

For fair comparison, each algorithm runs independently 30 times on each problem, and population size is set to 100. The maximum number of iterations is set to 2000, which is also used as the termination condition. The other parameters are used as the comparison algorithms suggested. The comparison results are summarized in Table II.

B. Comparison results

From Table II, we can see that EABC is superior to three ABCs on most of 12 test benchmark functions, except for f_9 and f_{11}. For f_9 and f_{11}, the performance of EABC is slight worse than that of GABC. For f_8, all of these algorithms have the same minimum value. For functions f_3-f_7, the accuracy of EABC is much higher than those of the other algorithms, especially in function f_7.

To intuitively compare the convergence rate of EABC and the other three ABCs, the convergence curves (benchmark functions f_2-f_7) of these algorithms are displayed in Fig 1. From Fig 1, it is easy to see that EABC can find a better solution when the algorithm terminates. For functions f_5 and f_6, EABC converges to the optimal solution at a faster speed. For functions f_2-f_4 and f_7, although the convergence rate of EABC was not dominant in the early stage, its performance in the later stage was better than other methods with the accumulation of experience. This also implies that the experience guidance mechanism introduced is effective.

V. Conclusion

In this paper, to provide a more favorable direction, we introduce empirical knowledge into ABC algorithm, which help each bee choose dimensions to change. Numerical experiments show the mechanism is very useful. In the next step, this method will be used to solve some practical problems.

References

[1] N. Gabere, “Simulated annealing driven pattern search algorithms for global optimization”, Masters thesis, University of the Witwatersrand, Johannesburg, South Africa, 2007.
[2] R. Storn, K. Price, “Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces”, *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.
[3] M.W. Guo, D. Wei, J.S. Wang, X.X. Ma, “An improved flower pollination algorithm based on muti-population co-evolutionary strategy to solve function optimization problems,” Engineering Letters, vol. 28, no. 2, pp. 1182-1190, 2020.

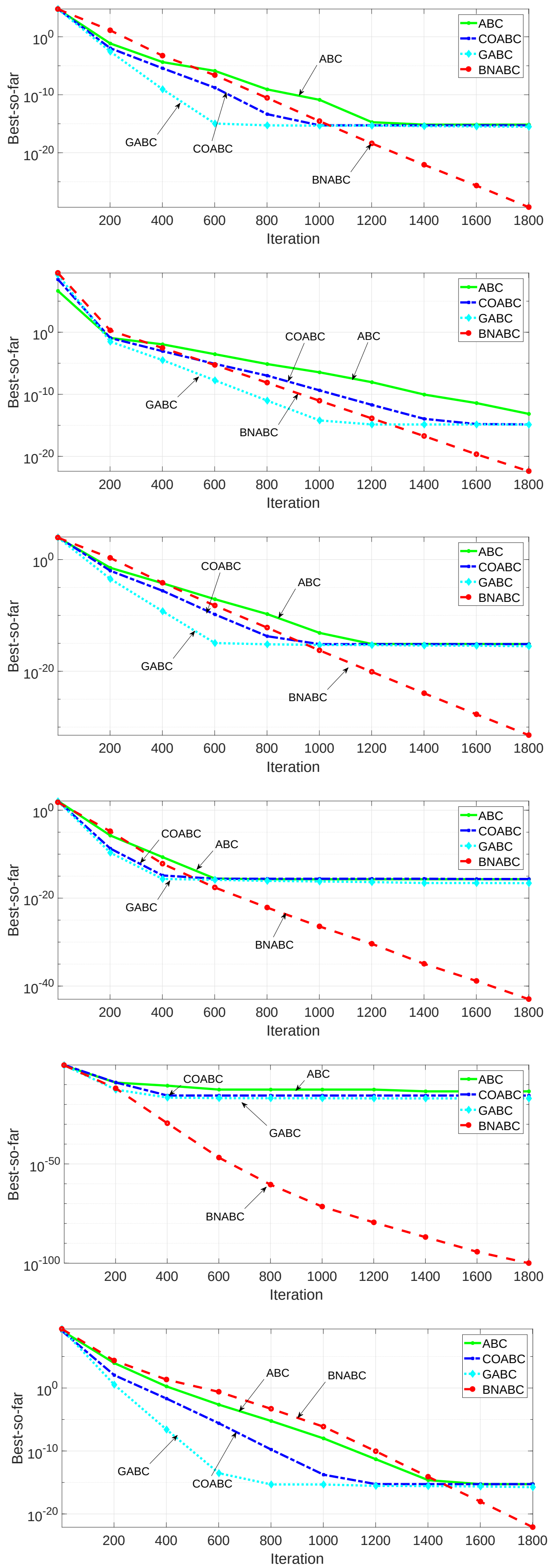

Fig.1 Convergence curves of f_2-f_7 with 30 dimensions

[4] K. Deep, M. Thakur, "A new mutation operator for real coded genetic algorithms", *Applied Mathematics and Computation*, vol. 193, no. 1, pp. 211-230, 2007.

[5] P. Kaelo, M.M. Ali, "Integrated crossover rules in real coded genetic algorithms", *European Journal of Operational Research*, vol. 176, no. 1, pp. 60-76, 2007.

[6] J. Kennedy, R. Eberhart, "Particle swarm optimization", in IEEE Int. Conf. Neural Networks, 1995.

[7] Q. Qu, M.L. Qi, et al., "An enhanced fireworks algorithm based on adaptive explosion amplitude and levy flight," Engineering Letters, vol. 28, no. 3, pp. 1348-1357, 2020.

[8] M. Dorigo, T. Stutzle, "Ant colony optimization", MIT Press, Cambridege, MA, 2004.

[9] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical Report-TR06, Erciyes University, Turkey, 2005.

[10] E. Amiri, M.N. Dehkordi, "Dynamic data clustering by combining improved discrete artificial bee colony algorithm with fuzzy logic", *International Journal of Bio-Inspired Computation*, vol. 12, no 2., pp. 164-172, 2018.

[11] L.B. Ma, X.W. Wang, et al., "Two-level master-slave RFID networks planning via hybrid multiobjective artificial bee colony optimizer", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 49, no. 5, pp. 861-880, 2019.

[12] L.B. Ma, X.W. Wang, et al., "A novel artificial bee colony optimiser with dynamic population size for multi-level threshold image segmentation", *International Journal of Bio-Inspired Computation*, vol. 13, no. 1, pp. 32-44, 2019.

[13] L.Z. Cui, G.H. Li, et al., "A ranking-based adaptive artificial bee colony algorithm for global numerical optimization", *Information Science*, vol. 417, pp. 169-185, 2017.

[14] P. Agarwal, S. Mehta. "ABC-DE-FP: a novel hybrid algorithm for complex continuous optimisation problems", *International Journal of Bio-Inspired Computation*, vol. 14, no. 1, pp. 46-61, 2019.

[15] B. Akay, D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization", *Information Science*, vol. 192, no. 1, pp. 120-142, 2012.

[16] D.L. Zhang, X.P. Guan, et al. "Modified artificial bee colony algorithms for numerical optimization", in: Proc. of 3rd International Workshop onIntelligent Systems and Applications, pp. 1-4, 2011.

[17] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization", *Expert Systems With Applications*, vol. 37, no. 8, pp. 5682-5687, 2010.

[18] G. Zhu, S. Kwong, "Gbest-guided artificial bee colony algorithm for numericalfunction optimization", *Applied Mathematics and Computatio*, vol. 217, no. 7, pp. 3166-3173, 2010.

[19] W.F. Gao, S.Y. Liu, "A modified artificial bee colony algorithm", *Computers and Operations Research*, vol. 39, no. 3, pp. 687-697, 2012.

[20] A. Barnharnsakun, T. Achalakul, B. Sirinaovakul, "The best-so-far selection inartificial bee colony algorithm", *Applied Soft Computing*, vol. 11, no. 2, pp. 2888-2901, 2011.

[21] D. Karaboga, B. Akay, "A modified artificial bee colony (ABC) algorithm forconstrained optimization problems", *Applied Soft Computing*, vol. 11, no. 3, pp. 3021-3031, 2011.

[22] M.S. Kiran, M. Guenduez, "A novel artificial bee colony-based algorithm for solvingthe numerical optimization problems", *International Journal of Innovative Computing Information and Control*, vol. 9, no. 1, pp. 6107-6121, 2012.

[23] Y. Liu, X. Ling, G. Liu, "Improved artificial bee colony algorithm with mutuallearning", *Journal of Systems Engineering and Electronics*, vol. 23, no. 2, pp. 265-275, 2012.

[24] W.F. Gao, S.Y. Liu, L.L. Huang, "A global best artificial bee colony algorithm for globaloptimization", *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2741-2753, 2012.

[25] W.F. Gao, S.Y. Liu, "A modified artificial bee colony algorithm", *Computers and Operations Research*, vol. 39, no. 3, pp. 687-697, 2012.

[26] W.F. Gao, S.Y. Liu, L.L. Huang, "A novel artificial bee colony algorithm based on modified search equation and orthogonal learning", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 43, no. 3, pp. 1011-1024, 2013.

[27] D. Bajer, B. Zoric, "An effective refined artificial bee colony algorithm for numerical optimisation", *Information Science*, vol. 504, pp. 221-275, 2019.

[28] H. Wang, W.J. Wang, "Artificial bee colony algorithm based on knowledge fusion", *Complex and Intelligent Systems*, vol. 7, pp. 1139-1152, 2021.

[29] J. Luo, Q. Wang, X.H. Xiao, "A modified artificial bee colony algorithm based on converge-onlooker approach for global optimization," *Applied Mathematics and Computation*, vol. 219, no. 20, pp. 10253-10262, 2013.

TABLE I: Benchmark test functions

Functions	Range	D	Optimal value
$f_1 = 4x_1^2 - 2.1x_1^4 + \frac{x_1^6}{3} + x_1x_2 - 4x_2^2 + 4x_2^4$	[-5,5]	2	-1.0316
$f_2 = \sum_{i=1}^{n} x_i^2$	[-100,100]	30	0
$f_3 = \sum_{i=1}^{D} \mid x_i \mid + \prod_{i=1}^{D} \mid x_i \mid$	[-10,10]	30	0
$f_4 = \sum_{i=1}^{D} ix_i^2$	[-10,10]	30	0
$f_5 = \sum_{i=1}^{D} ix_i^4$	[-1.28,1.28]	30	0
$f_6 = \sum_{i=1}^{D} \mid x_i \mid^{(i+1)}$	[-1,1]	30	0
$f_7 = \sum_{i=1}^{D} 10^{6\frac{i-1}{D-1}} x_i$	[-100,100]	30	0
$f_8 = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	[-1.28,1.28]	30	0
$f_9 = \sum_{i=1}^{D} ix_i^4 + random[0,1)$	[-1.28,1.28]	30	0
$f_{10} = \frac{1}{D} \sum_{i=1}^{D} (x_i^4 - 16x_i^2 + 5x_i)$	[-5,5]	30	-78.332
$f_{11} = \sum_{i=1}^{D} \mid x_i sin(x_i) + 0.1x_i \mid$	[-10,10]	30	0
$f_{12} = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	[-5.12,5.12]	30	0

TABLE II: The comparison results for different algorithms

Functions	ABC			$COABC$			$GABC$			$EABC$		
	Min	Mean	SD	Min	Mean	SD	Min	Mean	SD	Min	Mean	SD
f_1	-1.0316	-1.0315	8.88e-05	-1.0316	-1.0316	1.86e-05	-1.0316	-1.0316	4.24e-14	-1.0316	-1.0316	2.27e-16
f_2	1.36e-15	1.79e-15	2.68e-16	1.14e-15	1.45e-15	2.05e-16	8.43e-16	9.08e-16	6.25e-17	1.27e-20	4.12e-19	1.97e-20
f_3	2.63e-15	3.04e-15	2.48e-16	1.41e-15	1.60e-15	1.28e-16	7.07e-16	8.94e-16	1.179e-16	4.29e-26	1.24e-26	8.14e-26
f_4	4.71e-16	7.93e-16	1.96e-16	4.97e-16	5.15e-16	1.60e-17	1.29e-16	2.066e-16	7.63e-17	1.99e-37	9.83e-36	8.53e-36
f_5	1.94e-16	2.36e-16	4.53e-17	9.45e-17	1.75e-16	5.15e-17	1.69e-17	2.88e-17	1.50e-17	1.20e-50	2.73e-48	2.60e-48
f_6	4.31e-15	1.31e-14	7.82e-15	7.08e-17	1.77e-16	1.07e-16	7.66e-19	3.63e-18	4.39e-18	6.22e-108	6.24e-105	1.39e-105
f_7	5.28e-16	6.67e-16	8.42e-17	4.90e-16	6.06e-16	8.82e-17	2.01e-16	2.85e-16	9.29e-17	1.15e-27	3.21e-26	2.78e-26
f_8	0	0	0	0	0	0	0	0	0	0	0	0
f_9	1.97e-01	2.54e-01	4.24e-02	9.69e-02	1.56e-01	3.67e-02	4.87e-02	1.06e-01	1.82e-02	5.77e-02	1.00e-01	4.76e-02
f_{10}	-78.3	-78.3	1.42e-14	-78.3	-78.3	7.10e-15	-78.3	-78.3	2.75e-14	-78.3	-78.3	3.01e-14
f_{11}	6.19e-08	1.10e-07	4.18e-08	6.21e-08	1.81e-06	2.12e-06	7.09e-16	1.00e-15	3.59e-16	2.08e-12	6.51e-10	3.62e-10
f_{12}	0	1.06e-15	1.67e-15	0	1.77e-16	7.94e-06	0	0	0	0	0	0