An Empirical Balanced Artificial Bee Colony Algorithm

Zhen Wang and Xiangyu Kong

Abstract—**In the past decades, the artificial bee colony (ABC) algorithm has gained significant concern and has become an important metaheuristic algorithm. Despite the extensive research on various ABC variant algorithms, achieving a balance between exploratory and exploitative abilities, as well as escaping from local optimal positions, remains a major challenge. In this article, an empirical balanced artificial bee colony (EBABC) algorithm is proposed. Firstly, we utilize the information of randomly selected individual and the global optimal individual at present in a new searching equation to guide individuals towards favorable positions. Additionally, we introduce the diversity of the colony into another searching equation, allowing for the dynamic adjustment of the algorithm's exploratory and exploitative abilities. Furthermore, we design a mechanism for dynamically selecting equations to balance the exploratory and exploitative abilities of the algorithm. Lastly, we introduce a dynamic disturbance strategy for the optimal individual at present to prevent it from getting trapped in local optima. We conduct numerical experiments on 27 benchmark functions and 5 Non-negative Linear Least Squares Problems (NLLS). The results demonstrate the feasibility, effectiveness, and robustness of the proposed algorithm.**

Index Terms—**Artificial bee colony algorithm, optimization problems, empirical balance strategy, dynamic random search**

I. Introduction

ALONG with the development of world economy and advance in science and technology, a growing body of real world problems are complex optimization problems. To address these problems well, lots of researches have been done and numerous algorithms have been designed for different problems. All these algorithms can generally be classified into two main kinds: the traditional algorithm that the searching process relies on the gradient information of the problem, and the intelligent algorithm that the searching process imitates certain natural phenomena. As the problems getting complex, the traditional algorithm turns to be overstretched. More and more researchers shifted their sights to the intelligent algorithm due to its excellent optimal performance. Until now many intelligent algorithms have been designed for complex optimization problems, such as genetic algorithms (GAs)[1], particle swarm optimization (PSO)[2-5], ant colony optimization (ACO)[6], differential evolution (DE)[7], artificial bee colony algorithm (ABC)[8] and firefly algorithm (FA)[9,10] et al..

Among these algorithms, ABC algorithm excels other swarm intelligent algorithms due to its simple structure and high efficiency[11]. Since coming out, it has been attracted widespread attentions and applied in many areas. Although ABC was widely used and its performance had advantages over some state-of-art algorithm, there still had many modifications on the algorithm to adaptive the characteristics of different problems. In the original ABC, all the searching processes and searching equation shows the great exploratory ability of the algorithm which can efficiently prevent local convergence. However, the drawback is poor in exploitative ability, which also means the lower convergence speed. Therefore, most modifications of the ABC variants are focusing on the searching strategies. In [12], a Gbest-guided ABC algorithm is studied inspired by the searching equation of PSO, in which the current global optimal solution is considered into the searching equation to guide the searching direction. The experiment results indicate the GABC excels the original ABC in most test functions. In [13], a new solution generating method is designed by combining the current best solution and the fitness value of it, and used the adjustable search radius to enhance the population diversity. Gao et al. proposed several ABC variants, such as designed two different searching equations for artificial bees [14]; studied a MABC algorithm to balance the exploratory and exploitative abilities[15]; proposed a novel search equation inspired by GA[16]; combined three methods to enhance the performance, which are the orthogonal method, the opposition based learning technique and the chaotic strategy[17], and so on. Xue et al. given a SABC-GB by introducing the current global best solution in searching equations based on different strategies[18]. Cui et al. combined the depth first search method and a modified searching equation to obtain the better results[19]. Karaboga et al. proposed the qABC algorithm by using the best individual in searching equation at onlooker bee phase[20]. Wang et al. employed three kinds of approaches in the searching stages, the best solution set is designed and used for searching[21]. Wang et al. designed an improved ABC by experience guiding, in which the experience of individual improvement is collected for improving quality of the individuals[22].

Although, these ABC variant algorithms produce many successful results, there still exists improvement room to boost the optimal performance of the algorithm. Balancing the exploratory and exploitative abilities is the most important purpose for the algorithm modification. Moreover,

Manuscript received July 3, 2023; revised November 30, 2023.

This work was supported by the National Social Science Foundation of China under Grant 20BTJ026.

Zhen Wang is a professor of the Institute of Mathematics and Statistics, Xianyang Normal University, Shaanxi, 712000, China. (corresponding author. e-mail: jen715@163.com).

Xiangyu Kong is an associate professor of the Institute of Mathematics and Statistics, Xianyang Normal University, Shaanxi, 712000, China. (e-mail: kxywz08@163.com).

easily trapped into local optimization at the latter stages in the searching process is also a crucial task for algorithm modification. To conquer these issues, the well designed strategies should be considered.

In this paper, we would like to propose an empirical balanced artificial bee colony (EBABC) algorithm to overcome the issues mentioned above. The main contributions of this paper are as below. (a) A new searching equation is presented, in which the information of randomly selected individual and current global optimal individual are used. (b) Based on the diversity, another searching equation is designed, in which the exploratory and exploitative abilities are adaptive balanced. (c) A dynamically equation selection mechanism is given, which can help choose searching equation better in the artificial bees searching process. (d) A dynamic disturbance strategy is added to avoid the algorithm trapped into the local optimization earlier.

The rest of the paper is organized as follows. In Section 2, the ABC is illustrated in detail . In Section 3, the proposed EBABC is given. In Section 4, numerical experiments are given, and the results are discussed. The conclusion is presented in Section V.

II. Overview of basic artificial bee colony algorithm

On the basis of the real bees foraging behavior, Karaboga[8] designed a new metaheuristic algorithm named artificial bee colony algorithm (ABC) for unconstrained optimizations, in which the artificial bees simulated the searching behavior of food sources of real bee colony. In ABC, the artificial bees in colony can be separated into three kinds, i.e. employed bees, onlooker bees and scout bees. Each kind of artificial bees correspond to a stage of the algorithm. In the initialization, all artificial bees are randomly set in the domain based on the following equation:

$$x_i = x^{\min} + rand \cdot \left(x^{\max} - x^{\min}\right),\ i = 1,...,n, \quad (1)$$

where x_i is the i th feasible solution in the algorithm, $x^{\max}$ and $x^{\min}$ represent the upper bound and lower bound of the domain. For the employed bees, each bee explores the domain to find the food source and exploits it by

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}),\ i = 1,...,n, \quad (2)$$

where $j \in \{1,2,...,D\}$ is randomly selected, $k \in \{1,2,...,n\}$ and $k \neq i$. Then the employed bees send the food source information to the onlooker bees by waggle dance, which in the algorithm can be formulated as the roulette wheel selection of the food source to exploit. The selection probability can be defined as:

$$p_i = \frac{fit_i}{\sum_{i=1}^{n} fit_i},\ i = 1,...,n, \quad (3)$$

where fit_i represents the fitness value of the i th feasible solution calculated below for the minimum optimization problems:

$$fit_i = \begin{cases} \frac{1}{1+f_i}, & f_i \geq 0, \\ 1 + abs(f_i), & f_i < 0, \end{cases} \quad i = 1,...,n, \quad (3)$$

where f_i is the objective function value of the the i th food source. After the selection of the food source, the onlooker bees will exploit the food source by equation (2). If a solution cannot be updated enough times, it will become a abandoned food source. And the scout bee will generate a new one randomly by equation (1).

III. Empirical balanced artificial bee colony algorithm

In ABC, the search equation of individuals tends to boost the exploratory ability of searching process, because of the randomly selected individual in the equation. This feature ensures the searching process will not fall into local optimal easily, but it will affect the exploitative ability of the algorithm, consequently lower the convergent speed. So as to boost the exploitative ability, the current global best individual (x_{best}) has been employed into the searching equation of the ABC algorithm in [12]. The information of x_{best} will help to guide the individual towards to the better individual, and the information of the randomly selected individual (x_k) will keep the diversity of colony. However, the influence degree of x_{best} and x_k on the new individual is not distinguished in Zhu's searching equation. If the influence degree of x_{best} and x_k are considered, it will be more good for the individual's searching. Furthermore, as we all know that adaptive adjusting the convergence and diversity along with the searching process is an crucial aspect of the metaheuristic algorithm. Therefore, in this paper the dynamically adjusting of exploratory ability and exploitative ability is designed and a new ABC variant (EBABC) is proposed. The details of the proposed algorithm are below.

A. Best individual learning strategy

From the above analysis, if the influence degree of x_{best} and x_k is considered in the search process, it will be more directional for the individual searching. To achieve this goal, we will define tow metrics to represent the influence degrees, that is:

$$\omega_1 = \frac{fit_k}{fit_{gbest} + fit_k},$$

and

$$\omega_2 = \frac{fit_{gbest}}{fit_{gbest} + fit_k},$$

where fit_k and fit_{gbest} are the fitness value of x_k and x_{best} calculated by equation (3) respectively. Therefore, an artificial bee can search the domain by

$$v_i = x_i + \phi_i(\omega_1 \times x_k - x_i) + \varphi_i(\omega_2 \times x_{best} - x_i), \quad (4)$$

where $\phi_i \in [-1,1]$, $\varphi_i \in [0,C]$, $k \in \{1,2,...,n\}$ and $k \neq i$.

Apparently, the influence degree of x_k and x_{best} are considered into the individual searching equation by introducing ω_1 and ω_2.

B. Empirical balanced strategy

For the adaptive adjusting the exploratory ability and exploitative ability of EBABC, we introduce the diversity

TABLE I
BENCHMARK FUNCTIONS.

Functions	Name	D	C	Range
$f_1 = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	Beale	2	UN	[-4.5,4.5]
$f_2 = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	Bohachevsky	2	MS	[-100,100]
$f_3 = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	Booth	2	MS	[-10,10]
$f_4 = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	Branin	2	MS	[-5,10]× [0,15]
$f_5 = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2)$	Colville	4	UN	[-10,10]
$f_6 = -\cos x_1 \cos x_2 \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	Easom	2	UN	[-100,100]
$f_7 = \begin{bmatrix} 1 + (x_1 + x_2 + 1)^2 \\ (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \end{bmatrix} \cdot \begin{bmatrix} 30 + (2x_1 - 3x_2)^2 \\ (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \end{bmatrix}$	GoldStein-Price	2	MN	[-2,2]
$f_8 = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right]$, $c = [1.0, 1.2, 3.0, 3.2]$, $a = \begin{bmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}^T$, $p = \begin{bmatrix} 0.3689 & 0.1170 & 0.2673 \\ 0.4699 & 0.4387 & 0.7470 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{bmatrix}^T$	Hartman3	3	MN	[0,1]
$f_9 = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	Six Hump Camel Back	2	MN	[-5,5]
$f_{10} = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	Matyas	2	UN	[-10,10]
$f_{11} = \sum_{i=1}^{n/k} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} + 10x_{4i})^2 + (x_{4i-2} + 10x_{4i-1})^4 + 10(x_{4i-3} + 10x_{4i})^4$	Powell	4	UN	[-4,5]
$f_{12} = \sum_{k=1}^{n}\left[\sum_{i=1}^{4}(x_i^k) - b_k\right]^2$; $b = [8, 18, 44, 114]$	PowerSum	24	MN	[0,*D*]
$f_{13} = -\sum_{j=1}^{m}\left[\sum_{i=1}^{4}(x_i - a_{ij})^2 + c_i\right]^{-1}$, $a = \begin{bmatrix} 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 5.0 & 1.0 & 2.0 & 3.6 \\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 3.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \end{bmatrix}$, $c = \frac{1}{10}[1, 2, 2, 4, 4, 6, 3, 7, 5, 5]^T$	Shekel	4	MN	[0,10]
$f_{14} = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right) \cdot \left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$	Shubert	2	MN	[-10,10]
$f_{15} = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$	Trid6	6	UN	[-36,36]

D = Dimension; C = Characteristic; U = Unimodal; M = Multimodal; S = Separable; N = Non-Separable.

degree into the searching equation. Firstly, we define the diversity degree as

$$\rho = \frac{1}{n \times \|u - l\|} \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{D} (x_{ij} - \bar{x}_j)^2} ,$$

where n is the colony size, D is the problem's dimension, l and u are the lower bound and upper bound of the search domain respectively. $\bar{x}$ denote the mean value of all individuals. Thus, the searching equation with diversity information can be formulated as

$$v_i = x_i + \phi_i(\omega \times x_k - x_i) + \varphi_i((1-\omega) \times x_{best} - x_i) , \quad (5)$$

where $\omega = \exp(-\rho)$.

TABLE II
BENCHMARK FUNCTIONS.

Functions	Name	C	Range
$f_{16}=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	Ackley	MN	[-32,32]
$f_{17}=(x_1-1)^2+\sum_{i=2}^{n}i(2x_i^2-x_{i-1})^2$	Dixon-Price	UN	[-10,10]
$f_{18}=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	Griewank	MN	[-600,600]
$f_{19}=\sin^2(\pi y_1)+\sum_{i=1}^{n-1}\left[(y_i-1)^2(1+10\sin^2(\pi y_i+1))\right]+(y_n-1)^2(1+10\sin^2(2\pi y_n))$, $y_i=1+\frac{x_i-1}{4}, \quad i=1,\cdots,n$.	Levy	MN	[-10,10]
$f_{20}=-\sum_{i=1}^{n}\sin(x_i)\left(\sin(ix_i^2/\pi)\right)^{2m}, m=10$	Michalewicz	MS	$[0,\pi]$
$f_{21}=\sum_{k=1}^{n}\left[\sum_{i=1}^{2}(i^k+0.5)((x_i/i)^k-1)\right]^2$	Perm	MN	[- D , D]
$f_{22}=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	Rastrigin	MS	[-5.12,5.12]
$f_{23}=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	Rosenbrock	UN	[-30,30]
$f_{24}=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	Schwefel	MS	[-500,500]
$f_{25}=\sum_{i=1}^{n}x_i^2$	Sphere	US	[-100,100]
$f_{26}=\sum_{i=1}^{n}ix_i^2$	SumSquares	US	[-10,10]
$f_{27}=\sum_{i=1}^{n}x_i^2+\left(\sum_{i=1}^{n}0.5ix_i\right)^2+\left(\sum_{i=1}^{n}0.5ix_i\right)^4$	Zakharov	UN	[-5,10]

D = Dimension; C = Characteristic; U = Unimodal; M = Multimodal; S = Separable; N = Non-Separable.

From the above equation (5), it shows that the lager ρ is, the smaller ω is, which means the x_{best} has a bigger impact on the searching process boosting the exploitative ability of the algorithm.

Along with the changes of the colony's diversity, the whole searching process will adaptive balance the exploratory and exploitative by equation (5).

C. New search mechanism

In the above sections, tow searching equations are designed for different purposes. By equation (4) the main purpose is to distinct the impact of x_k and x_{best} on the new individual v_i ; while by equation (5) the main purpose is to adaptive adjust the exploratory ability and the exploitative ability of the algorithm . Generally, it is hoped that the information of current global optimal individual can be learned to accelerate the searching speed at the early stage of the searching process, and the information of the colony diversity can be considered to avert the algorithm trapped into the local optimal individual at the latter stage of the searching process. Therefore, the new searching mechanism is designed by dynamically selecting equation (4) and equation (5). Specifically, we introduce a selection index, i.e. $\theta=\exp(-iter/Maxiter)$, where $iter$ is the current iteration and $Maxiter$ is the total iteration. If a random number on $[0,1]$ is less than θ, the equation (4) is used for the artificial bees searching, otherwise the equation (5) is used.

D. Best-so-far individual improvement

To avoid the algorithm stuck in local optimum too early, the disturbance of optimal individual is introduced, which is

$$x_{best}^{'}=(1-\xi)\times x_{best}+\xi\times TS\ , \tag{6}$$

where $TS=l+rand\times(u-l)$ and $\xi=\frac{Maxiter-iter+1}{Maxiter}$.

From equation (6), as the iteration increasing, ξ decrease, and the difference between $x_{best}^{'}$ and x_{best} is getting small.

According to the above discussion, the procedure of EBABC is given in Algorithm 1.

Algorithm 1. Procedure of EBABC

Step 1: Initialization. Set all the parameters. And initialize the colony by equation (1).

Step 2: Employed bee stage. If a random number on $[0,1]$ is less than θ , the equation (4) is used for the artificial bees searching, otherwise the equation (5) is used. Update the individual by greedy selection.

Step 3: Onlooker bee stage. According to equation (3), calculate the probability and select a solution for onlooker bee. If a random number on $[0,1]$ is less than θ , the equation (4) is used for the artificial bees searching, otherwise the equation (5) is used. Update the individual by greedy selection.

Step 4: Scout bee stage. If there is an abandoned food source, the scout bee will generate a new one by equation (1) to replace it.

Step 5: Global best individual update. Generate a new individual by equation (6), and update the global best individual by greedy selection.

Step 6: If termination criteria is satisfied, stop the procedure and output the results; otherwise, switch to Step 2.

E. Computational time complexity

The time complexity of the original ABC algorithm is

TABLE III
EXPERIMENT RESULTS OBTAINED BY EBABC AND OTHER FOUR ABC VARIANT ALGORITHMS.

Function		ABC	GABC	EABC	MeanABC	EBABC
f_1	min	1.45E-10	1.22E-08	1.40E-12	4.22E-06	1.84E-06
	mean	1.53E-07	1.40E-05	4.21E-03	1.01E-03	1.53E-04
	std.	4.50E-07	2.30E-05	6.31E-03	1.09E-03	1.38E-04
	t-test	-	-	+	+	
	rank	1	2	5	4	3
f_2	min	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	0.00E+00	0.00E+00	3.64E-07	0.00E+00	0.00E+00
	std.	0.00E+00	0.00E+00	1.41E-06	0.00E+00	0.00E+00
	t-test	=	=	+	=	
	rank	1	1	5	1	1
f_3	min	3.87E-19	2.44E-19	7.69E-20	2.30E-06	3.52E-06
	mean	1.55E-17	1.36E-16	1.34E-05	3.43E-04	7.00E-05
	std.	1.67E-17	6.59E-16	4.58E-05	2.92E-04	7.31E-05
	t-test	-	-	-	+	
	rank	1	2	3	5	4
f_4	min	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	mean	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	std.	0.00E+00	0.00E+00	4.80E-09	1.04E-04	1.75E-05
	t-test	=	=	=	=	
	rank	1	1	1	1	1
f_5	min	-6.13E+06	-1.09E+10	-1.85E+156	-3.75E+09	-4.72E+258
	mean	-1.52E+06	-7.40E+08	-6.19E+154	-3.77E+08	-1.57E+257
	std.	1.47E+06	2.10E+09	#NUM!	9.46E+08	#NUM!
	t-test	+	+	+	+	
	rank	5	3	2	4	1
f_6	min	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00
	mean	-9.99E-01	-9.98E-01	-1.00E+00	-9.88E-01	-1.00E+00
	std.	9.46E-04	4.80E-03	1.73E-04	1.36E-02	3.84E-04
	t-test	+	+	=	+	
	rank	3	4	1	5	1
f_7	min	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	mean	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	std.	1.09E-03	2.53E-15	1.71E-12	9.18E-04	8.69E-06
	t-test	=	=	=	=	
	rank	1	1	1	1	1
f_8	min	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
	mean	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
	std.	2.27E-15	2.36E-15	1.07E-09	2.27E-04	4.65E-05
	t-test	=	=	=	=	
	rank	1	1	1	1	1

$O(max\,Gen \times SN \times D)$, where D is the problem dimension, $max\,Gen$ is the maximum number of iterations, and SN is the number of food sources. Compared with the original ABC algorithm, the time complexity of initialization for EABC is $O(SN \times D)$. At the employed bee stage with the new search mechanism, the time complexity is $O(max\,Gen \times SN \times D)$, and so is the time complexity of onlooker bee stage. Moreover, the time complexity of global best individual update is $O(1)$. Therefore, the total time complexity can be calculated as $O(SN \times D + max\,Gen \times SN \times D + max\,Gen \times SN \times D + 1)$. To sum up, the total computational time complexity of EABC is $O(max\,Gen \times SN \times D)$, same as the original ABC algorithm.

TABLE IV
EXPERIMENT RESULTS OBTAINED BY EBABC AND OTHER FOUR ABC VARIANT ALGORITHMS.

Function		ABC	GABC	EABC	MeanABC	EBABC
f_9	min	4.65E-08	4.65E-08	4.65E-08	5.19E-08	7.76E-08
	mean	4.65E-08	4.65E-08	4.66E-08	1.59E-06	6.12E-07
	std.	0.00E+00	9.55E-17	3.66E-10	1.60E-06	6.94E-07
	t-test	-	-	-	+	
	rank	1	1	1	5	4
f_{10}	min	5.94E-17	2.92E-07	5.52E-18	5.20E-09	4.47E-10
	mean	1.00E-09	1.48E-05	1.11E-05	4.69E-06	3.21E-08
	std.	3.76E-09	1.46E-05	5.31E-05	8.85E-06	5.52E-08
	t-test	-	+	+	+	
	rank	1	4	3	5	2
f_{11}	min	1.90E-07	1.40E-07	1.48E-10	2.19E-07	4.76E-09
	mean	2.93E-05	2.68E-05	3.52E-04	7.01E-06	2.52E-07
	std.	1.89E-05	4.19E-05	7.28E-04	7.36E-06	3.76E-07
	t-test	+	+	+	+	
	rank	4	3	5	2	1
f_{12}	min	5.17E-04	2.44E-04	4.15E-04	5.92E-03	1.45E-03
	mean	1.29E-02	1.27E-02	1.13E-02	3.50E-02	1.57E-02
	std.	1.28E-02	9.74E-03	1.11E-02	2.58E-02	1.60E-02
	t-test	-	-	-	+	
	rank	3	2	1	5	4
f_{13}	min	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01
	mean	-1.05E+01	-1.05E+01	-1.05E+01	-1.02E+01	-1.05E+01
	std.	3.18E-05	1.65E-15	5.42E-04	1.57E-01	5.21E-04
	t-test	=	=	=	+	
	rank	1	1	1	5	1
f_{14}	min	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02
	mean	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02	-1.87E+02
	std.	3.84E-14	3.54E-14	1.79E-11	1.25E-02	3.38E-03
	t-test	=	=	=	=	
	rank	1	1	1	1	1
f_{15}	min	-5.00E+01	-5.00E+01	-5.00E+01	-5.00E+01	-5.00E+01
	mean	-5.00E+01	-5.00E+01	-4.99E+01	-4.96E+01	-5.00E+01
	std.	1.79E-10	6.72E-10	2.73E-01	3.21E-01	5.78E-02
	t-test	=	=	+	+	
	rank	1	1	3	4	1

IV. NUMERICAL EXPERIMENTS

So as to test the performance of EBABC algorithm, the experiments are executed on 27 benchmark functions and 5 Non-negative Linear Least Squares Problems (NLLS). The performance of EBABC is contrasted with other four ABC variant algorithms, which are original ABC[8], GABC[12], EABC[17] and MeanABC[23].

In the experiments, the parameters setting are: colony size n is 40, the maximum number of iteration is 2000, so the FEs is 80000. All experiments are run 30 times independently.

A. Benchmark functions

The 27 benchmark functions, which have different characteristics, are selected to verify the capability of EBABC in different environments. The details of test functions are provided in Table I and II. In Table I, all test functions are given dimensions of their own. And the dimensions of test functions in Table II are all set $D = 30$.

B. Experiment results

The experiment results are given in Tables III-V. In these tables, min, mean and std. represent the best, mean and standard derivation of the function value over 30 times repeats. The results of Wilcoxons Signed-Ranked-Test (WSRT) over the function values of EBABC and other four

TABLE V
EXPERIMENT RESULTS OBTAINED BY EBABC AND OTHER FOUR ABC VARIANT ALGORITHMS.

Function		ABC	GABC	EABC	MeanABC	EBABC
f_{16}	min	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	mean	9.89E-15	1.01E-15	2.93E-08	8.88E-16	8.88E-16
	std.	1.40E-14	6.49E-16	1.50E-07	0.00E+00	0.00E+00
	t-test	+	+	+	=	
	rank	4	3	5	1	1
f_{17}	min	4.96E-05	1.04E-03	4.77E-05	3.08E-01	1.00E-02
	mean	7.58E-04	4.48E-01	4.59E-01	6.54E-01	1.49E-01
	std.	1.14E-03	3.31E-01	1.02E+00	6.56E-02	1.23E-01
	t-test	-	+	+	+	
	rank	1	3	4	5	2
f_{18}	min	0.00E+00	3.33E-16	0.00E+00	0.00E+00	0.00E+00
	mean	2.22E-03	1.14E-03	1.08E-03	4.44E-17	4.77E-12
	std.	5.77E-03	3.55E-03	4.36E-03	5.53E-17	2.61E-11
	t-test	+	+	+	-	
	rank	5	4	3	1	2
f_{19}	min	3.10E-16	4.57E-16	1.60E-16	3.39E-02	6.01E-08
	mean	5.49E-16	5.97E-16	2.83E-16	7.09E-02	2.08E-04
	std.	9.47E-17	1.09E-16	5.47E-17	2.33E-02	3.10E-04
	t-test	-	-	-	+	
	rank	2	3	1	5	4
f_{20}	min	-2.95E+01	-2.95E+01	-2.96E+01	-2.57E+01	-2.71E+01
	mean	-2.94E+01	-2.94E+01	-2.94E+01	-2.41E+01	-2.65E+01
	std.	4.90E-02	4.87E-02	1.54E-01	6.45E-01	3.77E-01
	t-test	-	-	-	+	
	rank	1	1	1	5	4
f_{21}	min	9.43E+78	3.69E+79	4.24E+76	7.04E+77	7.62E+77
	mean	1.77E+83	1.37E+83	1.80E+81	2.09E+82	8.28E+81
	std.	3.59E+83	6.87E+83	4.06E+81	4.62E+82	1.75E+82
	t-test	+	+	-	+	
	rank	5	4	1	3	2
f_{22}	min	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	mean	4.74E-14	7.96E-14	1.66E-01	0.00E+00	0.00E+00
	std.	5.80E-14	1.06E-13	4.59E-01	0.00E+00	0.00E+00
	t-test	+	+	+	=	
	rank	3	4	5	1	1
f_{23}	min	5.37E-04	1.28E-03	1.14E-02	1.97E+01	5.03E-05
	mean	3.44E-01	7.79E+00	4.04E+00	2.51E+01	2.03E-01
	std.	7.73E-01	1.83E+01	1.33E+01	1.65E+00	7.78E-01
	t-test	-	+	+	+	
	rank	1	4	3	5	2
f_{24}	min	3.82E-04	3.82E-04	3.82E-04	1.58E+02	3.82E-04
	mean	1.97E+01	5.11E+01	2.49E+01	2.46E+02	9.54E-03
	std.	4.49E+01	8.68E+01	8.04E+01	5.74E+01	1.83E-02
	t-test	+	+	+	+	
	rank	2	4	3	5	1
f_{25}	min	3.15E-16	5.14E-16	1.39E-16	3.15E-16	2.61E-16
	mean	5.84E-16	6.69E-16	2.80E-16	4.37E-16	3.97E-16
	std.	1.08E-16	9.03E-17	9.96E-17	7.60E-17	8.04E-17
	t-test	+	+	-	+	
	rank	4	5	1	3	2
f_{26}	min	4.02E-16	2.22E-16	1.61E-16	2.96E-16	2.82E-16
	mean	5.71E-16	5.75E-16	1.41E-11	4.35E-16	3.64E-16
	std.	9.25E-17	1.33E-16	7.75E-11	5.89E-17	6.73E-17
	t-test	+	+	+	+	
	rank	3	4	5	2	1
f_{27}	min	1.66E+02	2.75E+02	1.21E+02	1.73E+02	1.53E-03
	mean	2.46E+02	3.31E+02	2.52E+02	2.57E+02	2.62E+00
	std.	3.60E+01	3.32E+01	5.51E+01	3.34E+01	5.12E+00
	t-test	+	+	+	+	
	rank	5	4	2	3	1
Final Rank		62	71	68	88	50
Total Rank		2	4	3	5	1

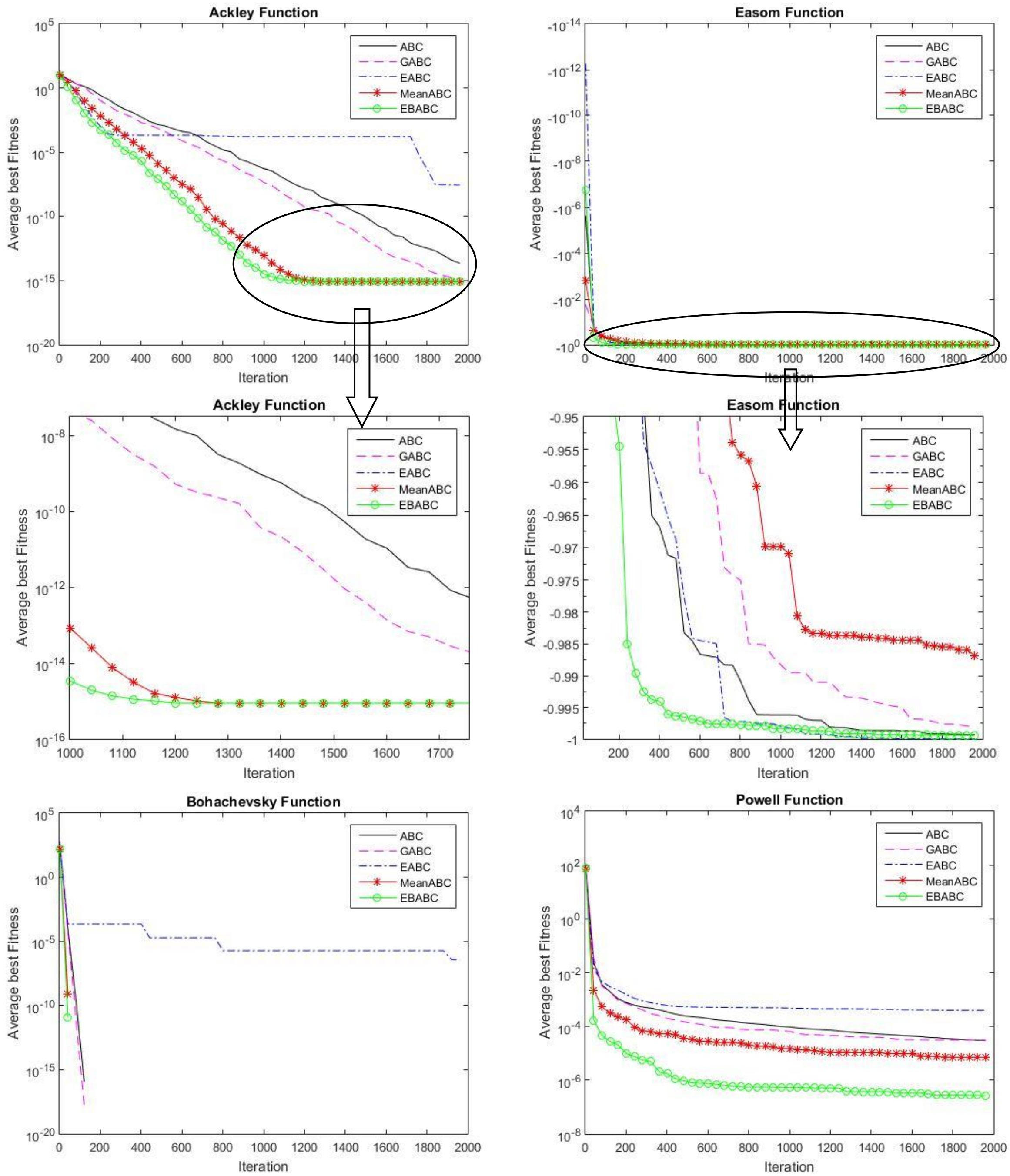

Fig. 1. Some convergence curves of EBABC and other four ABC variant algorithms.

ABC variant algorithms are also given in Tables III-V. The WSRT is executed at 0.05 confidence level, which can further compare the optimal performance between EBABC and other four ABC variant algorithms. The marks ‘+’, ‘−’ and ‘=’ represent the EBABC statistically better than, worse than and not different from the other compared algorithm, respectively.

From Table III-V, we can easily know that EBABC has the better or at least the same performance on almost all test functions compared with the other four algorithms. Moreover, in Table V the total rank of ABC, GABC, EABC, MeanABC and EBABC are 2, 4, 3, 5 and 1 respectively. According to the last two rows in Table V, the total rank and the final rank both indicate that EBABC has the best optimal capability over the other listed variant algorithms.

Furthermore, some convergence curves are given in Fig. 1 and Fig. 2, and the boxplots are drawn in Fig. 3 and Fig. 4 to illustrate the performance of EBABC more visually. From Fig. 1 and Fig. 2, we can see that for most functions EBABC can find a better solution in fewer iterations, especially for f_2 and f_{22}. From Fig. 3 and Fig. 4, the box-plots show that the stability of EBABC is far better than other four variant algorithms.

C. Experiments on Non-negative Linear Least Squares Problems

With rapid progress of science and technology, Non-negative Linear Least Squares Problem (NLLS) has been widely used in forecasting, systems engineering, economic, biological engineering and other fields.

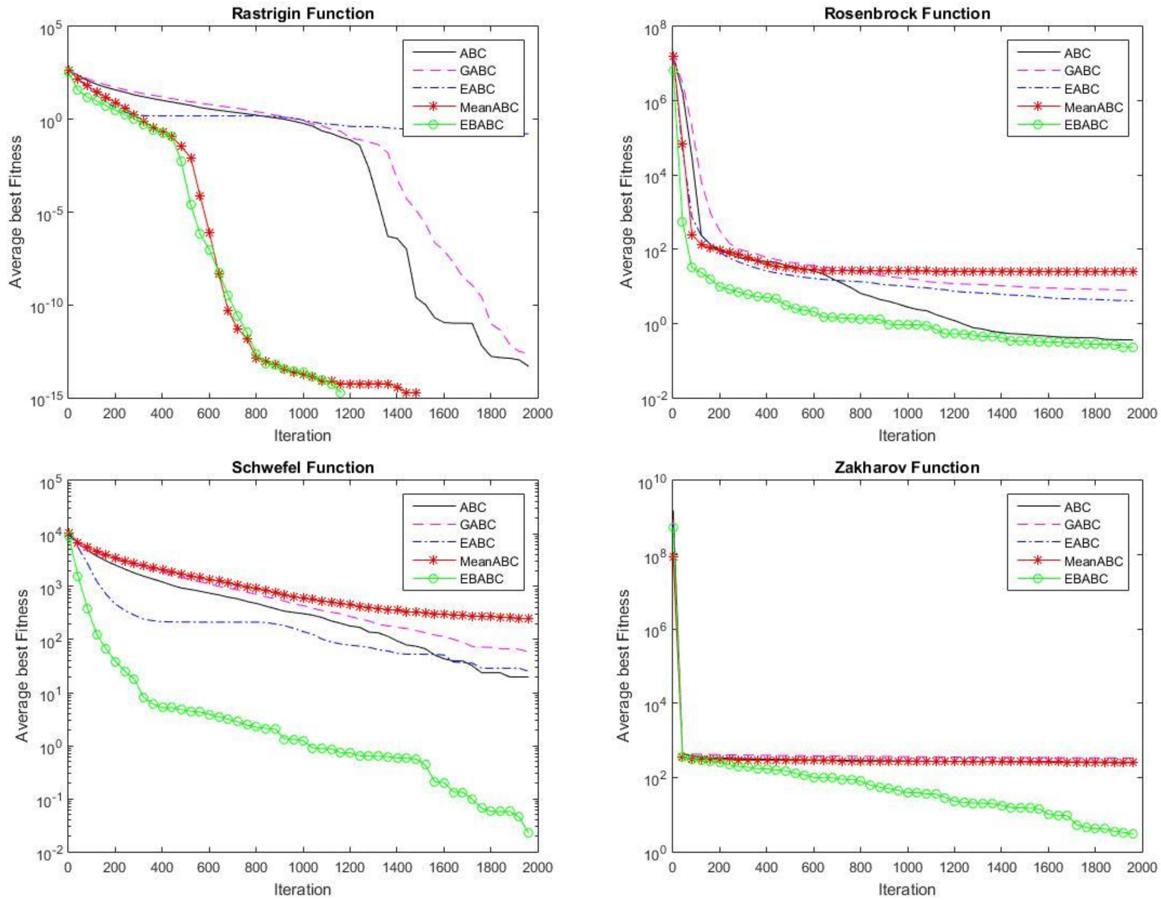

Fig. 2. Some convergence curves of EBABC and other four ABC variant algorithms.

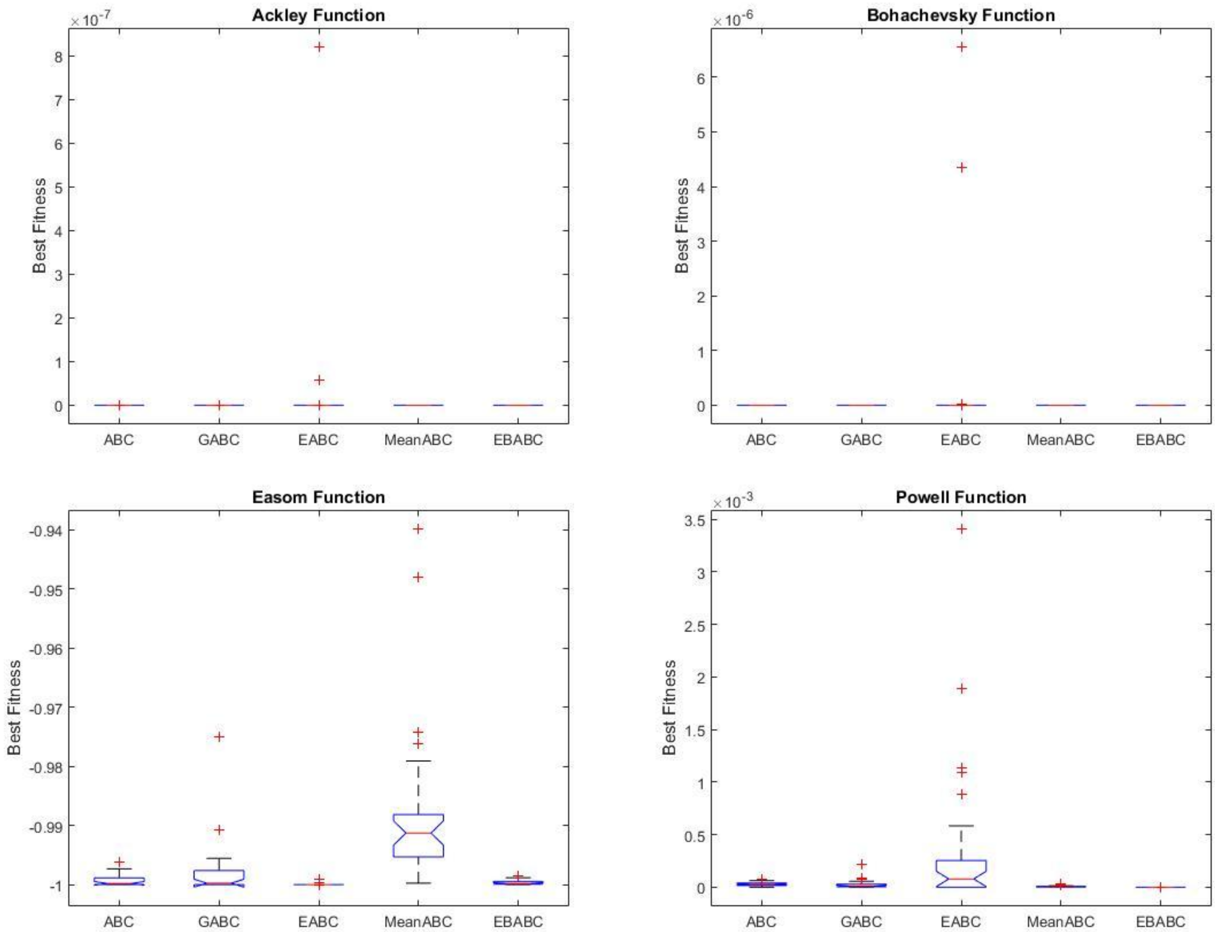

Fig. 3: Some boxplots of EBABC and other four ABC variant algorithms.

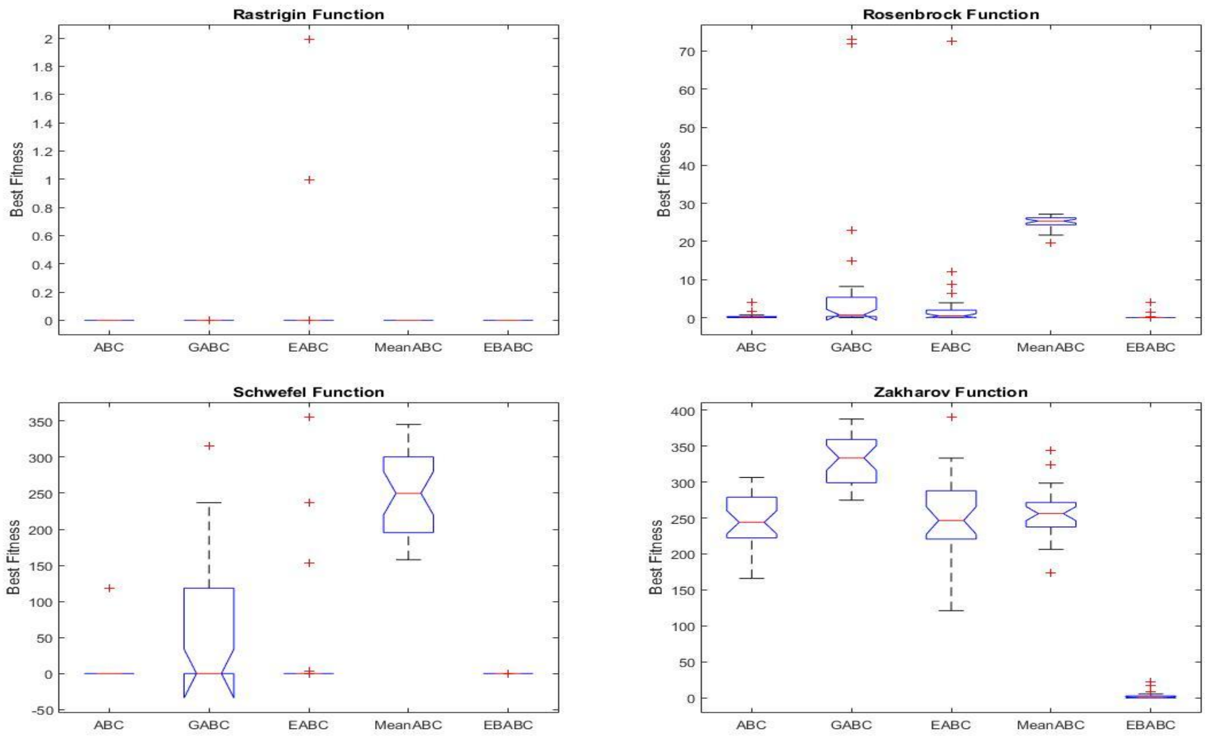

Fig. 4: Some boxplots of EBABC and other four ABC variant algorithms.

TABLE VI
EXPERIMENT RESULTS OBTAINED BY EBABC AND OTHER FOUR ABC VARIANT ALGORITHMS FOR NLLS PROBLEMS.

Problems		ABC	GABC	EABC	MeanABC	EBABC
NLLS1	min	4.90E-17	6.60E-19	2.69E-16	2.78E-04	1.33E-05
	mean	3.42E-16	6.07E-16	3.44E-04	4.83E-03	4.76E-04
	std.	2.41E-16	2.13E-15	8.61E-04	4.41E-03	6.67E-04
	t-test	-	-	-	+	
	rank	2	1	3	5	4
NLLS2	min	1.14E+01	1.18E+02	6.13E-03	7.65E+05	1.81E-01
	mean	2.15E+01	9.89E+02	7.87E-01	1.26E+06	3.13E+02
	std.	7.21E+00	8.73E+02	3.81E+00	2.28E+05	4.25E+02
	t-test	+	+	-	+	
	rank	3	4	1	5	2
NLLS3	min	4.41E+02	6.27E+04	1.43E+03	1.05E+05	1.05E-03
	mean	3.71E+03	1.45E+05	5.35E+03	4.28E+05	2.31E+01
	std.	1.17E+04	4.11E+04	4.79E+03	1.34E+05	5.60E+01
	t-test	+	+	+	+	
	rank	2	4	3	5	1
NLLS4	min	1.56E+01	3.30E+02	2.73E+01	2.22E+02	7.56E-06
	mean	2.88E+01	4.72E+02	6.55E+01	5.70E+02	1.35E+00
	std.	8.95E+00	6.56E+01	2.89E+01	1.69E+02	2.93E+00
	t-test	+	+	+	+	
	rank	2	5	3	4	1
NLLS5	min	7.89E+03	7.85E+05	8.81E+04	5.63E+05	5.64E-01
	mean	4.56E+04	9.69E+05	1.82E+05	1.09E+06	6.40E+02
	std.	7.73E+04	1.19E+05	5.92E+04	1.90E+05	1.35E+03
	t-test	+	+	+	+	
	rank	2	5	3	4	1
Final Rank		11	19	13	23	9
Total Rank		2	4	3	5	1

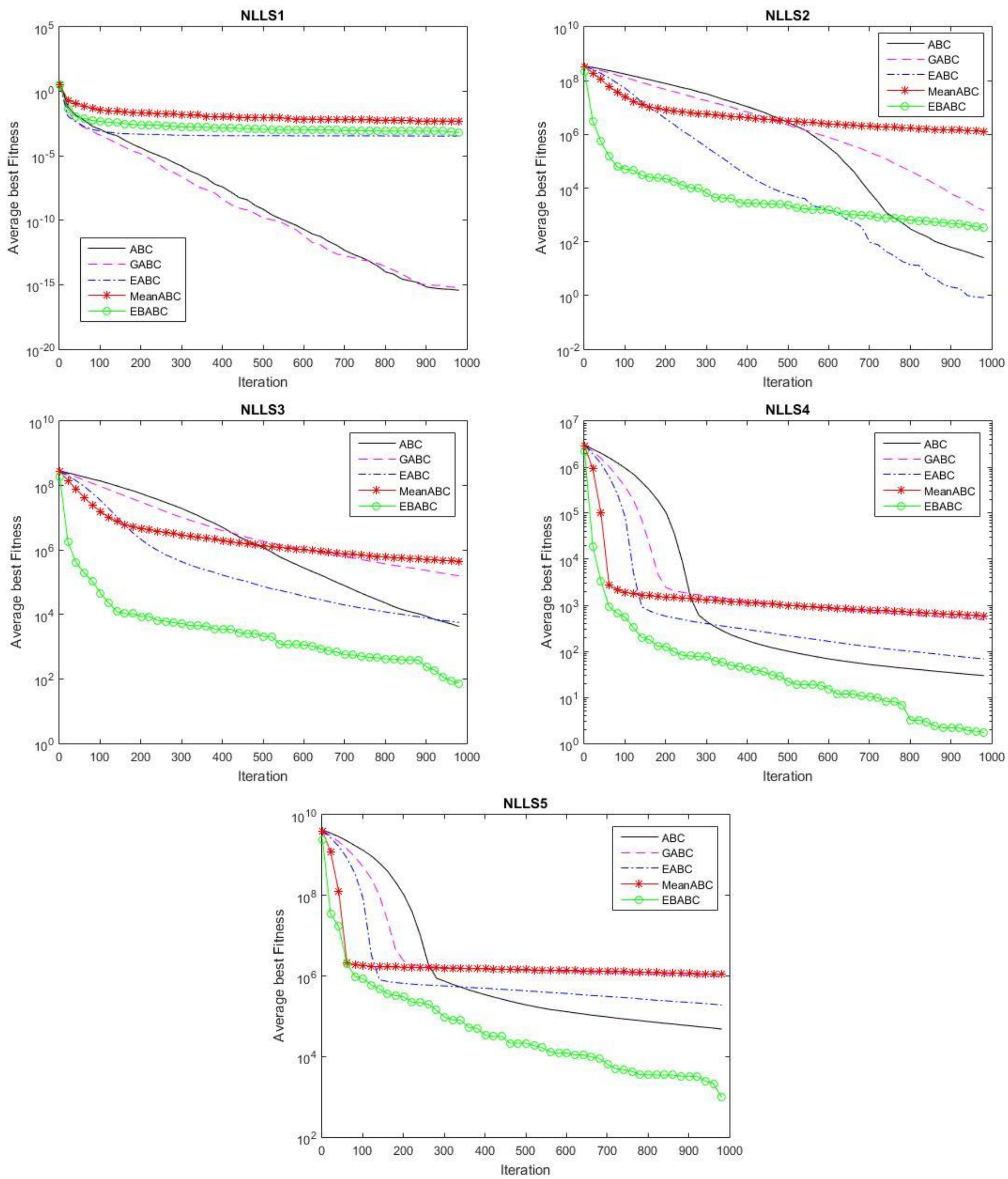

Fig. 5. Convergence curves of EBABC and other four ABC variant algorithms for NLLS problems.

The solution methods for NLLS are gradually established and developed. However, the traditional algorithms for solving NLLS relies heavily on the initial values, which often leads to the unsatisfactory results. As the intelligent algorithms are put forward, it became the effective method for solving NLLS. Generally, the NLLS problems can be defined as follows.

$$\min_{x\geq 0} f(\mathrm{x}) = \frac{1}{2}\| Ax-b \|^2 = \frac{1}{2}(Ax-b)^T(Ax-b) ,$$

where $A \in R^{m\times m}$, $m \geq n$, $rank(A)=n$, $b \in R^m$.

For testing the effectiveness of EABC, we conducts experimental comparison by solving 5 NLLS problems in literature [24]. And the experimental results are shown in Table VI.

From Table VI, contrast with the other listed variant ABC algorithms, EBABC has the better optimal capability on NLLS 3 - NLLS 5. Moreover, in Table VI the total rank of ABC, GABC, EABC, MeanABC and EBABC are 2, 4, 3, 5 and 1 respectively. According to the last two rows in Table VI, the total rank and the final rank both indicate that EBABC has the best optimal capability over the other four algorithms.

Furthermore, in Fig. 5 and Fig. 6 the convergence curves and the boxplots are given to illustrate the performance of EBABC more visually. All these figures show that EBABC can find a better solution in fewer iterations, and more stable than other four variant algorithms.

V. Conclusion

To enhance the optimal searching capability of metaheuristic algorithms, achieving the balance between exploratory and exploitative abilities is crucial. This paper presented the empirical balanced artificial bee colony

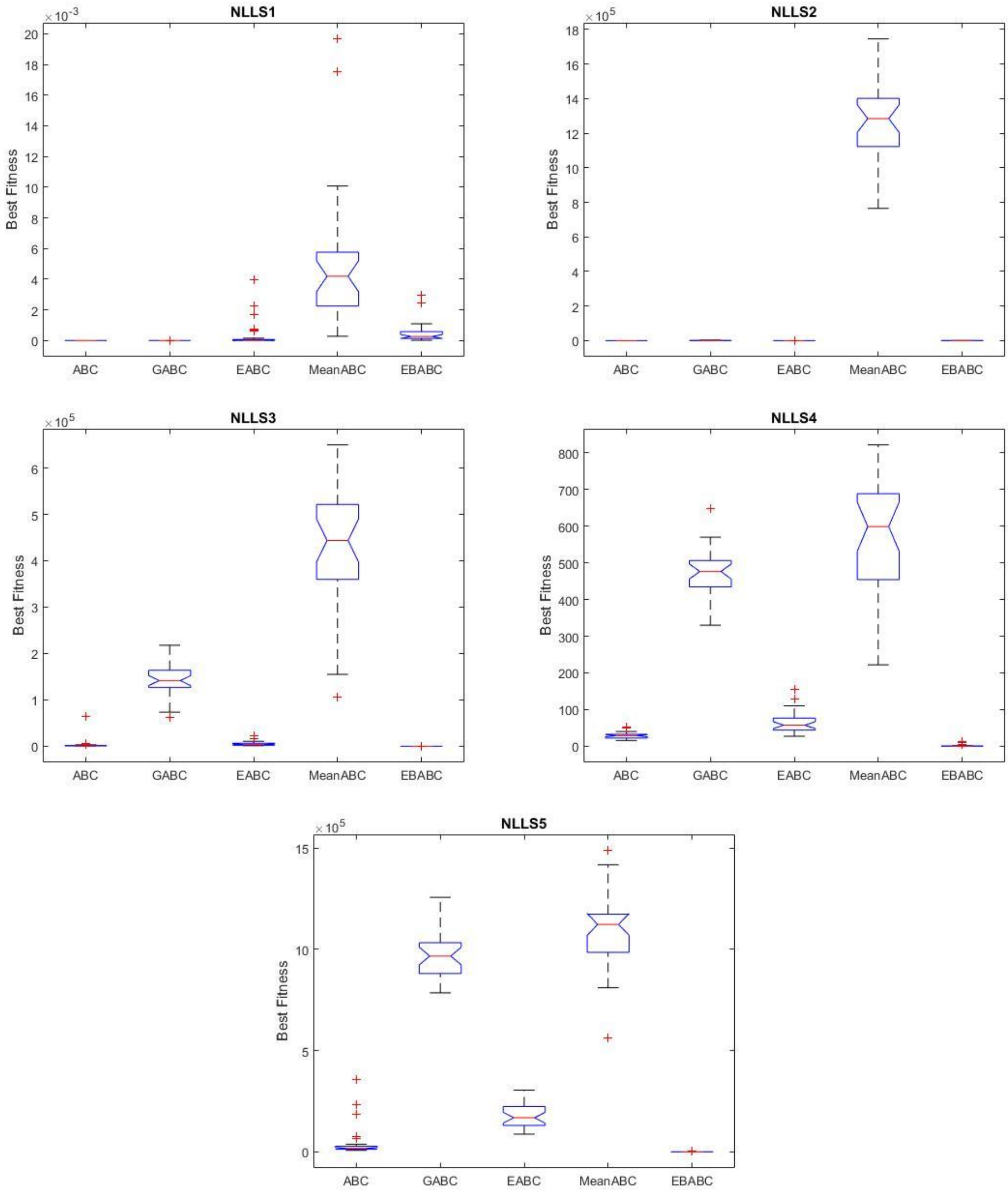

Fig. 6: Boxplots of EBABC and other four ABC variant algorithms for NLLS problems.

(EBABC) algorithm as a solution to this challenge. In EBABC, two searching equations were designed with different purposes and were adaptive selected throughout all searching stages. Additionally, the algorithm utilized the information of both a randomly selected individual and the global optimal individual at present to strike a balance between exploratory and exploitative abilities. Furthermore, a disturbance strategy for the global optimal individual at present was introduced to enhance the algorithm's diversity. Experimental results demonstrated that EBABC is superior to the other four ABC variants in terms of optimal performance.

REFERENCES

[1] K. S. Tang, K. F. Man, S. Kwong and Q. He, "Genetic algorithms and their applications," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 22-37, 1996.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization", *In IEEE International Conference on Neural Networks*, Piscataway, NJ, 1995, pp. 1942-1948.

[3] Y. Xing, J. Wang, Y. Zheng and Y. Wang, "Particle Swarm Optimization Algorithm Based on Different Inertia Weights for Solving the P-Hub Allocation Problem," *IAENG International Journal of Applied Mathematics*, vol. 52, no. 4, pp. 1025-1039, 2022.

[4] Y. Hou, C. Wang, W. Dong and L. Dang, "An Improved Particle Swarm Optimization Algorithm for the Distribution of Fresh Products," *Engineering Letters*, vol. 31, no. 2, pp. 494-503, 2023.

[5] C. Wang, P. Shang and X. Wu, "Hierarchical Particle Swarm Optimization Based on Mean Value," *IAENG International Journal of Applied Mathematics*, vol. 53, no. 2, pp. 695-703, 2023.

[6] M. Dorigo and T. Stutzle, *Ant Colony Optimization*, Cambridge: MAMIT Press, 2004.

[7] R. Storn and 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.
[8] D. Karaboga, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Technical report-tr06, Kayseri, Turkey: Erciyes University, 2005.
[9] X. S. Yang, *Firefly Algorithms for Multimodal Optimization*, Lecture Notes in Computer Science, vol 5792. Springer, Berlin, Heidelberg, 2009.
[10] J. Zhang, X. Xie, M. Wang and M. Zhang, "A Hybrid Firefly Algorithm with Butterfly Optimization Algorithm and its Application," *Engineering Letters*, vol. 30, no. 2, pp. 453-462, 2022.
[11] A. Das, H. S. Das and H. S. Das, *Impact of Cuckoo Algorithm in Speech Processing*, in: Applications of Cuckoo Search Algorithm and Its Variants, Springer, 2021, pp. 207-228.
[12] G. Zhu and S. Kwong, “Gbest-guided artificial bee colony algorithm for numerical function optimization,” *Applied Mathematics and Computational*, vol. 217, no. 7, pp. 3166-3173, 2010.
[13] A. Banharnsakun, T. Achalakul and B. Sirinaovakul, “The best-so-far selection in artificial bee colony algorithm,” *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2888-2901, 2011.
[14] W. F. Gao, L. L. Huang, S. Y. Liu, F.T. Chan, C. Dai and X. Shan, “Artificial bee colony algorithm with multiple search strategies,” *Applied Mathematics and Computation*, no. 271, pp. 269-287, 2015.
[15] W. F. Gao and S. Y. Liu, “A modified artificial bee colony algorithm,” *Computers & Operations Research*, vol. 39, no. 3, pp. 687-697, 2012.
[16] W. F. Gao, S. Y. Liu and L. L. Huang, “A novel artificial bee colony algorithm based on modified search equation and orthogonal learning,” *IEEE Transactions on Cybernetics*, vol. 43. no. 3, pp. 1011-1024, 2013.
[17] W. F. Gao, S. Y. Liu and L. L. Huang, “Enhancing artificial bee colony algorithm using more information-based search equations,” *Information Sciences*, no. 270, pp. 112–133, 2014.
[18] Y. Xue, J. Jiang, B. Zhao and T. Ma, “A self-adaptive artificial bee colony algorithm based on global best for global optimization,” *Soft Computing*, vol. 22, no. 9, pp. 2935-2952, 2018.
[19] L. Cui, G. Li, Q. Lin, Z. Du, W. Gao, J. Chen and N. Lu, “A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation,” *Information Sciences*, no. 367, pp. 1012-1044, 2016.
[20] D. Karaboga and B. Gorkemli, “A quick artificial bee colony (qabc) algorithm and its performance on optimization problems,” *Applied Soft Computing Journal*, no. 23, pp. 227-238, 2014.
[21] H. Wang, W. Wang, X. Zhou, J. Zhao, Y. Wang, S. Xiao and M. Xu, “Artificial bee colony algorithm based on knowledge fusion,” *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1139-1152, 2021.
[22] C. Wang, P. Shang and L. Liu, "Improved Artificial Bee Colony Algorithm Guided by Experience," *Engineering Letters*, vol. 30, no. 1, pp. 261-265, 2022.
[23] A. Alrosan, W. Alomoush, N. Norwawi, M. Alswaitti and S. N. Makhadmeh, “An improved artificial bee colony algorithm based on mean best-guided approach for continuous optimization problems and real brain MRI images segmentation,” *Neural Computing & Applications*, vol. 33, no. 5, pp. 1-27, 2021.
[24] J. Chen, L. Yong, X. Kong and Z. Wang, “A Novel Global Harmony Search Algorithm for Nonnegative Linear Least Squares,” *Journal of Information and Computational Science*, vol. 10, no. 5, pp. 1424-1433, 2013.