Computer Science and Information Systems 16(3):939–957 https://doi.org/10.2298/CSIS180923034D

Enhanced Artificial Bee Colony with Novel Search Strategy and Dynamic Parameter

Zhenxin Du[1,2] and Keyin Chen[3,4]

[1] School of Computer Information Engineering,Hanshan Normal University
Chaozhou 521041, China
duzhenxinmail@163.com
[2] College of Information Engineering, Shanghai Maritime University
Shanghai 201306, China
1355121995@qq.com
[3] Nanjing Research Institute for Agricultural Mechanization, Ministry of Agriculture and Rural Affairs
Nanjing 210014, China
308829497@qq.com
[4] School of Information and Communication Engineering, HeZhou University
Hezhou 542899, China

Abstract. There is only one guiding solution in the search equation of Gaussian bare-bones artificial bee colony algorithm (ABC-BB), which is easy to result in the problem of premature convergence and trapping into the local minimum. In order to enhance the capability of escaping from local minimum without loss of the exploitation ability of ABC-BB, a new triangle search strategy is proposed. The candidate solution is generated among the triangle area formed by current solution, global best solution and any randomly selected elite solution to avoid the premature convergence problem. Moreover, the probability of crossover is controlled dynamically according to the successful search experience, which can enable ABC-BB to adapt all kinds of optimization problems with different landscapes. The experimental results on a set of 23 benchmark functions and two classic real-world engineering optimization problems show that the proposed algorithm is significantly better than ABC-BB as well as several recently-developed state-of-the-art evolution algorithms.

Keywords: artificial bee colony, triangle search, dynamic parameter, engineering optimization.

1. Introduction

With the continuous development of science and technology, many global optimization problems constantly exist in all most of engineering and science fields, such as queuing system [46]and structural design [4][16]. Unfortunately, many of these problems are often characterized as non-convex, discontinuous or non differentiable, thus it is difficult to deal with them with traditional optimization algorithms. Evolution algorithms (EAs), as a powerful tool, are playing an increasingly important role in solving such kind of problems, such as Particle Swarm Optimization (PSO) [42], Ant Colony Optimization (ACO) [18], Differential Evolution (DE) [43], Artificial Bee Colony (ABC) algorithm [30]. ABC is a new EA proposed by Karaboga, which is drawn inspiration from the intelligent foraging behavior of honey bees. The comparison results indicate that ABC performs competitively and effectively when compared to the selected state-of-the-art EAs, such as DE and PSO [31]. Owing to its simple structure, easy implementation and outstanding performance, ABC has received increasing interest and has been widely used in many engineering optimization problems since its invention such as network problems [36], image problems [9], engineering problems [29], clustering problems [20][32][34].

However, similar to other EAs, ABC also tends to suffer from the problem of poor convergence. The possible reason is that the solution search equation which is used to generate new candidate solutions, has good exploration capability but poor exploitation capability [21][49], and thereby it causes the problem of slow convergence speed. Therefore, the performance of ABC can be improved by enhancing the exploitation ability and finding better balance between exploration and exploitation. A large number of improved ABC variants have been presented by exploiting the valuable information from the current best solution or other good solutions. Firstly, Zhu and Kwong proposed [49] a global best (gbest)-guided ABC (GABC) algorithm based on the inspiration of PSO. In GABC, the information of the gbest solution is incorporated into the solution search equation of ABC to improve the exploitation. However, as claimed in [25], the search equation of GABC may cause an oscillation phenomenon and thus may also degrade convergence since the guidance of the last two terms may be in opposite directions. Afterwards, Zhou [48] proposed an improved ABC with Gaussian bare-bones search equation (ABC-BB for short) algorithm based on the utilization of the global best solution to make up the defect of GABC. Moreover, they proposed an ensemble algorithm composed of ABC-BB and general opposition learning initialization strategy (GOBL), named GBABC. In ABC-BB, an important feature of the newly proposed search equation is that positions of the new food sources are sampled through a Gaussian distribution with dynamical mean value and variance value. The experimental results demonstrate the effectiveness of ABC-BB and GBABC in solving complex numerical optimization problems when compared with other algorithms.

However, in ABC-BB and GBABC, there is only one guiding individual in the search equation of Gaussian bare-bones artificial bee colony algorithm (ABC-BB), which is easy to result in the problem of premature convergence. In order to overcome this drawback, a new triangle search strategy is proposed in this paper. The candidate solution is generated among the triangle area formed by current solution, global best solution and randomly selected elite solution, which is beneficial to augment the search area and prevent premature convergence. Moreover, the probability of crossover is controlled dynamically according to the successful search experience, which can enable ABC-BB to adapt all kinds of optimization problems with different landscapes. The experimental results show that the proposed algorithm is significantly better than ABC-BB as well as several recently-developed variants of PSO, DE and ABC.

The rest of this paper is organized as follows. In section 2, the related works are briefly reviewed, including the basic ABC and some improved ABCs. In section 3, we present the proposed approach in detail. Section 4 presents and discusses the experimental results. Finally, the conclusion is drawn in section 5.

2. Related Works

2.1. Basic ABC

The ABC algorithm is a swarm-based stochastic optimization method, which is inspired by the waggle dance and intelligent foraging behavior of honey bees. In ABC, the position of a food source denotes a possible solution of the optimization problem, and the nectar amount of each food source denotes the quality (fitness) of the corresponding solution. In order to find the best food source, three different types of bees, i.e., employed bee, onlooker bee and scout bee, are responsible for the different search tasks. Firstly, the first half of the colony consists of employed bees, which are responsible for randomly searching better food source in the neighborhood of the corresponding parent food source and then passing information of the food sources to onlooker bees. Secondly, the second half of the colony is composed of onlooker bees, which further search the better food sources according to the information provided by employed bees. Thirdly, if the quality of a food source is not improved by a preset number of times (*limit*), this food source is abandoned by its employed bee, and then this employed bee becomes a scout bee that begins to seek a new random food source. The original ABC algorithm includes four phases, i.e., initialization phase, employed bee phase, onlooker bee phase and scout bee phase. After the initialization phase, ABC turns into a loop of employed bee phase, onlooker bee phase and scout bee phase until the termination condition is satisfied. The details of each phase are described as follows.

Initialization phase: Similar to other EAs, ABC also starts with an initial population of SN randomly generated food sources. Each food source x_i=($x_{i,1}$,$x_{i,2}$,...,$x_{i,D}$) are generated randomly according to Eq.(1).

$$x_{i,j} = a_j + rand\,(0,1)\,(b_j - a_j) \tag{1}$$

where *SN* is the number of food sources, and *SN* is equal to the number of employed and onlooker bees; *D* is the dimensionality (variables) of the search space; a_j and b_j are the lower and upper bounds of the *j*th variable respectively; $rand(0,1)$ is a random real number in range of [0,1]. Then, the fitness values of the food sources are calculated by Eq.(2).

$$fit_i = \begin{cases} 1/(1+f_i), f_i \geq 0 \\ 1+|f_i|\,, f_i < 0 \end{cases} \tag{2}$$

Employed bee phase :In this phase, each employed bee generates a new food source $v_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,D})$ in the neighborhood of its parent position $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$ using Eq.(3).

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) \tag{3}$$

where $k \in \{1, 2, ..., SN\}$ and $j \in \{1, 2, ...D\}$,are randomly chosen indexes; k has to be different from i;$\phi_{i,j}$is a random number in the range [-1,1]. If the new food source v_i is better than its parent x_i, then x_i is replaced with v_i.

Onlooker bee phase:After receiving the information from the employed bees, the onlooker bees begin to select food sources for exploitation. The probability of selecting a food source depends on its nectar amount, which can be calculated as follows.

$$p_i = fit_i / \sum_{i=1}^{SN} fit_i \quad (4)$$

Where p_i is the selection probability and fit_i is the fitness value of the ith food source. Once an onlooker bee completes its selection, it would also produce a modification on its chosen food source using Eq.(3). As in the case of the employed bees, the greedy selection method is employed to retain a better one from the old food source and the modified food source as well.

Scout bee phase:If a position x_i cannot be improved for at least $limit$ times, then that food source is assumed to be abandoned. Then, the scout produces a food source randomly as in Eq.(1) to replace x_i.

2.2. Improved ABCs

The solution search equation plays an important role in ABC. Eq.(3) indicates that the solution search equation of ABC is good at exploration but poor at exploitation. To improve the exploitation ability and utilize the beneficial information of the current best solution, Zhu and Kwong [49] proposed a new search equation (names GABC), shown in Eq.(5).

$$v_{i,j} = x_{i,j} + \phi_{i,j}.(x_{i,j} - x_{k,j}) + \psi_{i,j}.(x_{best,j} - x_{i,j}) \quad (5)$$

Where the third term in the right-hand side of Eq.(5) is a new added term called gbest term, $x_{best,j}$ is the jth element of the global best solution, is a uniform random number within [0,C], and C is a nonnegative constant and is suggested to set to 1.5. Although this new search equation has been shown superior to the original one, its mechanism of utilizing the gbest can still cause inefficiency to the search ability of the algorithm and slow down convergence speed. Because the guidance of the last two terms may be in opposite directions, and this can cause an oscillation phenomenon [25]. Therefore, in order to address these issues in Eq.(5), Zhou et al. [48] designed a Gaussian bare-bones search equation inspired by the concept of BBPSO, shown in Eq.(6).

$$v_{i,j} = \begin{cases} N((x_{i,j} + x_{best,j})/2, |x_{i,j} - x_{best,j}|), if\ rand_j \leq CR \\ x_{i,j}, otherwise \end{cases} \quad (6)$$

Where $N(.)$ represents a Gaussian distribution with mean and variance ; x_{best} is the global best solution of current population, $rand_j$ is a uniformly distributed random number within the range [0,1]; CR is a new introduced parameter which controls how many elements in expectation can be derived from its parent x_i for v_i. Since there is only one dimension of x_i to be updated for v_i in the original search equation, the introduction of CR is helpful to inherit more information from x_{best} to enhance the exploitation. In Eq.(6), new candidate solutions are generated in the search space formed by the current solution x_i and the global best solution x_{best}. Compared with the original search equation Eq.(3), and the Eq.(5), the Gaussian bare-bones search equation Eq.(6) has two advantages. First, Eq.(6) takes advantages of the global best solution to guide the search of new candidate solutions, which is beneficial to enhance the exploitation ability of ABC. Second, the oscillation phenomenon can be avoided because Eq.(6) can be considered as one single term. To better balance the exploration and exploitation, Eq.(6) is only used in the onlooker bee phase, while the employed bee phase still use the original solution search equation Eq.(3) for generating new candidate solutions. By adopting the new equation Eq.(6) in the onlooker bee phase, the newly proposed algorithm is named ABC-BB. Moreover, Zhou [48] also proposed a general opposition learning strategy (GOBL) in the scout bee phase. By combining the ABC-BB and GOBL, they [48] proposed a new ensemble algorithm (GBABC). Similarly, Cui et al. [15] also proposed a novel search equation to avoid the oscillation phenomenon of GABC, shown in Eq.(7).

$$v_{i,j} = \begin{cases} x_{i,j} + \phi_{i,j}.(x_{i,j} - x_{k,j}), if\ rand < P \\ x_{i,j} + \psi_{i,j}.(x_{best,j} - x_{i,j}), otherwise \end{cases} \tag{7}$$

Where P is a parameter defined by the user. x_{best} is the current best solution of the population, and x_i is the ith food source. x_k is a randomly selected food source from the population, which is different from x_i. is the uniformly distributed random number in the range of [-1, 1] and is a uniform random number in the range of [0, 1.5]. j is randomly selected from $1, 2, \ldots, D$. Obviously, with the guidance from only one term, the novel search strategy can easily avoid the oscillation phenomenon. Moreover, parameter P could be used to control how to appropriately exploit the valuable information of the current best solution. In 2015, Gao et al. [23] also proposed a kind of Gaussian search equation, shown in Eq.(8).

$$v_{i,j} = N((x_{k,j} + x_{best,j})/2, |x_{k,j} - x_{best,j}|) \tag{8}$$

Where k is a randomly chosen index from 1,2,*SN* with the constraint that $k \neq i$, the meaning of the other symbol is the same as Eq.(7). Gao et al. [23] proposed a novel ensemble ABC variants by combining Eq.(8), CABC [25], parameter adaption strategy and fitness-based neighborhood mechanism, named BCABC. The experimental results show that the performance of BCABC is better than some newly proposed state-of-the-art algorithms.

3. The Proposed Algorithm

In this section, we will firstly propose a novel enhanced ABC with triangle search strategy and parameter strategy, then the pseudo code of the novel ABC variants will be given.

3.1. The New Triangle Search Strategy

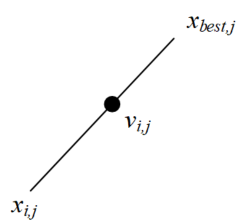
In the search equation Eq.(6) of ABC-BB, the new candidate $v_{i,j}$ is generated in the line determined by the current solution $x_{i,j}$ and the gbest $x_{best,j}$ solution. Fig.1(a) demonstrate the evolution process of Eq.(6). However, we can see from Fig.1(a) that there is only one guiding individual (gbest) in the search equation Eq.(6), which is easy to result in the problem of premature convergence and locally convergence because the population is always guided by the global best solution monotonically. In order to overcome this drawback, a new search equation based on triangle search strategy is proposed in this paper, shown in Eq.(9).

$$v_{i,j} = \begin{cases} N(\frac{x_{i,j}+x_{best,j}+x_{e,j}}{3}, \frac{|x_{i,j}-x_{best,j}|+|x_{best,j}-x_{e,j}|+|x_{e,j}-x_{i,j}|}{3}),\ \ if\ rand_j \leq CR \\ x_{i,j}, otherwise \end{cases} \tag{9}$$

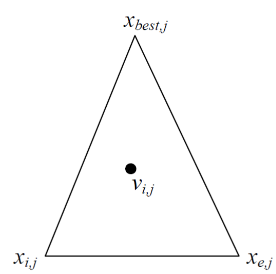
Where CR is the crossover probability, the meaning of $N(.)$ and x_{best} are the same as Eq.(6). x_e is a randomly chosen from the elite solutions (the top $p \cdot SN$solutions in current population, $p \in (0, 1)$. In Eq.(9), the candidate solution $v_{i,j}$ is generated among the triangle area determined by the current solution, the global best solution and a randomly selected elite solution, which is beneficial to augment the search area, increasing information sharing and thereby prevent premature convergence. The search equation of Eq.(9) is given in Fig.1(b).

Li et at.[35] indicated that "the more information is efficiently utilized to guide the flying, the better performance the PSO algorithm will have". Now that Eq.(9) uses more elite solutions to guide the search, we have reason to believe that the novel search equation will decrease the premature problem of ABC-BB and enhance the performance of ABC-BB.

In EABC-BB, the mutation strategy uses more information of elite solutions. Because of its randomness characteristic, it may hold down the premature problem of ABC-BB and enable ABC-BB have more chance to explore more peaks denoted by different elite solutions.



(a) ABC-BB



(b) EABC-BB

Fig. 1. Evolution process of a solution according to (a):Eq.(6) and (b):Eq.(9) in 2-D parametric space.

3.2. Parameter Adaption Strategy

Like other DE variants, the parameter CR also greatly affects the performance of ABC-BB [48]. The experimental results of literature [48] show that a higher value of CR is more suitable for solving unimodal problems, while a lower one is better for multimodal problems. Therefore, it is difficult to determine the optimal control parameters because they are problem dependent. In this paper, a simple self-adaptive strategy is proposed to dynamically update CR. In some well-known self-adaptive DE algorithms, such as SaDE [41] and JADE [47], the initial value of CR is independently generated by a normal distribution of mean 0.5 and standard deviation 0.1. After a predefined number of generations, the CR is updated according to the search experiences of successful crossover probabilities. By following this idea, a new self-adaptive CR strategy is proposed as follows:

$$CR = N(\bar{S}_{CR}, 0.1) \tag{10}$$

Where $N(\bar{S}_{CR}, 0.1)$ represents a normal distribution whose mean and standard deviation are $\bar{S}_{CR}$ and 0.1, respectively. $\bar{S}_{CR}$ is set to be 0.3 considering that it is suitable for CR in literature [48], then the value will be adjusted dynamically at the end of each generation as follows:

$$\bar{S}_{CR} = \text{mean}(S_{CR}) \tag{11}$$

where mean(.) is the general arithmetic average; and S_{CR} is the set of all successful *CR* at generation g.

By combining the novel search equation Eq.(9) and the adaptive parameter adaption strategy, a novel ABC variants, named enhanced ABC-BB (EABC-BB for short), is proposed. The pseudo code of EABC-BB is described in Algorithm 1, where *FE*s is the number of consumed fitness function evaluations, and max_*FE*s, as the stopping criterion, is the maximal number of fitness function evaluations. $trial_i$ records the unchanged times of x_i's fitness value.

Clearly, Compared to ABC-BB, EABC-BB only modifies the search equation and the parameter adaption strategy. Therefore, EABC-BB does not increase the ABC-BB's complexity any more, both EABC-BB and the ABC-BB have the same time complexity O($g_{max} \cdot$ *SN*$\cdot$ *D*), where g_{max} is the maximum number of generations.

Algorithm 1:Pseudo code of the proposed EABC-BB

```
Initialization:Generate SN solutions that contain D variables according to Eq.(1), FEs=0
while FEs< max_FEs
  for i=1 to SN //employed bee phase
    Generate a new solution v_i using Eq.(3), then calculate f(v_i)
    if f(v_i)<f(x_i)
        Replace x_i by v_i and trial_i=0
    else
        trial_i=trial_i +1
    end if
  end for //end employed bee phase
    Select the top p.SN solutions as elite solutions from population
  for i=1 to SN //onlooker bee phase
        Select a solution x_e from elite solutions randomly to search
        Generate a new candidate solution v_e in the neighborhood of x_e using Eq.(9) and calculate f(v_e)
    if f(v_e)< f(x_e)
        Replace x_e by v_e and trial_i=0
    else
        trial_e=trial_e +1
    end if
  end for //end onlooker bee phase
    FEs = FEs + SN × 2        Select the solution x_max with maximum trial value //Scout bee phase
  if trial(max)>limit //only one food source with max trial value can be initialized
        Replace x_max by a new solution generated according to Eq.(1), FEs=FEs+1, trial (max)=0;
  end if //end scout bee phase
end while
Output the food source with the smallest objective value
```

4. Computational Study

In this section, the experimental results will be given. Specifically, in section 4.1, the benchmark functions will be given, then in section 4.2, the new parameter p will be analyzed. In section 4.3, the proposed EABC-BB will be compared with other recently-proposed ABC variants. The section 4.4 give the comparisons between the EABC-BB and some newly proposed state-of-the-art EAs. At last, the new EABC-BB will be applied to solve two well-known engineering optimization problems.

4.1. Benchmark Functions

In this section, 23 well-known benchmark functions [48] with different dimensions (D=30 and D=50, respectively) are employed to validate the performance of EABC-BB. The 13 test functions are scalable problems. Among these problems, F01-F04 are unimodal func-tions, and F05 is the Rosenbrock function which is multimodal when $D>3$. F06 is a step function which has one minimum and is discontinuous, while F07 is a noisy quartic function. F08-F13 are multimodal functions with many local minima. The remaining 10 functions (F14-F23) are shifted and/or rotated types taken from the CEC 2005 competition[44]. The definitions of these functions are given in [48].

4.2. Adjusting the Parameter p

In EABC-BB, only one parameter p is used whose value may affect the performance. Therefore, in the subsection, we investigate different p values to select the best one for maximizing the performance of EABC-BB. The available p values are in the range [0.05,0.2] in steps of 0.05, i.e., there are four choices for p in total. F01-F13 are selected as test functions. All the selected 13 functions are tested at D=30, the maximal number of fitness function evaluations (*max_FEs*) is set to 5,000·D. The number of food sources *SN* and *limit* are set at 30 and 100, respectively. Each function is run 30 times, and the mean error ($f(x)$-$f(x^*)$), x^* is the global optimum) and standard deviation values are recorded. The experimental results are listed in Table 1. In the last line of Table 1, in order to compare the significance between two parameters values, the Wilcoxon singed-rank test [17]is conducted. The Wilcoxon signed-rank test is a non-parametric statistic hypothesis test, which can be used as an alternative to the paired t test when the results cannot be assumed to be normally distributed. "†", "‡" and "≈" indicate the EABC-BB with p=0.1 is better than, worse than, and similar to the EABC-BB with other p values. As seen from Table 1, the EABC-BB at p=0.1 performs best, while a higher or lower p value will weaken the performance of EABC-BB. Therefore, the parameter p is set at 0.1 in the following all experiments.

4.3. Comparison with Other ABCs

This subsection presents a comparative study of EABC-BB with MGABC [15], BC-ABC [23], ABC-BB [48] at both D=30 and 50. These three algorithms are all proposed recently and have relatively good performance according to their reports. Moreover, each of the three algorithms have proposed a kind of improved search equations, respectively.

Table 1. The impact of different *p* values on the EABC-BB performance for 13 test functions

Func	*p*=0.05	*p*=0.15	*p*=0.2	*p*=0.1
	Mean error(std dev)	Mean error(std dev)	Mean error(std dev)	Mean error(std dev)
F01	1.45E-34(2.51E-34)†	1.69E-76(2.92E-76)†	2.79E-76(3.78E-76)†	4.66E-81(3.29E-80)
F02	4.86E-11(7.39E-11)†	1.87E-56(3.25E-56)‡	8.02E-48(1.38E-47)‡	1.69E-41(1.08E-40)
F03	2.05E+02(1.64E+02)†	1.88E+03(1.79E+03)†	2.63E+03(6.39E+02)†	1.15E+02(2.16E+02)
F04	2.25E+01(1.33E+02)†	5.36E-03(6.73E-03)‡	3.71E-03(2.37E-03)‡	6.40E-01(9.95E-01)
F05	1.16E+02(7.20E+01)†	1.75E+00(2.36E+00)‡	1.12E+01(4.49E+00)‡	1.52E+01(4.44E+00)
F06	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
F07	5.13E-03(1.25E-03)†	8.71E-03(3.10E-03)†	8.65E-03(2.42E-03)†	2.74E-03(3.56E-03)
F08	3.94E+01(6.83E+01)†	3.94E+01(6.83E+01)†	1.33E-02(9.96E-09)†	3.82E-04(6.84E-12)
F09	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
F10	3.04E-14(2.14E-14)†	7.99E-15(0.00E+00)†	7.99E-15(0.00E+00)†	3.39E-15(4.24E-15)
F11	7.37E-03(1.27E-02)†	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
F12	3.32E-26(5.74E-26)†	1.57E-32(4.53E-32)†	1.57E-32(7.31E-32)†	6.28E-33(7.77E-33)
F13	6.97E-32(1.18E-31)†	1.50E-33(5.24E-34)†	1.50E-33(2.38E-33)†	5.99E-34(7.42E-34)
†‡≈	11/0/2	7/3/3	7/3/3	–

Note that the GBABC proposed in literature [48] is added an additional initialization method (General Opposite Based Learning, GOBL) at ABC-BB, this is unfair to the other algorithms because they are all randomly initialized. Therefore, We only use the ABC-BB for the comparison.

For a fair comparison, all the competitive algorithms have the same parameter settings, i.e., *SN*=30, *limit*=100, *Max_FEs*=5000·*D* for F01-F13 and *max_FEs*=10,000·*D* for F14-F23.For other specific parameters, *C*=1.5 and *P*=0.3 for MGABC, *CR*=0.3 for ABC-BB, which are the same as in the original literatures. Each algorithm is run 30 times per function, and the mean error and standard deviation values are presented in Tables 2 and 3 for *D*=30 and 50, respectively. Moreover, to compare the significance between two algorithms, the paired Wilcoxon signed-rank[17] test is used. "†", "‡" and "≈" indicated EABC-BB is better than ,worse than, and similar to its competitor according to the Wilcoxon signed-ranked test at α=0.05.

For *D*=30, from the results presented in Table 2, it is clear that EABC-BB performs significantly better than the other three competitors on the majority of test functions. To be specific, EABC-BB outperforms MGABC on 17 out of 23 test functions, while MGABC only achieves better result on the F05 (Rosenbrock) and F19 (Shifted Rosenbrock). As F05 and F19 are Rosenbrock and shifed Rosenbrock functions and their global optimum is inside a long, narrow, parabolic shaped flat valley, the variables are strongly dependent, and the gradients do not generally point towards the optimum. If the population is guided by the global best solution or some other good solutions, the search will fall into some unpromising areas[14]. Therefore, EABC-BB is beaten by MGABC at these two functions. As a multi strategies ensemble algorithm, BCABC also performs better than EABC-BB on functions F01, F05, F16 and F19, while EABC-BB beats it on the other 16 functions. For ABC-BB, EABC-BB wins on16 functions and only loses on function F04 and F19.

Table 2. The results achieved by MGABC, BCABC, ABC-BB and EABC-BB at *D*=30 (Mean error±std dev)

Func.	MGABC	BCABC	ABC-BB	EABC-BB
F01	1.66E-62±4.05E-62†	3.46E-85±7.21E-84‡	4.89E-48±2.28E-48†	4.66E-81±3.29E-80
F02	2.43E-32±1.48E-32†	3.92E-36±5.81E-37†	2.36E-29±6.17E-29†	1.69E-41±1.08E-40
F03	5.36E+03±1.20E+03†	2.17E+03±1.29E+03†	3.51E+03±2.19E+02†	1.15E+02±2.16E+02
F04	1.06E+00±2.37E-01†	1.55E+00±3.62E+00†	2.32E-02±5.17E-03‡	6.40E-01±9.95E-01
F05	5.95E-01±1.02E+00‡	5.30E+00±3.17E+00‡	2.15E+01±3.66E+00†	1.52E+01±4.44E+00
F06	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
F07	2.37E-02±5.24E-03†	1.01E-02±8.83E-02†	1.84E-02±8.21E-03†	2.74E-03±3.56E-03
F08	3.82E-04±2.64E+01≈	3.82E-04±4.83E-12≈	3.82E-04±5.75E-12≈	3.82E-04±6.84E-12
F09	0.00E+00±0.00E+00≈	2.43E-12±6.17E-13†	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
F10	3.67E-14±3.43E-15†	7.72E-14±5.85E-14†	1.46E-14±2.59E-15†	3.39E-15±4.24E-15
F11	0.00E+00±0.00E+00†	6.79E-12±3.27E-12†	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
F12	1.57E-32±2.80E-48†	1.57E-32±8.38E-38†	1.57E-32±7.09E-46†	6.28E-33±7.77E-33
F13	1.49E-33±2.87E-33†	1.35E-32±8.13E-32†	1.35E-32±7.96E-48†	5.99E-34±7.42E-34
F14	5.96E-14±1.27E-14†	5.68E-14±2.73E-14†	6.82E-14±2.77E-14†	4.11E-14±1.72E-14
F15	6.47E+03±1.39E+03†	6.17E+03±3.66E+03†	3.67E+02±2.26E+02†	9.14E-02±2.96E-01
F16	8.88E+06±2.09E+06†	1.54E+06±5.23E+06‡	6.11E+06±1.33E+06†	4.83E+06±4.89E+06
F17	3.29E+04±4.94E+03†	7.52E+03±4.81E+03†	4.92E+03±2.92E+03†	3.14E+03±1.14E+03
F18	8.38E+03±1.73E+03†	2.73E+03±4.63E+02†	2.04E+03±6.73E+02†	1.86E+03±8.12E+02
F19	1.36E+01±2.14E+01‡	7.42E+01±3.99E+01‡	6.42E+01±8.17E+01‡	8.19E+01±8.35E+01
F20	4.91E-01±2.83E-01†	2.20E-02±3.89E-02†	2.32E-02±3.57E-02†	2.11E-02±5.64E-02
F21	2.09E+01±4.80E-02≈	2.09E+01±3.37E-02≈	2.09E+01±6.86E-02≈	2.09E+01±5.59E-02
F22	5.40E-13±1.27E-14†	6.82E-13±4.16E-13†	7.57E-14±3.62E-14†	5.22E-14±1.55E-14
F23	2.14E+02±2.30E+01†	1.12E+02±4.82E+01†	1.22E+02±3.46E+01†	1.07E+02±2.13E+01
†/‡/≈	17/2/4	16/4/3	16/2/5	-

In order to investigate the scalability of EABC-BB, we also compare EABC-BB with all the competitors on the 23 test functions with 50*D*. The experimental results are given in Table 3. As seen from Table 3, EABC-BB consistently gets significantly better results than its competitors. Overall, EABC-BB outperforms MGABC, BCABC and ABC-BB on 15, 18 and 15 out of 23 functions. As can be seen from Tables 2 and 3, it is clear that EABC-BB is the best algorithm among four algorithms.

4.4. Comparison with Some Newly Proposed EAs

To further investigate the performance of EABC-BB, we compare EABC-BB with four newly proposed EAs, including DE and PSO variants and an ensemble algorithm GBABC:

- sinusoidal differential evolution algorithm (sinDE) (Draa et al. 2015) [19]
- Gaussian bare-bones artificial bee colony (GBABC) (Zhou et al. 2016), note that GBABC=ABC-BB+GOBL [48].
- Self regulating particle swarm optimization algorithm (SRPSO) (Tanweer et al. 2015) [45]
- Social Learning Particle Swarm Optimization (SLPSO) (Cheng et al. 2015) [10]

For a fair comparison, the control parameters of four competitive EAs are set to the suggested values offered by their corresponding literatures.

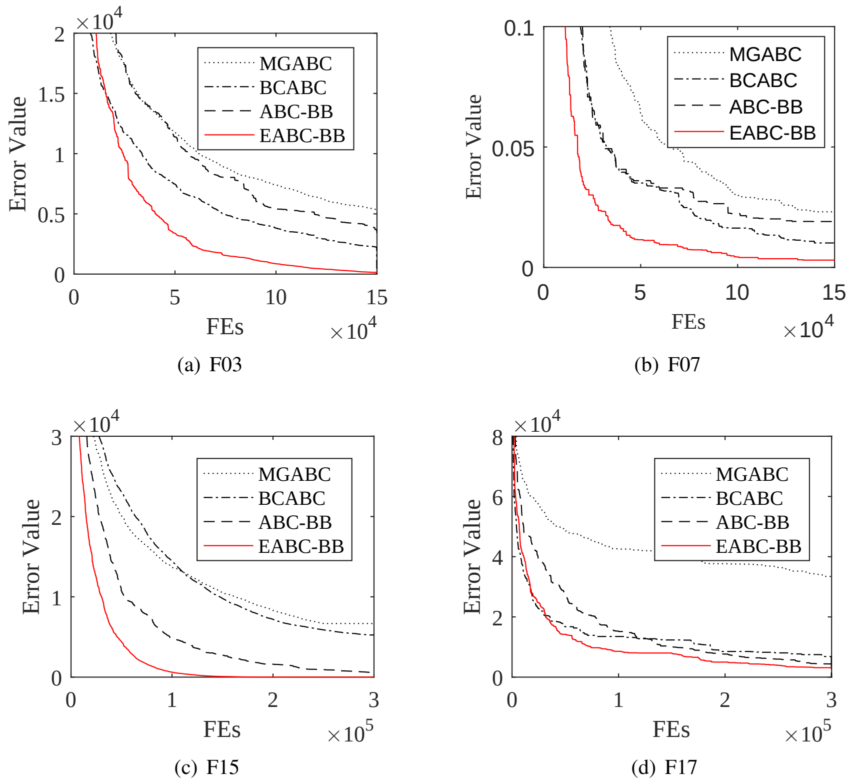


Fig. 2. Convergence performance of different ABCs on 4 functions

The stopping criterion is the same as previous subsections. Each algorithm is run 30 times per function, and the mean error and standard deviation values are given in Table 5. From Table 5, we can see that EABC-BB obtains significantly better results on the majority of test functions compared with other competitors. To be specific, sinDE wins on 5 test functions compared with EABC-BB, but on the other 17 test functions EABC-BB performs better.

SRPSO adopt self regulation strategy to control the premature convergence problem, which is similar to EABC-BB. Therefore, SRPSO has excellent performance. But in contrast to EABC-BB, SRPSO outperforms EABC-BB on only 6 test functions, but loses on 15 functions. SLPSO performs better than EABC-BB on only 2 functions, while EABC-BB outperforms it on other 21 test functions. GBABC is composed of ABC-BB and GOBL strategy, it outperforms EABC-BB on 5 functions and loses on 14 test functions compared with EABC-BB.

Overall, it is clear that EABC-BB is the best algorithm among 5 newly improved algorithms.

The convergence curves of MGABC, BCABC, ABC-BB and EABC-BB on four representative functions are shown in Fig.2. As seen from Fig.2, EABC-BB has faster convergence ability than other algorithms. The reason why EABC-BB have an advantage over

Table 3. The results achieved by MGABC,BCABC,ABC-BB and EABC-BB at *D*=50 (Mean error±std dev)

Func.	MGABC	BCABC	ABC-BB	EABC-BB
F01	3.22E-61± 2.25E-61†	5.85E-09±1.17E-08†	3.86E-41±4.51E-41†	6.84E-63± 1.53E-62
F02	2.84E-31± 1.63E-31‡	1.70E-09±3.41E-09†	2.87E-26±1.01E-26†	1.20E-29± 1.64E-29
F03	2.37E+04± 7.16E+02†	1.70E-09±3.58E+03†	3.53E+04±3.11E+03†	2.14E+04± 9.16E+03
F04	2.28E+01± 4.08E+00†	1.09E+01±6.97E-01†	1.25E+00±6.47E-01†	1.22E+00± 5.98E-01
F05	1.25E+00±1.92E+00‡	1.90E+01±3.42E+01‡	9.60E+01±4.94E+01†	5.32E+01± 2.78E+01
F06	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
F07	0.26E+00± 4.34E-02†	4.37E-02±2.35E-02†	3.82E-02±6.45E-03†	1.43E-02± 1.68E-03
F08	2.27E-11± 3.48E-12†	2.09E+04±1.83E-10†	2.10E+04±6.83E+01†	0.00E+00± 0.00E+00
F09	0.00E+00±0.00E+00≈	6.13E-04±1.10E-03†	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
F10	7.37E-14± 4.58E-15†	5.02E-06±1.00E-05†	3.01E-14±1.77E-15†	1.86E-14± 3.55E-15
F11	0.00E+00±0.00E+00≈	6.36E-03±5.70E-03†	0.00E+00±0.00E+00≈	0.00E+00±6.29E-03
F12	9.42E-33±0.00E+00†	5.49E-11±1.09E-10†	9.42E-33±0.00E+00†	4.99E-33±1.08E-33
F13	1.50E-33±0.00E+00≈	4.58E-33±5.37E-33†	1.50E-33±0.00E+00≈	1.50E-33±0.00E+00
F14	1.56E-13± 2.84E-14†	5.80E-09±1.16E-08†	1.27E-13±2.84E-14†	5.68E-14±0.00E+00
F15	2.97E+04± 3.60E+02†	2.63E+04±3.55E+03†	1.93E+04±4.60E+03†	6.36E+03±6.08E+03
F16	3.07E+07± 3.83E+06‡	3.19E+07±2.88E+06‡	3.20E+07±7.64E+06‡	4.00E+07±1.02E+07
F17	1.16E+05± 1.03E+04†	1.15E+05±1.41E+04†	4.96E+04±5.55E+03‡	5.61E+04±2.95E+03
F18	2.07E+04± 1.28E+03†	2.02E+04±9.02E+02†	6.88E+03±2.38E+03‡	7.49E+03±4.9E+03
F19	4.67E+00± 6.93E+00‡	4.12E+01±5.38E+01‡	6.85E+01±7.17E+01‡	8.04E+01±8.63E+00
F20	5.52E-01± 2.39E-01†	3.95E-02±4.29E-02†	5.61E-06±7.78E-06†	1.52E-02±1.74E-02
F21	2.11E+01± 2.39E-02†	2.11E+01±2.33E-02†	2.11E+01±2.89E-02†	2.11E+01±2.35E-02
F22	1.13E-13±0.00E+00†	4.79E-09±7.37E-09†	1.27E-13±2.84E-14†	7.95E-14±3.11E-14
F23	6.72E+02± 8.41E+01†	2.73E+02±3.91E+01‡	3.08E+02±2.20E+01†	2.75E+02±2.51E+01
†/‡/≈	15/4/4	18/4/1	15/4/4	–

Table 4. Parameter settings for sinDE, SRPSO, SLPSO, GBABC and EABC-BB

Algorithms	Parameter settings
sinDE	$NP = 30, freq = 0.25$
SRPSO	$N = 30, w_{initial} = 1.05, w_{final} = 0.5, c_1 = c_2 = 1.49445, V_{max} = 0.06078 * Range$
SLPSO	$M = 30$
GBABC	$SN = 30, limit = 100, CR = 0.3$
EABC-BB	$SN = 30, limit = 100, p = 0.1$

other algorithms can be found from the novel search equation Eq.(9) and the parameter adaption strategy (see Eq.(10)). By contrast the mean value of the normal distribution of Eq.(9) with that of Eq.(6), we can found that the ratio of the current solution $x_{i,j}$ is 1/2 in Eq.(6) because there are only two solutions ($x_i$and the global best solution x_{best}), but the ratio of $x_{i,j}$ in Eq.(9) is 1/3 because there are three solutions (x_i, the elite solution x_e and the global best solution x_{best}). That is to say, the EABC-BB is guide by more elite solutions and has stronger exploitation ability. Meanwhile, because different elite solutions have equal chance to be a leader, thus the premature convergence phenomenon in ABC-BB can be effectively avoided and the novel search equation can enable EABC-BB explore different function peaks denoted by different elite solutions. Moreover, the parameter *CR* adaption strategy can also enhance the performance of ABC-BB.

Table 5. The comparison results between EABC-BB and other EAs at *D*=30

Func.	index	sinDE	SRPSO	SLPSO	GBABC	EABC-BB
F01	Mean error	3.07e-53†	3.08e-74†	3.60e-37†	2.63e-49†	4.66E-81
	std dev	8.70E-53	1.30E-75	8.31E-37	5.27e-49	3.29E-80
F02	Mean error	1.72e-38†	8.23e-48‡	1.80e-16†	2.53e-31†	1.69E-41
	std dev	4.00E-38	3.87E-48	3.14E-16	2.69e-31	1.08E-40
F03	Mean error	1.37e+03†	3.59e+00‡	1.38e+03†	4.48e+01‡	1.15E+02
	std dev	4.30E+02	3.44E+00	3.13E+03	7.18E+01	2.16E+02
F04	Mean error	3.50e+00†	4.22e-02‡	1.39e+01†	1.39e-07‡	6.40E-01
	std dev	3.81E+00	3.05E-02	9.32E+01	1.88e-07	9.95E-01
F05	Mean error	5.19e+01†	3.00e+01†	3.50e+01†	3.63e+00†	1.52E+01
	std dev	2.05E+01	3.15E+01	2.60E+01	3.40E+00	4.44E+00
F06	Mean error	0.00e+00≈	0.00e+00≈	2.75e-06†	0.00e+00≈	0.00E+00
	std dev	0.00E+00	0.00E+00	2.25E-06	0.00E+00	0.00E+00
F07	Mean error	3.97e-03†	8.93e-03†	3.34e-02†	9.82e-05†	2.74E-03
	std dev	9.98E-04	2.08E-03	8.66E-03	7.06e-05	3.56E-03
F08	Mean error	4.47e+02†	9.93e+02†	3.95e+02†	3.82e-04†	3.82E-04
	std dev	3.18E+02	2.72E+02	5.57E+02	4.54e-13	6.84E-12
F09	Mean error	1.56e+01†	3.06e+01†	2.02e+01†	0.00e+00≈	0.00E+00
	std dev	3.01E+00	4.52E+00	1.80E+01	0.00E+00	0.00E+00
F10	Mean error	7.99e-15†	3.07e-14†	9.00e-14†	4.44e-16‡	3.39E-15
	std dev	0.00E+00	3.17E-15	8.89E-14	0.00E+00	4.24E-15
F11	Mean error	2.44e-50†	6.89e-03†	3.69e-16†	0.00e+00≈	0.00E+00
	std dev	6.88E-50	1.00E-02	6.96E-14	0.00E+00	0.00E+00
F12	Mean error	1.85e-34‡	2.07e-32†	4.33e-13†	1.57e-32†	6.28E-33
	std dev	5.23E-34	4.63E-02	6.40E-13	5.47e-48	7.77E-33
F13	Meanerror	3.15e+02†	5.49e-32†	2.10e-33†	1.35e-32†	5.99E-34
	std dev	1.26E+02	6.45E-32	3.64E-33	5.47e-48	7.42E-34
F14	Mean error	3.29e+03†	6.13e-13†	1.49e-14†	6.44e-14†	4.11E-14
	std dev	1.88E+03	4.92E-14	3.35E-14	1.93e-14	1.72E-14
F15	Mean error	2.06e+03†	4.42e-04‡	3.31e+03†	2.94e+02†	9.14E-02
	std dev	4.43E+02	5.82E-04	1.85E+03	1.11E+02	2.96E-01
F16	Meanerror	5.18e+01‡	5.37e+06†	1.57e+07†	2.80e+06‡	4.83E+06
	std dev	2.28E+01	4.18E+06	1.14E+07	1.02E+06	4.89E+06
F17	Mean error	3.70e+03†	5.39e+01‡	6.52e+03†	7.38e+03†	3.14E+03
	std dev	4.86E-13	4.22E+01	2.73E+03	1.46E+03	1.14E+03
F18	Mean error	2.11e+01‡	3.39e+03†	7.26e+03†	2.43e+03†	1.86E+03
	std dev	3.10E-02	7.87E+02	1.89E+03	4.72E+02	8.12E+02
F19	Mean error	7.85e+00‡	2.72e+01‡	4.62e+02†	3.64e+01‡	8.19E+01
	std dev	2.50E+00	2.80E+01	3.31E+02	4.94E+01	8.35E+01
F20	Mean error	5.56e+01†	1.87e-02†	2.41e+03†	1.86e-02†	1.61E-02
	std dev	1.22E+01	1.14E-02	3.21E+03	1.63e-02	5.64E-02
F21	Mean error	6.60e+01†	2.09e+01≈	2.05e+01‡	2.09e+01≈	2.09E+01
	std dev	3.32E+01	3.34E-02	1.18E+01	8.62e-02	5.59E-02
F22	Mean error	5.19e+04†	1.65e+01†	5.50e+01†	6.63e-14†	5.22E-14
	std dev	3.66E+04	1.93E+00	1.56E+01	2.12e-14	1.55E-14
F23	Mean error	5.63e+00‡	1.51e+02†	6.81e+01‡	1.28e+02†	1.07E+02
	std dev	6.58E-01	3.60E+01	8.08E+00	1.77E+01	2.13E+01
†/‡/≈	–	17/5/1	15/6/2	21/2/0	14/5/4	–

4.5. Application to Two Real Optimization Problems

This section is about the performance evaluation of EABC-BB algorithm on the two well-known non-linear optimization problems. The explanations of the two problems and the obtained results via EABC-BB are given in the following sub-section in details.

Pressure Vessel Design (PVD) Problem This benchmark engineering problem is an engineering optimization problem. In this problem,the total cost, including a combination of welding cost, material and forming cost, is to be minimized. The welded beam is fixed and designed to support a load. The involved variables are the thickness $Ts(x_1)$, thickness of the head $Th(x_2)$, the inner radius $R(x_3)$, and the length of the cylindrical section of the vessel $L(x_4)$.The problem can be expressed as follows:

$$x = (x_1, x_2, x_3, x_4)^T := (T_s, T_h, R, L)^T \tag{12}$$

$$minf(x) = 0.6224x_1x_2x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{13}$$

$$g_1(x) = 0.019x_3 - x_1 \leq 0 \tag{14}$$

$$g_2(x) = 0.00954x_3 - x_2 \leq 0 \tag{15}$$

$$g_3(x) = 1,296,000 - \pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 \leq 0 \tag{16}$$

$$g_4(x) = x_4 - 240 \leq 0 \tag{17}$$

$$x \in R^4 : (0, 0, 10, 10)^T \leq x \leq (99, 99, 200, 200)^T \tag{18}$$

As can be seen, the PVD problem is a constraint problem with four constraints, i.e., g_1, g_2, g_3 and g_4. As the ABC algorithm originally developed for unconstrained continuous global optimization problems, it also requires a constraint handling method while dealing with the constrained global optimization problems. This paper adopts the penalty function method introduced in literature [5] to transfer the constraint problem to a unconstrained problem.

The parameters setting are the same as literature [5], i.e.: the maximum function evaluations (*Max_FEs*) is set at 500,000,the population size (*SN*) is set at 100.Finally, the results were obtained after running the EABC-BB algorithm 20 times for each problem. The experimental results are given in Table 6. From Table 6, it is clear that EABC-BB is the best algorithms to solve the PVD problem, since the EABC-BB algorithm has the ability to produce better results than the previous literature with all of the aforementioned constraint handling methods. It's worthy to note that DGABC is another DE and ABC hybrid algorithm, which is second to EABC-BB in PVD problem. Similar to GBABC, in DGABC a chaotic opposition-based population initialization method is employed, but to make a fair comparison, all algorithms except for GBABC adopt random initialization strategy.

Because GBABC has compared with some state-of-the-art EAs [48], the comparison with GBABC can be looked on as the comparison with these EAs indirectly.

According to Table 6, the best solution obtained by EAB-BB is *x*=[0.77817354 0.3847-4404 40.31987228 199.99647520], its constraints are g_1(*x*)=-4.996000058099526E-09, g_2(*x*)=-9.245844880001464E-05, g_3(*x*)=-0.024784596171230 and g_4(*x*)= -40.0035248-00000008. We can see that the solution *x* satisfy all constraints.

Table 6. Experimental results for the pressure vessel design (PVD) problem. The experimental results of the first nine algorithms are taken from [5]

Literature/ methods	x_1(best)	x_2(best)	x_3(best)	x_4(best)	f(best)	f(worst)	f(average)	std.dev.
Baykasoglu [5]	0.8125	0.4375	42.0975	176.648	6059.83	6823.602	6149.727	210.70
Coello [12]	0.8125	0.4375	40.3239	200.000	6288.74	-	-	-
Coello and Mezura-Montes [13]	0.8125	0.4375	40.0973	176.654	6059.94	-	-	-
Mezura-Montes et al. [39]	0.8125	0.4375	42.0983	176.637	6059.71	6846.628	6355.343	256.00
Mezura and Coello [38]	0.8125	0.4375	42.0984	176.636	6059.71	-	6379.938	210.00
Akhtar et al. [3]	0.8125	0.4375	41.9768	182.284	6171.00	-	-	-
He et al. [28]	0.8125	0.4375	42.0984	176.636	6059.71	-	-	-
Parsopoulos and Vrahatis [40]	-	-	-	-	6154.71	9387.770	8016.370	745.80
Aguirre et al. [1]	0.8125	0.4375	42.0984	176.636	6059.71	-	6071.013	15.10
He and Wang [26]	0.8125	0.4375	42.0912	176.746	6061.07	6363.804	6147.133	86.45
He and Wang [27]	0.8125	0.4375	42.0984	176.636	6059.71	6288.677	6099.932	86.20
Cagnina et al. [8]	0.8125	0.4375	42.0984	176.636	6059.71	-	6092.049	12.17
Maruta et al. [37]	0.8125	0.4375	42.0984	176.636	6059.71	7332.841	6358.156	372.7
Kim et al. [33]	0.8125	0.4375	42.0984	176.636	6059.71	6060.074	6059.727	0.065
Chun et al. [11]	0.8125	0.4375	42.0984	176.636	6059.71	6090.526	6060.330	4.357
Akay and Karaboga [2]	-	-	-	-	6059.71	-	6245.308	205.00
Brajevic and Tuba [7]	0.8125	0.4375	42.0984	176.636	6059.71	-	6192.116	204.00
Gandomi et al. [22]	0.8125	0.4375	42.0984	176.636	6059.71	6318.950	6179.130	137.20
Baykasoglu and Ozsoydan [6]	0.8125	0.4375	42.0984	176.636	6059.71	6090.526	6064.336	11.28
WSA [5]	0.7865	0.3934	40.7526	194.78	5929.62	5984.756	5958.410	15.07
SLPSO [10]	0.7884	0.3897	40.8542	192.688	5903.20	6425.556	6139.762	215.4
SRPSO [45]	0.9638	0.4764	49.9397	98.8235	6284.45	6290.601	6287.521	2.326
sinDE [19]	0.8713	0.4327	44.7686	147.183	6144.05	6309.273	6277.016	68.69
MGABC [15]	0.7788	0.3849	40.3404	199.741	5888.75	5941.013	5893.319	21.25
BCABC [23]	0.7924	0.3917	41.0587	189.959	5910.16	6051.407	5975.132	37.45
ABC-BB [48]	0.7828	0.3881	40.5345	197.198	5903.80	5972.571	5941.527	37.83
GBABC [48]	0.7834	0.3869	40.5319	197.07	5901.55	5975.849	5941.258	38.43
ABC-elite [14]	0.7796	0.3851	40.3643	199.385	5891.19	5933.783	5900.800	16.72
DGABC[24]	0.779	0.3851	40.3613	199.421	5887.12	5939.237	5919.170	24.78
EABC-BB	0.7781	0.3847	40.3198	199.996	**5885.34**	5914.163	**5888.892**	17.34

The Frequency Modulated (FM) Sound Wave Synthesis Problem The FM sound wave synthesis plays an important role in modern music systems. It provides a simple and efficient method creating complex sound timbres. This subsection applies the proposed EABC-BB method to optimize the parameters of an FM synthesizer. The details of the problems are described as follows.

The FM sound synthesis aims to optimize the parameter of an FM synthesizer with a D-dimensional vector $\boldsymbol{x}$. In this paper, we only consider the case of D=6 by the suggestions of the literature[48]. The objective of this problem is to optimize a six-dimensional vector x=$\{a_1,w_1,a_2,w_2,a_3,w_3\}$ of the sound wave given in Eq.(19). The problem is to generate a sound similar to the object sound. The formulas for the estimated sound wave and the target sound are given as follows:

$$\mathrm{y}(t) = a_1 \times \sin(w_1 t \cdot \theta + a_2 \times \sin(w_2 t \cdot \theta + a_3 \times \sin(w_3 t \cdot \theta))) \tag{19}$$

$$\mathrm{y}_0(t) = 1.0 \times \sin(5.0t \cdot \theta - 1.5 \times \sin(4.8t \cdot \theta + 2.0 \times \sin(4.9t \cdot \theta))) \tag{20}$$

where $\theta = 2\pi/100, x_i \in [-6.4, 6.35]$.

The goal of this problem is to minimize the sum of squared errors between the estimated sound and the target sound, as given by Eq.(20).This problem is a highly complex multimodal one having strong epistasis, with minimum value $f(x)$=0.

Table 7. Comparison of EABC-BB with different methods for the FM sound wave synthesis problem

Algorithms	Mean	Std dev	Best	Worst
ABC	6.78	6.52	1.76	11.83
SLPSO	6.91	4.42	4.12	12.78
SRPSO	6.74	5.96	**0**	13.53
sinDE	6.49	**0.58**	5.92	7.89
GABC	4.88	5.23	0.036	11.82
MGABC	5.58	5.92	7.72E-05	14.36
BCABC	4.54	4.49	3.12E-16	12.23
ABC-BB	2.85	2.68	0.27	11.83
GBABC	2.23	3.52	0.22	10.28
ABC-elite	0.77	2.14	1.62E-12	8.94
DGABC	0.26	3.12	**0**	9.72
EABC-BB	**0.072**	1.51	**0**	5.82

$$f = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \tag{21}$$

In the experiment, EABC-BB as well as the other 11 EAs are applied to solve this problem. For all algorithms, the parameters are set according to suggestions of the literature [48], i.e., *SN*=30, *max_FEs*=200,000. Each algorithm is run 30 times, and the mean and standard deviation values are reported in Table 7. From the results, it is clear that EABC-BB performs better than other 11 EAs in terms of the quality of the final solutions. Similar to the PVD problem, in order to make a fair comparison, we use DGABC without chaotic opposition strategy. DGABC also shows relatively good performance in FM problem, which shows again that the DE and ABC hybrid algorithm is an effective method to improve the performance of ABC and DE.

5. Conclusions

This paper presents an enhanced ABC-BB (EABCBB for short), which aims to overcome the premature convergence problem of ABC-BB. In EABC-BB, the solution search equation of onlooker bees is replaced with trigonometry based search equation. In this equation, the candidate is generated around the current solution, the global best solution and a randomly selected elite solution. By doing so, more function peaks represented by different elite solutions can be explored and the premature convergence problem of ABC-BB thereby can be effectively decreased. The reason is that the trigonometry method can keep up the population diversity and improve the exploration capability without lowering the exploitation ability of ABC-BB. Meanwhile, in order to overcome the crossover (*CR*) parameter sensitive problem in ABC-BB, a simple self-adaptive strategy is proposed to dynamically update *CR* according to the search experiences of successful *CR* probabilities. A comprehensive set of experiments is conducted on 23 benchmark functions and two well-known real-world engineering optimization problems to verify the performance of our proposed approach EABC-BB. Some other well-known ABCs and state-of-the-art EAs, such as newly developed PSO variants and DE variants, are used for comparison. The experimental results demonstrate that our approach achieve better performance in term of solution accuracy and convergence speed, thus we may reasonably conclude that the proposed EABC-BB is a new competitive algorithm. It is worthy note that another DE and ABC hybrid algorithm DGABC also show excellent performance in aforementioned two engineering optimization problems, which demonstrates again that the DE and ABC hybrid algorithm is a promising research direction if the algorithm is well designed.

Acknowledgements This research is funded by the Science and Technology Plan Project of Chaozhou under Grant No. 2018GY45, National Natural Science Foundation of China (No. 61672338) and National Key Research and Development Program of China (No. 2016YFD0702100), National Natural Science Area Foundation Project of China (No. 61863011) and Guangxi Natural Science Youth Foundation Project of China (No. 2015-GXNSFBA139264).

References

1. Aguirre, A.H., Zavala, A.M., Diharce, E.V., Rionda, S.B.: COPSO: constrained optimization via PSO algorithm. Center for Research in Mathematics (2007)
2. Akay, B., Karaboga, D.: Artificial bee colony algorithm for large-scale problems and engineering design optimization. Journal of Intelligent Manufacturing 23(4), 1001–1014 (2012)
3. Akhtar, S.and Tai, K., Ray, T.: A socio-behavioural simulation model of engineering design optimization. Engineering Optimization 34(3), 341–354 (2002)
4. Aydogdu, I., Akin, A., Saka, M.P.: Design optimization of real world steel space frames using artificial bee colony algorithm with levy flight distribution. Advances in Engineering Software 92(1), 1–14 (2016)
5. Baykasoğlu, A., Akpinar, S.: Weighted superposition attraction (wsa): a swarm intelligence algorithm for optimization problems–part 2: constrained optimization. Applied Soft Computing 37(4), 396–415 (2015)
6. Baykasoğlu, A., Ozsoydan, F.B.: Adaptive firefly algorithm with chaos for mechanical design optimization problems. Applied soft computing 36(11), 152–164 (2015)
7. Brajevic, I., Tuba, M.: An upgraded artificial bee colony (abc) algorithm for constrained optimization problems. Journal of Intelligent Manufacturing 24(4), 729–740 (2013)
8. Cagnina, L.C., Esquivel, S.C., Coello, C.A.C.: Solving engineering optimization problems with the simple constrained particle swarm optimizer. Informatica 32(3), 319–326 (2008)
9. Chen, J., Yu, W., Tian, J.: Image contrast enhancement using an artificial bee colony algorithm. Swarm and Evolutionary Computation 38(2), 287–294 (2017)
10. Cheng, R., Jin, Y.: A social learning particle swarm optimization algorithm for scalable optimization. Information Sciences 291(3), 43–60 (2015)
11. Chun, S., Kim, Y.T., Kim, T.H.: A diversity-enhanced constrained particle swarm optimizer for mixed integer-discrete-continuous engineering design problems. Advances in Mechanical Engineering 5(1), 1–11 (2015)
12. Coello, C.A.C.: Use of a self-adaptive penalty approach for engineering optimization problems. Computers in Industry 41(2), 113–127 (2000)
13. Coello, C.A.C., Mezura-Montes, E.: Use of dominance-based tournament selection to handle constraints in genetic algorithms. Intelligent Engineering Systems through Artificial Neural Networks 11(1), 177–182 (2001)

14. Cui, L., Li, G., Lin, Q.: A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation. information sciences. Information Sciences 367(3), 1012–1044 (2016)
15. Cui, L., Zhang, K., Li, G.: Modified gbest-guided artificial bee colony algorithm with new probability mode. Soft Computing 22(1), 1–27 (2017)
16. Deng, Y.H.: What is the future of disk drives, death or rebirth? ACM Computing Surveys 43(3), 1–27 (2011)
17. Derrac, J., S, G., Molina, D.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 11(1), 3–18 (2011)
18. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics 26(1), 29–41 (1996)
19. Draa, A., Bouzoubia, S., Boukhalfa, I.: A sinusoidal differential evolution algorithm for numerical optimisation. Applied Soft Computing 27(2), 99–126 (2015)
20. Du, Z., Han, D., Li, K.C.: Improving the performance of feature selection and data clustering with novel global search and elite-guided artificial bee colony algorithm. The Journal of Supercomputing (2019), https://doi.org/10.1007/s11227-019-02786-w
21. Du, Z., Han, D., Liu, G.: An improved artificial bee colony algorithm with elite-guided search equations. Computer Science and Information Systems 14(3), 751–767 (2017)
22. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems,. Engineering with Computers 29(1), 17–35 (2013)
23. Gao, W., Chan, F., L., H.: Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood. Information Sciences 316(3), 180–200 (2015)
24. Gao, W., L., H., J., W.: Enhanced artificial bee colony algorithm through differential evolution. Applied Soft Computing 48(11), 137–150 (2016)
25. Gao, W., Liu, S., Huang, L.: A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. IEEE Transactions on Cybernetics 43(3), 1011–1024 (2013)
26. He, Q., Wang, L.: An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence 20(1), 89–99 (2007)
27. He, Q., Wang, L.: A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Applied Mathematics and Computation 186(2), 1407–1422 (2007)
28. He, S., Prempain, E., Wu, Q.H.: An improved particle swarm optimizer for mechanical design optimization problem. Engineering Optimization 36(5), 585–605 (2004)
29. Jafrasteh, B., Fathianpour, N.: A hybrid simultaneous perturbation artificial bee colony and back-propagation algorithm for training a local linear radial basis neural network on ore grade estimation. Neurocomputing 38(2), 287–294 (2017)
30. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. APPLIED MATHEMATICS AND COMPUTATION 214(1), 108–132 (2009)
31. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. Advances in Engineering Software 39(3), 459–471 (2007)
32. Karaboga, D., Ozturk, C.: A novel clustering approach: artificial bee colony (abc) algorithm. Applied Soft Computing 11(2), 652–57 (2011)
33. Kim, T.H., Maruta, I., Sugie, T.: A simple and efficient constrained particle swarmoptimization and its application to engineering design problems. Journal of Mechanical Engineering Science 224(2), 389–400 (2010)
34. Kumar, Y., Sahoo, G.: A two-step artificial bee colony algorithm for clustering. Neural Computing and Applications 28(3), 537–551 (2017)

35. Li, Y., Z., Z., Lin, S.: Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems. Information Sciences 293(3), 370–382 (2015)
36. Lozano, M., García-Martínez, C.: Optimizing network attacks by artificial bee colony. Information Sciences 377(1), 30–50 (2017)
37. Maruta, I., Kim, T.H., Sugie, T.: Fixed-structure h∞ controller synthesis: a metaheuristic approach using simple constrained particle swarm optimization. Automatica 45(3), 553–559 (2009)
38. Mezura-Montes, E., Coello, C.A.C.: Useful infeasible solutions in engineering optimization with evolutionary algorithms. In: Mexican International Conference on Artificial Intelligence. pp. 652–662. Springer, Verlag Berlin Heidelberg (2005)
39. Mezura-Montes, E., Coello, C.C., Landa-Becerra, R.: Engineering optimization using a simple evolutionary algorithm. In: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence. pp. 149–156. IEEE, Sacramento, CA, USA (2003)
40. Parsopoulos, K.E., Vrahatis, M.N.: Unified particle swarm optimization for solving constrained engineering optimization problems. In: International Conference on Natural Computation. pp. 582–591. Springer, Berlin, Heidelberg (2005)
41. Qin, A.K., Huang, V.L.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE transactions on Evolutionary Computation 13(2), 398–417 (2009)
42. R., E., J., K.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science. pp. 39–43. Ieee, Nagoya, Japan (1995)
43. Storm, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
44. Suganthan, P.N., Hansen, N., J., L.J.: Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical Report, Singapore. Nanyang Technological University, Nanyang Technological University (2005)
45. Tanweer, M.R., Suresh, S., Sundararajan, N.: Self regulating particle swarm optimization algorithm. Information Sciences 294(3), 182–202 (2015)
46. Wei, Y.H., Xu, C., Hu, Q.Y.: Transformation of optimization problems in revenue management, queuing system, and supply chain management. International Journal of Production Economics 146(2), 588–597 (2013)
47. Zhang, J., Sanderson, A.C.: Jade: adaptive differential evolution with optional external archive. IEEE Transactions on evolutionary computation 13(5), 945–958 (2009)
48. Zhou, X., Wu, Z., Wang, H.: Gaussian bare-bones artificial bee colony algorithm. Soft Computing 20(3), 907–924 (2016)
49. Zhu, G., S., K.: Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied Mathematics and Computation 217(7), 3166–3173 (2010)

Zhenxin Du received the B.S. degree in computer science from Zhejiang Sci-tech University in 2011. He is is a lecturer of computer science and engineering at Hanshan Normal University. His main research interests include computation intelligence, data mining and cloud computing.

Keyin Chen (corresponding author) received the Ph.D degree from South China Agricultural University. He is currently a post doctor of agricultural electrification and automation engineering at Nanjing Institute of Agricultural Mechanization and also at the same time is lecturer of electrical engineering engineering at Hezhou University. His research interests include agricultural robot, machine vision and bionic intelligence technology.

Received: September 23, 2018; Accepted: August 27, 2019.