An improved artificial bee colony algorithm based on Bayesian estimation

Chunfeng Wang 1,2 · Pengpeng Shang 2 · Peiping Shen 3

Received: 13 September 2021 / Accepted: 8 April 2022 / Published online: 29 April 2022
© The Author(s) 2022

Abstract

Artificial bee colony (ABC) algorithm was proposed by mimicking the cooperative foraging behaviors of bees. As a member of swarm intelligence algorithms, ABC has some advantages in handling optimization problems. However, it has the exploration capacity over the exploitation capacity, which may lead to slow convergence speed and lower solution accuracy. Hence, to enhance the performance of the algorithm, a novel ABC based on Bayesian estimation (BEABC) is presented in this paper. First, instead of using the fitness ratio, the selection probability in ABC is replaced with a new probability calculated by Bayesian estimation. Second, to help the bees adopt more useful information during updating new food sources, a directional guidance mechanism is designed for onlooker bees and scout bees. Finally, the comprehensive performance of BEABC is evaluated by 24 single-objective test functions. The numerical experiment results indicate that BEABC dominates its peers over most test functions, and the significant statistics show that the significant excellence rate of BEABC is 76% in the overall comparison. In addition, to further test the performance of BEABC, seven multi-objective problems and two real-word optimization problems are solved. The comparison results show that BEABC can achieve better results than other EA competitors.

Keywords  Swarm intelligence · Artificial bee colony · Bayesian estimation · Directional guidance strategy

Introduction

Over the past years, inspired by animals and social behaviors in nature, researchers have proposed different evolutionary algorithms (EAs) to deal with some complex optimization problems. As a branch of EAs, swarm intelligence has attracted extensive interest of many scholars. Until now, many excellent algorithms have been developed, such as firefly algorithm (FA) [1], genetic algorithm (GA) [2], differential evolution (DE) [3], artificial bee colony algorithm (ABC) [4], particle swarm algorithm (PSO) [5], and so on. These methods display good ability in solving optimization problems.

Motivated by the cooperative foraging behaviors of bees, Karaboga first proposed ABC algorithm in 2005 [6]. Compared with other algorithms, ABC has the features of simple structure, fewer control parameters and more powerful search capability. Thus, it has been widely studied by many scholars, and has been used to deal with some complex problems. Moreover, in the fields of real-word optimization problems, such as electrical engineering [7], engineering physics [8], path planning [9], feature classification [10,11], vehicular networks [12], ABC has been widely adopted.

Related work

Although ABC could achieve good performance in many areas, it does a poor job in maintaining a balance between global search capacity and local search capacity. In other words, it may be caught in a local optimal position in a high likelihood and get a low precision solution, as well as slow convergence speed. Therefore, to overcome these shortcomings, it is necessary to develop potential ABC algorithms. To
this end, there are many terrific ABC variants (ABCs) have been presented from different aspects in recent years.

According to the strategies used in literatures, they can be grouped into three categories: parameter-based strategy, probability-based strategy and movement-equation-based strategy. The brief reviews are as follows:

(1) Parameter-based strategy. As pointed out in Ref. [13], the convergence of EAs can be adjusted well by introducing reasonable control parameters. So, to adjust the parameters in EAs, some scholars have proposed many adjustable strategies. For example, by analyzing the control parameters, Akay and Karaboga [14] discussed the influences of parameters in ABC. In Ref. [15], based on a new control parameter modification rate (MR), a modified vision ABC was proposed, which could achieve the purpose of controlling frequency of perturbation. According to the movement information of the previous solution, Kiran et al. [16] designed a control parameter \( d \) to guide the bees in a directional movement. For solving binary problems, Durgut and Aydin [17] proposed an adaptive hybrid approach to devise ABC algorithm, which includes multiple operators selected adaptively strategy and credit assignment mechanism.

(2) Probability-based strategy. As we know, the onlooker bees select individuals according to the probability defined by the fitness ratio in ABC. But this probability selection strategy may cause great selection pressure in the late stage. To overcome this deficiency, based on knowledge fusion, Wang et al. [18] presented a ABC variant KFABC, which wisely utilizes three kinds of knowledge strategies to search new solutions by the selection rules. In Ref. [19], Cui et al. proposed a novel ABC named DFSABC_elite, which designs two modified movement equations and a depth-first search framework (DFS). This method implemented the selection of the solutions by designing a new parameter mechanism. For constrained optimization problems, Liu et al. [20] proposed a ABC variant DPLABC, which includes dynamic penalty method, Levy flight with logistic map, further search mechanism and new boundary handling mechanism. A novel ABC named ABC-AHC has proposed by Chu et al. [21]. This study designed a new selection probability model, and presented an adaptive heterogeneous competition augmented ABC algorithm. In dealing with multi-objective problem, Chaudhuri and Sahu [22] proposed a multi-objective ABC-based feature weighting technique for Naïve Bayes. In the article, a mutation probability was designed to control the selection of movement equations in the phases of employed bee and onlooker bee, and the selection probability was calculated on the bases of the sorting number each food source.

(3) Movement-equation-based strategy. In EAs, how to better tradeoff the capacities of global search and local search is always concerned. Many studies found that the purpose can be achieved by modified the movement equation. Zhou et al. [23] proposed an enhancing ABC with multi-elite guidance (MGABC), which chooses elite solutions to construct elite groups and designs two modified search equations using the elite groups information. Inspired by PSO algorithm, Zhu et al. [24] presented a Gbest-guided ABC algorithm (GABC). In GABC, using the contemporary global optimum solution, the new modified update equation was shown. In terms of the results of experiments, GABC could be superior to the conventional ABC up to a point. By utilizing the best individual of the previous, Luo et al. [25] provided a new ABC algorithm named COABC. In COABC, a novel update equation was designed in the phase of onlooker bee. Aiming at improving the performance of ABC, Zhou et al. [26] proposed a modified search strategy by using the current global optimal solution and designed a new Gaussian perturbation with evolutionary rate. Using the best neighbor-guided movement strategy, Peng et al. [27] presented an algorithm named NABC. Based on factor library and dynamic search balance, Yu et al. [28] proposed a hybrid, fast, and enhanced ABC, called HFEABC.

Table 1 summarizes some representative ABC variants, including their advantages, disadvantages and classification.

Motivation

Since ABC has excellent performance in solving optimization problems, it has been widely studied and applied in many fields [29,30]. When solving some complex optimization problems, it still has some shortcomings. For example, it has great selection pressure in the late stage, which may be caused by the probability mechanism on the basis of the fitness ratio [31]. To alleviate this problem, two new rules were designed in Refs. [18,19], which use previous experience to implement different strategies. However, pure empirical guidance may make the algorithm fall into oscillation, thus it is meaningful to retain the randomness of the probability mechanism. To reduce the blindness of choosing food sources to a certain extent, a novel probability is designed in this paper.

In addition, as we know, the movement equation plays a key role in maintaining a balance between global search and local search. To enhance the exploitation ability of ABC, many excellent ABCs focused on modifying the movement equations or designing good learning strategies have been proposed. In these methods, some useful information was used, such as the information of the current best solution or the elite solutions [23,32]. However, only using high-quality solutions without controlling the search range may cause the algorithm to exploit and fall into local optimal position. Therefore, reasonable neighbor selection and parameter design are considered and discussed in our approach.

Based on the above considerations, to compensate the shortcomings of conventional ABC, a novel ABC algorithm
based on Bayesian estimation [33] (BEABC) is proposed in this paper. The major contributions can be listed below:

1. A new selection probability is designed. Based on Bayesian estimation, the posterior probability of the food sources are calculated, which is used to select those food sources that can produce better offsprings in a high likelihood.

2. A novel directional guidance strategy is presented. In onlooker bee and scout bee phases, using the location information of neighbors and the current optimal solution, two solution search equations are designed, which can guide the bees to search in the right direction.

3. Two dynamically and adaptively adjusting parameters MR and $\lambda$ are introduced. By automatically controlling the disturbance frequency and adjusting the search region, respectively, the algorithm can keep balance between exploration and exploitation capacities.

The rest of the paper is outlined as follows: “Artificial bee colony algorithm (ABC)” introduces the conventional ABC framework. In “ABC based on Bayesian estimation (BEABC)”, the motivation and the specific processes of BEABC are given. “Numerical and engineering Experiments” presents and analyzes the experimental results of BEABC and other excellent EAs. The effectiveness and superiority of BEABC are verified by these experiments. In “Conclusion”, the conclusion is drawn. Note that all the abbreviations involved in this paper are explained in the appendix.

Table 1: Summary of related work

| References          | Year | Advantages                                           | Disadvantages                        | Classification       |
|---------------------|------|------------------------------------------------------|--------------------------------------|----------------------|
| Zhu et al. [24]     | 2010 | Global best solution is added into movement equation | The good neighbor information is not considered | Movement-equation-based |
| Akay and Karaboga [15] | 2012 | The new parameter modification rate is introduced   | More effective MR is not considered  | Parameter-based      |
| Li et al. [9]       | 2014 | Fully utilize the convergence status within the iteration system | The convergence rate is lower | Path planning        |
| Kiran et al. [16]   | 2015 | Control evolutionary strategy selection             | May have high computational complexity | Parameter-based      |
| Cui et al. [19]     | 2016 | A depth-first search framework is combined with ABC  | DFS alone has limited performance improvement | Probability-based    |
| Durgut et al. [17]  | 2018 | Adaptive operator selection and credit assignment rule are introduced accordingly | More effective operator selection schemes need to be developed | Parameter-based      |
| Yu et al. [28]      | 2018 | Combine different factors for different problems    | Premature is very likely              | Movement-equation-based |
| Chu et al. [21]     | 2020 | New probability model with the rate of successful searches and linear weight is presented | The pertinence of probability selection is not considered | Probability-based    |
| Thilak et al. [12]  | 2021 | Search strategies based on differential evolution and the method integrated chaotic and opposition learning are introduced | Value neighbor information is not considered | Movement-equation-based |
Artificial bee colony algorithm (ABC)

ABC mainly contains four phases: initialization phase, employed bee phase, onlooker bee phase and scout bee phase. Different kinds of bees can change their roles iteratively until the termination condition is met. Note that there is an associated counter for each food source. If one food source is not improved, the increment of its corresponding counter is 1; otherwise the counter resets to 0. If the quality of a solution has not been improved more than limit (preset parameter), the employed bee would be transformed into a scout bee.

Initialization phase

Let \( X = \{x_1, x_2, \ldots, x_{SN}\} \) be the initialization population, which are generated randomly in the entire space. The food source \( x_{ij} \) of the initial population is determined by the following equation:

\[
x_{ij} = x_{ij}^{\min} + r \times (x_{ij}^{\max} - x_{ij}^{\min}),
\]

where \( i = 1, 2, \ldots, SN; j = 1, 2, \ldots, D \).

Employed bee phase

In this phase, to generate new solutions \( v_i \), the employed bee executes a random neighborhood search around \( x_i \) using the following equation:

\[
v_{ij} = x_{ij} + \varphi_{ij} \times (x_{ij} - x_{kj}),
\]

where \( k \) and \( j \) are randomly selected from \( \{1, 2, \ldots, SN\} \) and \( \{1, 2, \ldots, D\} \), respectively, and \( k \neq i \). In (2), only one dimension of \( x_i \) is modified. If \( v_{ij} \) is more excellent than \( x_{ij} \), \( x_{ij} \) will be replaced by \( v_{ij} \), and the counter for \( x_i \) will be reset to 0; otherwise \( x_i \) keeps invariant, and its counter will go up by 1.

Onlooker bee phase

In this phase, the fitness ratio will be used as the selection probability to choose food sources. The probability can be calculated by (3):

\[
P_i = \frac{\text{Fit}_i}{\sum_{i=1}^{SN} \text{Fit}_i}.
\]

The fitness value of food source \( x_i \) is defined by (4):

\[
\text{Fit}_i = \begin{cases} 
\frac{1}{1 + f(x_i)} & \text{if } f(x_i) \geq 0, \\
1 + |f(x_i)| & \text{else.}
\end{cases}
\]

From (3) and (4), it is easy to see that a food source with a larger fitness value has a higher possibility to be selected by the onlooker bees. Similar to the employed bee phase, the new solutions are determined by (2), and the associated counters will be changed accordingly.

Scout bee phase

In this phase, if the counter of a certain solution is bigger than limit, the solution will be considered as a terrible solution, and the corresponding employed bee will be transformed into a scout bee. The location of the scout bee will be randomly generated in the entire space according to (1).

By the above description of ABC, the pseudo-code is displayed briefly in Algorithm 1:

ABC based on Bayesian estimation (BEABC)

In this section, we propose a new ABC variant based on Bayesian estimation, called BEABC, which tries to address the following issues: (1) how to maximize the probability of those better solutions selected by roulette; (2) how to reasonably design movement equations to enhance the exploitation capability; (3) how to keep balance between the exploration ability and the exploitation ability. The detailed of BEABC is given in the following subsections.

Selection probability based on Bayesian estimation

In ABC, to generate new offsprings, the bees select the food sources by (3), which can maintain population diversity to a certain extent. But it may face selection pressure and premature convergence, especially at the late stage. To
overcome such shortcomings, two probability mechanisms were presented [20,21]. However, they were lack of pertinence in the solutions with good quality. So, how to choose better solutions with high probability deserves to be studied. Combining mathematical knowledge, on the foundation of improving the quality of a solution, the probability that the solution is selected is a problem of posterior conditional probability. Thus, Bayesian estimation is a suitable way to calculate the probability. According to Bayesian estimation, if there are events $B_i$ satisfy $\bigcap_{i=1}^{n} B_i = \Omega$, $B_i \cap B_j = \emptyset$, $i \neq j$, $P(B_i) > 0$, then $B_1, B_2, \ldots, B_n$ can be seen as an exhaustive sequence of events. Given the occurrence of event $A$, the conditional probability $P(B_i | A)$ can be calculated by (5):

$$P(B_i | A) = \frac{P(B_i)P(A | B_i)}{\sum_{j=1}^{n} P(A | B_j)P(B_j)}, \quad (5)$$

When it is extended to ABC, without loss of generality, let $\{x_1, x_2, \ldots, x_{SN}\}$ be the exhaustive sequence of events, and $A$ be the event that the food source with a better offspring is selected. Assume that the solutions are finite and sufficient. According to the law of large numbers, the probability $P(B = x_i)$ and the conditional probability $P(A | B = x_i)$ can be given as below:

$$P(B = x_i) = \frac{1}{SN}, \quad (6)$$

$$P(A | B = x_i) = \frac{I(A, B = x_i)}{I(B = x_i)}. \quad (7)$$

Substituting (6) and (7) into (5), the posterior probability $P(B = x_i | A)$ can be calculated. It is clear that this probability helps the algorithm choose the better solutions with high quality, which may improve the convergence rate of the algorithm.

In addition, noted that the denominator of the Bayesian equation (5) is a constant, therefore, to reduce the amount of calculation, this paper omits the denominator in the computational process.

**Directional guidance mechanism**

As pointed out above, in ABC, the original movement equation performs very well in exploration but weakly in exploitation. To adjust the global and local search capacities, inspired by different swarm intelligent algorithms, scholars have proposed some new movement equations [5,34]. By (2), we know that, the purpose of randomly selecting neighbors is to maintain population diversity. However, this moving equation has great blindness, and lacks directional guidance. Thus, to reduce the blindness, a directional guidance mechanism is proposed as follows. Let $x_n$ be the neighbor selected randomly for $x_i$. Consider two cases:

**Case 1:** $f(x_n) < f(x_i)$. Under this case, the new solution $v_i$ is produced around $x_i$ as below:

$$v_{ij} = \begin{cases} (1-\lambda) \times x_{ij} - \lambda \times \psi_{ij} \times (x_i - x_{nj}), & \text{if } x_{ij} > x_{nj} \\ (1-\lambda) \times x_{ij} + \lambda \times \psi_{ij} \times (x_n - x_{ij}), & \text{if } x_{ij} < x_{nj}. \end{cases} \quad (8)$$

In this case, the location of $x_n$ may move closer to the global best solution than $x_i$, so it makes sense for $x_i$ to move towards $x_n$. From (8), we can see that it may ensure $x_i$ to move towards $x_n$ no matter where $x_i$ lies in. To help the readers better understand, the following graph is used to show this process:

In Fig. 1, it can be seen that, when $x_{ij} < x_{nj}$, the location of $x_{nj}$ has a great chance to appear on the left of the high-quality solution, thus $x_{ij}$ moves towards $x_{nj}$ may be more closer to the global best solution. In the other case, $x_{ij} > x_{nj}$ means that the location of $x_{ij}$ is on the right of $x_{nj}$, thus it makes more sense for $x_{ij}$ to move left, which may generate superior candidate solution. In a word, under the guidance of its neighbor $x_n$, $x_i$ always moves towards the global best solution.

**Case 2:** $f(x_n) > f(x_i)$. This case means that the neighbor $x_n$ is worse than $x_i$. For this case, by adopting the information of the current global optimal solution, a modified movement equation is presented below:

$$v_{ij} = \begin{cases} (1-\lambda) \times x_{ij} + \lambda \times \varphi_{ij} \times (x_{gj} - x_{ij}), & \text{if } \text{rand} < MR, \\ x_{ij}, & \text{else}. \end{cases} \quad (9)$$

From (9), the frequency of perturbation is controlled by $MR$, when $\text{rand}$ is less than $MR$, the candidate solution $v_{ij}$ is generated around $x_{gj}$, which guarantees that the new solution has relatively high quality; otherwise, $v_{ij}$ is equal to $x_{ij}$.

In scout bee phase, the task of the bee is to discard the obsolete solution and regenerate new one in the whole solution space. From ABC, it can be seen that, though the new food source generated by (1) can maintain the population diversity, it may have blindness to some extent and reduce the convergence rate. Based on the above analysis, if the
information of the current global best solution is adopted, a better food source may be obtained. Thus, to generate a better solution, the following equation is presented:

\[
x_{ij} = (1 - \lambda) \times x_{gj} + \lambda \times \varphi_{ij} \times (x_{gj} - x_{ij}). \quad (10)
\]

From (10), the new food source \(x_i\) is generated around \(x_g\). This strategy may increase the convergence rate of BEABC.

**Parameters design**

In Ref. [15], MR was proposed first and was set to a fixed constant. Generally, linearly varying parameters have some advantages over constant parameters in adjusting the algorithm [35]. Furthermore, for EAs, a nonlinear control parameter updating rule was proposed [36], which has a better performance than linearly varying parameters. In this paper, to strengthen the performance of BEABC, a new way of adaptively varying parameter MR is proposed, which is given as follows:

\[
MR = MR_{\text{max}} \times \exp \left( \frac{-Dt}{\text{MaxDt}} \right). \quad (11)
\]

From (11), MR decreases exponentially as the number of iterations increases, which means that the candidate solutions have more chances of being generated near the current global optimal solution at an early stage. This may improve the convergence rate to some extent. In the late stage of the algorithm, \(v_i\) will be generated around \(x_i\), which can enhance the local search ability.

Meanwhile, it should be noted that \(\lambda\) plays a key role in adjusting the search region at (8) and (9). In our paper, \(\lambda\) is introduced as follows:

\[
\lambda = \frac{\text{MaxDt} - Dt + 1}{\text{MaxDt}}. \quad (12)
\]

As shown in (12), it pays more weight on the second item of the movement equations (8) and (9) at the beginning, which is more likely to strengthen exploration capacity of BEABC. As the algorithm goes on, \(\lambda\) becomes smaller and smaller, so the search process will focus on the first item of the movement equations, which means that the exploitation ability of BEABC are enhanced.

On the basis of the description above, the detailed pseudo-code of BEABC is shown in Algorithm 2. The source code can be publicly available at https://www.mathworks.com/matlabcentral/fileexchange/104205-an-improved-artificial-bee-colony-algorithm-based-on-bayesia.

**Algorithm 2 The pseudo-code of BEABC**

1: /*Initialization phase*/
2: Randomly generate an initial population by (1);
3: Obtain \(x_g\) of the food sources at the initial iteration.
4: for \(T = 1\) to \(\text{Max Dt}\) do
5: /*Employed bee phase*/
6: Calculate MR by (10).
7: for \(i = 1\) to \(SN\) do
8: Generate new food sources \(v_i\) by (2).
9: if \(f(v_i) < f(x_i)\) then
10: Replace \(x_i\) with \(v_i\), and set \(\text{trial}_i = 0\).
11: else
12: \(\text{trial}_i = \text{trial}_i + 1\).
13: end if
14: end for
15: Calculate selection probability \(P_i\) by (5).
16: /*Onlooker bee phase*/
17: for \(i = 1\) to \(SN\) do
18: if \(\text{rand} < P_i\) then
19: if \(f(v_i) < f(x_i)\) then
20: Generate new solutions \(v_j\) by (8).
21: else
22: Generate new solutions \(v_j\) by (9).
23: end if
24: end if
25: if \(f(v_i) < f(x_i)\) then
26: Replace \(x_i\) with \(v_i\), and set \(\text{trial}_i = 0\).
27: else
28: \(\text{trial}_i = \text{trial}_i + 1\).
29: end if
30: end for
31: /*Scout bee phase*/
32: if \(\text{Max(trial)}_i = \text{limit}\) then
33: \(\text{trial}_i = 0\);
34: Generate a new food source \(x_i\) by (10).
35: end if
36: end for
37: Output the final solution.
38: End.

The flowchart of BEABC is shown in Fig. 2. From the figure, the selection probability is calculated by (5). In the onlooker bee and scout bee phases, the food sources are updated according to the new movement formulas (8)–(10), respectively. In addition, evolutionary perturbations are controlled by (11) in each iteration.

**Numerical and engineering experiments**

To validate the performance of BEABC, 31 classical test functions (24 single-objective and 7 multi-objective test functions) and 2 real-world optimization problems are implemented separately in this section. The experiment results are given and discussed. All these algorithms are coded in Matlab R2017a, and ran on PC with Core i5, 2G memory, Windows 7.

In “Sensitivity test for the parameter MR”, a sensitivity test of MR is discussed. In “Experiment 1”, the benchmark functions selected are tested by BEABC and other excellent ABCs. Meanwhile, the discussion and analysis of the results are given. In “Experiment 2”, BEABC are compared with some excellent EAs in dealing with 2 practical engineering problems.
As pointed out in Ref. [15], the control parameter and modified rate (MR) can effectively enhance the performance of ABC. From (11), we can see that MR with a large value means that the candidate solution \( v_i \) obtains more information from the parents, which can improve the quality of \( v_i \). However, the population diversity may be sacrificed. While MR with a small value has limited room to improve the performance of the algorithm, especially in the early stage. Thus, an exponentially varying MR which gradually decreases from MR\(_{\text{max}}\) is introduced, and a sensitivity test is carried out to study the affect of MR\(_{\text{max}}\). To test the sensitivity of MR, several complex test functions with \( D=30 \) are selected. Considering the improvement of the exploitation capacity of BEABC, different values for MR\(_{\text{max}}\) are set to test its affect, i.e., 0.5, 0.6, 0.7, 0.8 and 0.9. The numerical results and the box-plots are given in Table 2 and Fig. 3, respectively. From Table 2, it can be seen that, BEABC has a better performance when MR\(_{\text{max}}\) has a larger value on most functions. The reason may be that the candidate solutions have more opportunities to inherit the parents’ information. From Fig. 3, it also can be seen that BEABC has a better performance when MR\(_{\text{max}}\) is 0.9. From the above discussion, setting MR\(_{\text{max}}\) to 0.9 may be a good choice.

**Experiment 1**

**Single-objective test functions**

In this experiment, 24 classical test functions [1,37] are selected to validate the effectiveness of BEABC. Table 3 gives the detailed information of these test functions. \( f_1- f_8 \) are unimodal functions, \( f_9- f_{16} \) are multimodal functions, \( f_{17}- f_{24} \) are rotated and shifted functions. Some excellent ABCs are chosen to compare with BEABC, including dABC [16], MGABC [23], GABC [24] and COABC [25]. The comparison results obtained by all comparative algorithms are shown in Tables 5, 6 and 7, respectively. And the evaluation indicators include minimum (Min), Mean (mean), standard (Std), rank and \( t \) test. Based on Mean, the ranking makes it very clear how the algorithms are performing. \( t \) test is used to ascertain whether the results obtained by BEABC are statistically alien to its competitors. The significance level \( \alpha \) is set to 0.05. ‘+’, ‘≈’ and ‘-‘ mean that BEABC is statistically better, equal to or significantly inferior to its peers, respectively. To visually show the convergence rate of these algorithms, the convergence curves of the benchmark functions are displayed in Figs. 4, 5 and 6.

To ensure fairness, the basic parameters and running environment of all the algorithms are the same. Table 4 displays the specific parameters of these algorithms, which are the same as the corresponding literatures. Meanwhile, each numerical function is independently run 30 times with each algorithm.

**Unimodal functions:** In Table 5, the numerical results of unimodal functions are given. From the table, BEABC can obtain the global best solution on each function and dominates all ABCs in terms of Min. According to Mean, BEABC takes the first place in all algorithms, and \( t \) test implies that BEABC is statistically significantly superior to the com-
### Table 2  Test results of BEABC with different MR\(_{\text{max}}\)

| Function | Indicators | MR\(_{\text{max}} = 0.5\) | MR\(_{\text{max}} = 0.6\) | MR\(_{\text{max}} = 0.7\) | MR\(_{\text{max}} = 0.8\) | MR\(_{\text{max}} = 0.9\) |
|----------|------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| \(f_6\)  | Min        | 5.30E−06                 | 2.45E−06                 | 4.72E−06                 | 2.11E−06                 | 1.40E−06                 |
| \(f_8\)  | Min        | 0                        | 0                        | 0                        | 0                        | 0                        |
| \(f_{15}\)| Min       | −78.3323                 | −78.3323                 | −78.3323                 | −78.3323                 | −78.3323                 |
| \(f_{16}\)| Min       | 1.85E−13                 | 4.05E−12                 | 8.89E−13                 | 4.67E−13                 | 2.15E−13                 |
| \(f_{23}\)| Min       | −199.1047                | −200.333                 | −198.8984                | −200.4625                | −222.5878                |
| \(f_{24}\)| Min       | 103.9459                 | 103.5215                 | 102.5717                 | 101.5214                 | 100.2322                 |

The optimal results are marked in bold.

### Fig. 3  Sensitivity test for MR

### Table 3  Benchmark test functions

| Function types | Test functions | Range       | Optimal value |
|----------------|----------------|-------------|---------------|
| Unimodal       | \(f_1 = \sum_{i=1}^{D} x_i^2\) | \([-100,100]\) | 0             |
|                | \(f_2 = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i|\) | \([-10, 10]\) | 0             |
|                | \(f_3 = \sum_{i=1}^{D} |x_i| + 1\) | \([-1, 1]\) | 0             |
|                | \(f_4 = \sum_{i=1}^{D} 10^D |x_i|\) | \([-100, 100]\) | 0             |
|                | \(f_5 = \sum_{i=1}^{D} (|x_i| + 0.5)^2\) | \([-1.28, 1.28]\) | 0             |
|                | \(f_6 = \sum_{i=1}^{D} i x_i^2 + \text{random}[0, 1]\) | \([-1.28, 1.28]\) | 0             |
|                | \(f_7 = -\exp(-0.5 + \sum_{i=1}^{D} x_i^2)\) | \([-1, 1]\) | −1            |
|                | \(f_8 = \sum_{i=1}^{D} (\sum_{j=1}^{D} x_j)^2\) | \([-100, 100]\) | 0             |
pared algorithms on $f_1$, $f_2$, $f_4$, $f_6$ and $f_8$. For $f_3$ and $f_5$, BEABC performs equal to MGABC but superior to GABC, dABC and COABC. Although all ABCs can obtain the global best solution and perform equally well on $f_1$, BEABC and MGABC have samll Std. Figure 4 shows that all these algorithms obtained a local optimum on $f_6$, but BEABC has the fastest convergence rate. The reason may be that the new probability mechanism adopted by BEABC prefers to select good solutions.

Multimodal functions: From Table 6, for $f_{10}$, $f_{12}$, $f_{14}$ and $f_{15}$, in terms of Min and Mean, BEABC outperforms the compared ABCs and takes the first place. Meanwhile, t test indicates that BEABC is statistically significantly better than all compared ABCs in most functions. For $f_9$ and $f_{13}$, BEABC performs equal to MGABC and dABC while performs equal to GABC, COABC. However, for $f_{16}$, BEABC only performs better than dABC. From Fig. 5, although more than one algorithm get the theoretical global best solutions on $f_9$, $f_{11}$ and $f_{13}$, the convergence rate of all peers is much lower than that of BEABC. Unfortunately, all these algorithms are caught in a poor location on $f_{16}$. Based on the above discussion, although BEABC can really do well in most multimodal functions, there is still room to enhance the search ability of BEABC.

Rotated and shifted: From Table 7, on $f_{17}$–$f_{20}$, BEABC performs well and obtains the global best solutions. Moreover, t test implies that it is significantly better than all its competitors. For $f_{23}$–$f_{24}$, BEABC did not perform very well on the shifted functions, and it is second to MGABC and COABC in terms of Rank. However, though the performance of BEABC is inferior to MGABC, it is still significantly superior than its peers in most cases at Rank and t test. From Fig. 6, BEABC converges rapidly in rotated functions but it shows signs of premature convergence in shifted functions. The reason may be that the shifted behavior does not fit the search process of the algorithm.
The results of above experiments indicate that BEABC performs well on most single-objective functions. Although the oscillation are inevitable in solving some complex optimization problems, the convergence speed is obviously dominant. Due to the particularity of some functions, especially the shifted functions, the performance of the proposed algorithm needs to be further improved.

**Significance statistics**

To analyze the significant difference between BEABC and other ABCs, the comprehensive statistical results got by $t$ test with $\alpha = 0.05$ level are given in Table 8. From the table, compared with its peers, the significant excellent times of BEABC is 73, and the overall experimental success ratio is
76%. The statistical results show that BEABC has a better performance than its comparison algorithms.

### Multi-objective functions

To further investigate the comprehensive performance of the proposed algorithm, seven continuous multiobjective optimization problems (MOPs) are carefully selected as the test functions, including UF1–UF7 [38]. All MOPs are unconstrained (bound constrained) two-objective test problems, and the number of decision variables $n$ is 30. Tables 9 and 10 give the details of UF1–UF7, including their function representations, PF and PS.

BEABC is compared with several different algorithms, including MOEA/D [39], NSGA-IIIS [40], MTS [41], OMOEII [42] and MO-ABC/DE [43]. Let $M$ be a set of uniformity distributed points along the Pareto front in the feasible space, and $C$ be an approximate set to the Pareto front. IGD is used as the comparative indicators to evaluate the results obtained, which is defined as follows:

$$
IGD(C, M) = \frac{\sum_{v \in C} d(v, C)}{|M|}. \tag{13}
$$

Due to the space limit, the results in this paper are taken directly from [43], except for BEABC. The average IGD is obtained by running each algorithm 30 times independently for each function. The results are given in Table 11. From the table, we find that BEABC takes the first place for most test functions. For UF3 and UF7, the performance of BEABC is slightly worse than that of MOEA/D. In addition, the IGD results obtained by BEABC are very close to 0, which means that the PF obtained by BEABC are very close to the true Pareto front.
The optimal results are marked in bold.

### Computational time complexity

The computational time complexity of our algorithm and other ABCs are analyzed in this subsection. The details are presented in Table 12. For a problem $f$, assume that $O(f)$ is the computational time complexity of evaluating its function value. SN is the population size. The time complexity of conventional ABC is $O(SN \times f + SN \times f) = O(SN \times f)$ [44]. For BEABC, it designs a dynamic control parameter MR to determine the frequency of search, and the complexity of

#### Table 5 Computational results of unimodal test functions

| Functions | Algorithm | Min | Mean | Std | Rank |
|-----------|-----------|-----|------|-----|------|
| $f_1$     | BEABC     | 0   | 0    | 0   | 1    |
|           | MGABC     | 2.16E−308 | 2.97E−298 | 0 | 2  |
|           | GABC      | 2.10E−74 | 2.91E−73 | 2.00E−73 | 4  |
|           | dABC      | 2.26E−24 | 8.37E−24 | 5.49E−24 | 5  |
|           | COABC     | 3.95E−110 | 8.00E−109 | 9.86E−109 | 3  |
| $f_2$     | BEABC     | 0   | 0    | 0   | 1    |
|           | MGABC     | 5.83E−183 | 8.96E−181 | 0 | 2  |
|           | GABC      | 3.93E−38 | 9.85E−38 | 5.04E−38 | 4  |
|           | dABC      | 5.40E−20 | 1.60E−19 | 1.21E−19 | 5  |
|           | COABC     | 1.41E−59 | 6.90E−59 | 4.32E−59 | 3  |
| $f_3$     | BEABC     | 0   | 0    | 0   | 1    |
|           | MGABC     | 0   | 0    | 0   | 1    |
|           | GABC      | 1.18E−110 | 6.87E−103 | 1.55E−102 | 3  |
|           | dABC      | 5.49E−22 | 2.61E−19 | 6.42E−19 | 4  |
|           | COABC     | 7.11E−269 | 1.18E−254 | 0 | 2  |
| $f_4$     | BEABC     | 0   | 0    | 0   | 1    |
|           | MGABC     | 3.88E−302 | 2.79E−295 | 0 | 2  |
|           | GABC      | 7.81E−65 | 3.48E−64 | 2.87E−64 | 4  |
|           | dABC      | 4.32E−20 | 2.48E−19 | 1.80E−19 | 5  |
|           | COABC     | 2.52E−106 | 4.99E−105 | 4.84E−105 | 3  |
| $f_5$     | BEABC     | 0   | 0    | 0   | 1    |
|           | MGABC     | 0   | 0    | 0   | 1    |
|           | GABC      | 0   | 0    | 0   | 1    |
|           | dABC      | 0   | 0    | 0   | 1    |
|           | COABC     | 0   | 0    | 0   | 1    |
| $f_6$     | BEABC     | 4.12E−07 | 1.04E−05 | 8.98E−06 | 1    |
|           | MGABC     | 5.30E−06 | 7.42E−05 | 8.53E−05 | 1  |
|           | GABC      | 0.0097 | 0.0171 | 0.004 | 3  |
|           | dABC      | 0.0107 | 0.0252 | 0.0062 | 4  |
|           | COABC     | 0.0062 | 0.0138 | 0.0029 | 2  |
| $f_7$     | BEABC     | 1   | 1    | 0   | 1    |
|           | MGABC     | 1   | 1    | 0   | 1    |
|           | GABC      | 1   | 1    | 7.14E−17 | 4  |
|           | dABC      | 1   | 1    | 4.12E−17 | 3  |
|           | COABC     | 1   | 1    | 2.06E−17 | 2  |
| $f_8$     | BEABC     | 0   | 0    | 0   | 1    |
|           | MGABC     | 1.71E−14 | 9.54E−12 | 1.53E−11 | 2  |
|           | GABC      | 581.445 | 4.07E+03 | 1.63E+03 | 5  |
|           | dABC      | 1.04E+04 | 1.66E+04 | 2.53E+04 | 4  |
|           | COABC     | 540.7858 | 1.14E+03 | 365.0086 | 3  |
Table 6: Computational results of multiunimodal test functions

| Functions | Algorithm | Min   | Mean  | Std   | Rank |
|-----------|-----------|-------|-------|-------|------|
| $f_9$     | BEABC     | 0     | 0     | 0     | 1    |
|           | MGABC     | 0     | 12.174| 7.3898| 2    | +   |
|           | GABC      | 0     | 0     | 0     | 1    | ≈   |
|           | dABC      | 57.6034| 163.0894| 18.2044| 3    | +   |
|           | COABC     | 0     | 0     | 0     | 1    | ≈   |
| $f_{10}$  | BEABC     | $-8.88E-16$ | $-8.88E-16$ | 0    | 1   |
|           | MGABC     | 2.66E-15| 2.66E-15| 0    | 2    | +   |
|           | GABC      | 2.40E-14| 2.92E-14| 2.59E-15| 4    | +   |
|           | dABC      | 5.31E-12| 1.02E-11| 3.40E-12| 5    | +   |
|           | COABC     | 1.33E-14| 1.66E-14| 3.48E-15| 3    | +   |
| $f_{11}$  | BEABC     | 0     | 0     | 0.0028| 0.0052| 2    | +   |
|           | MGABC     | 0     | 0     | 0     | 1    | ≈   |
|           | GABC      | 0     | 0     | 0     | 1    | ≈   |
|           | dABC      | 0     | 0     | 0     | 1    | ≈   |
|           | COABC     | 0     | 0     | 0     | 1    | ≈   |
| $f_{12}$  | BEABC     | 0     | 0     | 0     | 1    |
|           | MGABC     | 1.19E-261| 2.42E-15| 2.27E-15| 3    | +   |
|           | GABC      | 1.26E-23| 3.97E-09| 1.43E-08| 4    | +   |
|           | dABC      | 0.0018| 0.0171| 0.0058| 5    | +   |
|           | COABC     | 1.15E-71| 1.28E-16| 5.55E-16| 2    | +   |
| $f_{13}$  | BEABC     | 0     | 0     | 0.0028| 0.0052| 2    | +   |
|           | MGABC     | 0     | 0     | 0     | 1    | ≈   |
|           | GABC      | 0     | 0     | 0     | 1    | ≈   |
|           | dABC      | 126.774| 164.3472| 16.3538| 3    | +   |
|           | COABC     | 0     | 0     | 0     | 1    | ≈   |
| $f_{14}$  | BEABC     | 0     | 0     | 0     | 1    |
|           | MGABC     | 0.00097| 0.0354| 0.007 | 2    | +   |
|           | GABC      | 0.127 | 0.2289| 0.0464| 5    | +   |
|           | dABC      | 0.0372| 0.0374| 8.03E-04| 3    | +   |
|           | COABC     | 0.0372| 0.136 | 0.0316| 4    | +   |
| $f_{15}$  | BEABC     | $78.332$ | $78.3321$ | 7.26E-05| 1   |
|           | MGABC     | $-20$ | $-19.5681$ | 0.3201| 3    | +   |
|           | GABC      | $-20$ | $-20$ | 0     | 2    | +   |
|           | dABC      | $-20$ | $-20$ | 0     | 2    | +   |
|           | COABC     | $-20$ | $-20$ | 0     | 2    | +   |
| $f_{16}$  | BEABC     | 5.13E-08| 8.41E-08| 2.47E-08| 2   |
|           | MGABC     | $5.00E-33$ | $5.00E-33$ | 0    | 1   | −   |
|           | GABC      | $5.00E-33$ | $5.00E-33$ | 0    | 1   | −   |
|           | dABC      | 1.46E-09| 0.0677| 0.2425| 4    | +   |
|           | COABC     | $5.00E-33$ | $5.00E-33$ | 0    | 3   | −   |

The optimal results are marked in bold.

The initialize phase is $O(SN \ast f)$. So the complexity are $O(MR \ast (SN \ast f + SN \ast f))$ in the employed and onlooker bee phases. Therefore, the total computational complexity of BEABC is $O(SN \ast f)$ at each iteration.

From Table 12, we can see that most ABCs have the same complexity as conventional ABC except for dABC. The main reason is that they do not have redundant loops to calculate function values. Moreover, BEABC does not need to calculate the fitness value of the function, which may save time to a certain extent.
Table 7  Computational results of rotated and shifted test functions

| Functions | Algorithm | Min        | Mean      | Std       | Rank | $t$ test |
|-----------|-----------|------------|-----------|-----------|------|----------|
| $f_{17}$  | BEABC     | 0          | 0         | 0         | 1    |          |
|           | MGABC     | 3.80E−06   | 2.63E+03  | 1.44E+04  | 3    | +        |
|           | GABC      | 198.876    | 1.87E+03  | 1.50E+03  | 2    | +        |
|           | dABC      | 1.29E+04   | 4.48E+04  | 2.25E+04  | 5    | +        |
|           | COABC     | 526.3128   | 3.56E+03  | 1.71E+03  | 4    | +        |
| $f_{18}$  | BEABC     | -8.88E−16  | -8.88E−16 | 0         | 1    |          |
|           | MGABC     | 2.66E−15   | 0.0811    | 0.3176    | 3    | +        |
|           | GABC      | 2.0214     | 4.6478    | 3.8222    | 5    | +        |
|           | dABC      | 1.13E−11   | 4.12E−11  | 1.99E−11  | 2    | +        |
|           | COABC     | 2.4114     | 3.6252    | 0.7955    | 4    | +        |
| $f_{19}$  | BEABC     | 0          | 0         | 0         | 1    |          |
|           | MGABC     | 2.33E−302  | 1.38E−297 | 0         | 2    | +        |
|           | GABC      | 2.66E−69   | 3.75E−68  | 3.10E−68  | 4    | +        |
|           | dABC      | 7.63E−21   | 3.59E−20  | 3.07E−20  | 5    | +        |
|           | COABC     | 3.86E−107  | 7.98E−106 | 1.67E−105 | 3    | +        |
| $f_{20}$  | BEABC     | 0          | 0         | 0         | 1    |          |
|           | MGABC     | 0.0025     | 0.0046    |           | 5    | +        |
|           | GABC      | 2.22E−16   | 1.07E−15  | 6.77E−16  | 2    | +        |
|           | dABC      | 2.65E−08   | 2.36E−06  | 6.57E−06  | 4    | +        |
|           | COABC     | 1.11E−16   | 1.76E−15  | 6.91E−15  | 3    | +        |
| $f_{21}$  | BEABC     | -301.3735  | -290.4609 | 5.4953    | 3    |          |
|           | MGABC     | -323.2276  | -309.2738 | 12.1412   | 2    | –        |
|           | GABC      | -330       | -330      | 1.06E−14  | 1    | –        |
|           | dABC      | -283.6052  | -202.1892 | 37.9428   | 4    | +        |
|           | COABC     | -330       | -330      | 0         | 1    | –        |
| $f_{22}$  | BEABC     | 440.7887   | 779.3214  | 177.1517  | 3    |          |
|           | MGABC     | -450       | -115.1108 | 863.8464  | 1    | –        |
|           | GABC      | -41.6327   | 4.66E+03  | 2.30E+03  | 5    | +        |
|           | dABC      | 1.83E+03   | 4.19E+03  | 1.10E+03  | 4    | +        |
|           | COABC     | 25.0538    | 824.0376  | 410.2707  | 2    | –        |
| $f_{23}$  | BEABC     | -240.6713  | -162.7325 | 37.1307   | 2    |          |
|           | MGABC     | -308.1109  | -284.8146 | 16.5199   | 1    | –        |
|           | GABC      | -205.556   | -155.5573 | 27.745    | 3    | +        |
|           | dABC      | -158.3339  | -137.0086 | 9.76      | 4    | +        |
|           | COABC     | -189.6751  | -118.1012 | 31.0623   | 5    | +        |
| $f_{24}$  | BEABC     | 101.3422   | 105.995   | 2.6534    | 2    |          |
|           | MGABC     | 96.9175    | 103.6899  | 3.6251    | 1    | –        |
|           | GABC      | 114.8005   | 118.6542  | 1.9475    | 4    | +        |
|           | dABC      | 118.2062   | 122.0539  | 1.461     | 5    | +        |
|           | COABC     | 113.6376   | 117.5387  | 1.8989    | 3    | +        |

The optimal results are marked in bold

**Experiment 2**

To verify the effectiveness of one optimization algorithm, according to [45,46], the real-world engineering optimization problems are good choices. so, to evaluate BEABC effectively, two engineering optimization problems are selected in this subsection. However, since these two practical optimization problems are basically constrained, we need to transform them into unconstrained optimization problems to facilitate the calculation of the algorithms. The following methods can be used, including rejection method, repair method, operator correction method and penalty function method. In this paper, the penalty function method is used.
Table 8  Significance statistics

| Algorithms | + | − | ≈ |
|------------|---|---|---|
| MGABC      | 16| 5 | 3 |
| GABC       | 18| 2 | 4 |
| dABC       | 22| 0 | 2 |
| COABC      | 17| 3 | 4 |
| Total      | 73| 10| 13|
| Ratio      | 76%|10.4%|13.6%|

Table 8: Significance statistics

Structural design of tension–pressure spring

The optimization design problem is described in Ref. [47], which minimizes the weight of a tension spring under the constraints such as shear stress, frequency of vibration and so on. There are three main design variables: coil diameter $d(x_1)$, average diameter of spring coil $D(x_2)$ and the effective number of circles $N(x_3)$. Figure 7 indicates the schematic diagram of this design problem. The engineering optimization problem is depicted as:

$$\min \quad f(x) = x_2 x_1^2 (x_3 + 2)$$

s.t. $g_1(x) = 1 - \frac{x_2 x_3}{71/85 x_1} \leq 0,$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1 - x_3)} + \frac{1}{5108 x_1^2} - 1 \leq 0,$$

$$g_3(x) = 1 - \frac{140,45x_1}{x_2 x_3} \leq 0,$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0,$$

where

$$0.05 \leq x_1 \leq 2, \quad 0.25 \leq x_2 \leq 1.3,$$

$$2 \leq x_3 \leq 15.$$

Several optimization algorithms have been developed to solve the optimization problem, such as CNPSO [5], SPA [48], MPM [49], IOD [50], TGA [51] and CPSO [52]. Table 13 contains the optimum solutions and the constrained solutions obtained by these algorithms. Besides, the Min, Mean and Std are used as the evaluation indicators. Table 14 shows the statistical results. Except for BEABC, the solutions offered by these algorithms are taken directly from [5]. By the results, though the constraint $g_1(x)$ is approximately satisfied, the small enough difference 1.5814040E−07 can be ignored in practice. In terms of Min ($f(x)$), BEABC outperforms other algorithms, and we can come to the conclusion that BEABC is efficient in solving this optimization problem.

Design of a speed reducer gearing system

The gear design problem is to minimize the weight of a speed reducer at the mercy of constraints on bending stress of the gear teeth, transverse deflections of the shafts, surface stress and stresses in the shafts [53]. This optimization problem involves seven variables, and $x_1$-$x_7$ are used to represent these variables, respectively. All the variables are incessant except number of teeth ($x_3$) which is an integer value. The reducer diagram is depicted in Fig. 8. All above, the design objective function is described as:

$$\min \quad f(x) = 0.7854x_1 x_2^2 (3.3333x_3^2$$

$$+ 14.9334x_3 - 43.0934) - 1.508x_1 (x_5^6$$

$$+ x_7^2) + 7.4777(x_3^3 + x_3^4) + 0.7854(x_4 x_6^2 + x_5 x_7^2)$$

s.t. $g_1(x) = \frac{27}{x_1 x_2^2 x_3} - 1 \leq 0,$

$$g_2(x) = \frac{59.75}{x_1 x_2^2 x_3} - 1 \leq 0,$$

$$g_3(x) = \frac{1.93x_4^3}{x_2 x_5^2 x_3} - 1 \leq 0,$$

$$g_4(x) = \frac{1.93x_5^3}{x_2 x_5^2 x_3} - 1 \leq 0,$$

$$g_5(x) = \frac{(745x_1^4 x_2^2 + 1.69x_1^2)}{110x_6^2} - 1 \leq 0,$$

$$g_6(x) = \frac{(745x_1^4 x_2^2 + 157.5x_1^2)}{85x_7^2} - 1 \leq 0,$$

$$g_7(x) = \frac{x_2 x_3}{40} - 1 \leq 0,$$

$$g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0,$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0,$$

$$g_{10}(x) = \frac{1.5x_2 + 1.9}{x_4} - 1 \leq 0,$$

$$g_{11}(x) = \frac{1.1x_2 + 1.9}{x_5} - 1 \leq 0,$$

where

$$2.6 \leq x_1 \leq 3.6, \quad 0.7 \leq x_2 \leq 0.8,$$

$$17 \leq x_3 \leq 28, \quad 7.3 \leq x_4, x_5 \leq 8.3,$$

$$2.9 \leq x_6 \leq 3.9, \quad 5 \leq x_7 \leq 5.5.$$

By reviewing the literatures, we find that some intelligent algorithms have been applied to solve this design problem, including DELC [54], DEDS [55], HEAA [56], PSO-DE [57], MDE [58] and MBA [59]. The optimal solutions and the statistical results obtained by each algorithm, which are taken directly from [59], are given in Tables 15 and 16, respectively. From the results, although the Min and Mean of BEABC are
Table 9  Multi-objective benchmark functions

| Functions | Range          | Functions expression, PF and PS |
|-----------|----------------|---------------------------------|
| UF1       | $[0,1] \times [-1, 1]^{m-1}$ | $f_1 = x_1 + \frac{2}{\sqrt{n}} \sum_{j \in J_1} |x_j - \sin(6\pi x_1 + \frac{j\pi}{n})|^2$  
$f_2 = 1 - \sqrt{n} + \frac{2}{\sqrt{n}} \sum_{j \in J_2} |x_j - \sin(6\pi x_1 + \frac{j\pi}{n})|^2$  
$J_1 = \{ j | j \text{ is odd and } 2 \leq j \leq n \}$  
$J_2 = \{ j | j \text{ is even and } 2 \leq j \leq n \}$  
PF : $f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$  
PS : $x_j = \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, \ldots, n, 0 \leq x_1 \leq 1$ |
| UF2       | $[0,1] \times [-1, 1]^{m-1}$ | $f_1 = x_1 + \frac{2}{\sqrt{n}} \sum_{j \in J_1} y_j^2$  
$f_2 = 1 - \sqrt{n} + \frac{2}{\sqrt{n}} \sum_{j \in J_2} y_j^2$  
$J_1 = \{ j | j \text{ is odd and } 2 \leq j \leq n \}$  
$J_2 = \{ j | j \text{ is even and } 2 \leq j \leq n \}$  
$y_j = \begin{cases} 
0.3x_j^2 \cos \left(24\pi x_1 + \frac{4j\pi}{n}\right) \sin \left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_1 \\
0.3x_j^2 \cos \left(24\pi x_1 + \frac{4j\pi}{n}\right) \sin \left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_2 
\end{cases}$  
PF : $f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$  
PS : $x_j = \begin{cases} 
0.3x_j^2 \cos \left(24\pi x_1 + \frac{4j\pi}{n}\right) \sin \left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_1, 0 \leq x_1 \leq 1 \\
0.3x_j^2 \cos \left(24\pi x_1 + \frac{4j\pi}{n}\right) \sin \left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_2, 0 \leq x_1 \leq 1 
\end{cases}$ |
| UF3       | $[0,1]^m$       | $f_1 = x_1 + \frac{2}{\sqrt{n}} \left(4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos \left(\frac{20\pi y_j}{\sqrt{n}}\right) + 2\right)$  
$f_2 = 1 - \sqrt{n} + \frac{2}{\sqrt{n}} \left(4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos \left(\frac{20\pi y_j}{\sqrt{n}}\right) + 2\right)$  
$J_1 = \{ j | j \text{ is odd and } 2 \leq j \leq n \}$  
$J_2 = \{ j | j \text{ is even and } 2 \leq j \leq n \}$  
$y_j = x_j - x_1^{0.5(1.0+\frac{\sqrt{2}}{n})}, j = 2, \ldots, n$  
PF : $f_2 = 1 - \sqrt{f_1}, 0 \leq f_1 \leq 1$  
PS : $x_j = x_1^{0.5(1.0+\frac{\sqrt{2}}{n})}, j = 2, \ldots, n, 0 \leq x_1 \leq 1$ |
| UF4       | $[0,1] \times [-2, 2]^{m-1}$ | $f_1 = x_1 + \frac{2}{\sqrt{n}} \sum_{j \in J_1} h(y_j)$  
$f_2 = 1 - x_1^2 + \frac{2}{\sqrt{n}} \sum_{j \in J_2} h(y_j)$  
$J_1 = \{ j | j \text{ is odd and } 2 \leq j \leq n \}$  
$J_2 = \{ j | j \text{ is even and } 2 \leq j \leq n \}$  
$y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, \ldots, n$  
$h(t) = \frac{1}{1+e^t}$  
PF : $f_2 = 1 - f_1^2, 0 \leq f_1 \leq 1$  
PS : $x_j = \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, \ldots, n, 0 \leq x_1 \leq 1$ |
| UF5       | $[0,1] \times [-1, 1]^{m-1}$ | $f_1 = x_1 + (\frac{1}{\sqrt{\pi}} + \varepsilon) \sin(2N\pi x_1) + \frac{2}{\sqrt{n}} \sum_{j \in J_2} h(y_j), \varepsilon = 0.1$  
$f_2 = 1 - x_1 + (\frac{1}{\sqrt{\pi}} + \varepsilon) \sin(2N\pi x_1) + \frac{2}{\sqrt{n}} \sum_{j \in J_2} h(y_j), \varepsilon = 0.1$  
$J_1 = \{ j | j \text{ is odd and } 2 \leq j \leq n \}$  
$J_2 = \{ j | j \text{ is even and } 2 \leq j \leq n \}$  
$y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, \ldots, n$  
$h(t) = 2t^2 - \cos(4\pi t) + 1$  
PF : $\left(\frac{1}{\sqrt{\pi}}, 1 - \frac{\pi}{\sqrt{\pi}}\right), i = 0, 1, \ldots, 2N, N = 10$ |
The computational time complexity

| Algorithm | Initialize | Function calculation | Total computational complexity |
|-----------|------------|----------------------|------------------------------|
| ABC       | $O(SN \ast f)$ | $O(SN \ast f + SN \ast f)$ | $O(SN \ast f)$ |
| BEABC     | $O(SN \ast f)$ | $O(MR \ast (SN \ast f + SN \ast f))$ | $O(SN \ast f)$ |
| MGABC     | $O(SN \ast f)$ | $O(SN \ast f + SN \ast f + 0.1 \ast SN \ast f)$ | $O(SN \ast f)$ |
| GABC      | $O(SN \ast f)$ | $O(SN \ast f + SN \ast f)$ | $O(SN \ast f)$ |
| dABC      | $O(SN \ast f)$ | $O(SN \ast f + SN^2 \ast f)$ | $O(SN^2 \ast f)$ |
| COABC     | $O(SN \ast f)$ | $O(SN \ast f + 3 \ast SN \ast f)$ | $O(SN \ast f)$ |

The optimal results are marked in bold

Table 10  Multi-objective benchmark functions (continue)

| Functions | Range          | Functions expression, PF and PS |
|-----------|----------------|----------------------------------|
| UF6       | $[0,1] \times [-1, 1]^{n-1}$ | $f_1 = x_1 + \max\{0, 2\left(\frac{n^2}{\sqrt{n}} + \varepsilon\right) \sin(2N\pi x_1)\}$ |
|           |                | $+ \frac{2}{\sqrt{n}} \left(4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos \left(\frac{20y_j\pi}{\sqrt{n}}\right) + 2\right), \ \varepsilon = 0.1$ |
|           |                | $f_2 = 1 - x_1 + \max\{0, 2\left(\frac{1}{\sqrt{n}} + \varepsilon\right) \sin(2N\pi x_1)\}$ |
|           |                | $+ \frac{2}{\sqrt{n}} \left(4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos \left(\frac{20y_j\pi}{\sqrt{n}}\right) + 2\right), \ \varepsilon = 0.1$ |
|           |                | $J_1 = \{j | j \text{ is odd and } 2 \leq j \leq n\}$ |
|           |                | $J_2 = \{j | j \text{ is even and } 2 \leq j \leq n\}$ |
|           |                | $y_j = x_j - \sin \left(6\pi x_1 + \frac{L x}{n}\right), \ j = 2, \ldots, n$ |
|           |                | $\text{PF}: f_2 = 1 - f_1, f_1 \in \bigcup_{m=1}^{N} \left(\frac{2m+1}{2\sqrt{n}}, \frac{2m}{2\sqrt{n}}\right). \ N = 2$ |
|           |                | $f_1 = \sqrt{\frac{n^2}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{j \in J_1} y_j^2}$ |
|           |                | $f_2 = 1 - \sqrt{\frac{n^2}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{j \in J_2} y_j^2}$ |
|           |                | $J_1 = \{j | j \text{ is odd and } 2 \leq j \leq n\}$ |
|           |                | $J_2 = \{j | j \text{ is even and } 2 \leq j \leq n\}$ |
|           |                | $y_j = x_j - \sin \left(6\pi x_1 + \frac{L x}{n}\right), \ j = 2, \ldots, n$ |
|           |                | $\text{PF}: f_2 = 1 - f_1, 0 \leq f_1 \leq 1$ |
|           |                | $\text{PS}: x_j = \sin \left(6\pi x_1 + \frac{L x}{n}\right), \ j = 2, \ldots, n, \ 0 \leq x_1 \leq 1$ |

Table 11  Comparison results using the Mean (IGD) and Std (IGD) of MOPs

| MOPs | Indicators | BEABC | MTS | MO-ABC/DE | OMOEAIH | MSGAIIIS | MOEA/D | Rank |
|------|------------|-------|-----|-----------|---------|----------|--------|------|
| UF1  | Mean (IGD) | 0.0018| 0.00647| 0.00632| 0.08565| 0.01153| 0.00435| 1    |
|      | Std (IGD)  | 5.5872e−04| 3.49e−04| 6.13e−05| 4.07e−03| 7.30e−03| 2.90e−04|      |
| UF2  | Mean (IGD) | 0.0014| 0.00616| 0.00614| 0.03057| 0.1237| 0.00679| 1    |
|      | Std (IGD)  | 2.9697e−04| 5.08e−04| 8.16e−05| 1.61e−03| 9.11e−03| 1.82e−03|      |
| UF3  | Mean (IGD) | 0.0179| 0.05311| 0.04552| 0.27142| 0.10603| 0.00742| 2    |
|      | Std (IGD)  | 4.3e−03| 1.17e−02| 1.84e−03| 3.76e−02| 6.86e−02| 5.89e−03|      |
| UF4  | Mean (IGD) | 0.0020| 0.02356| 0.02896| 0.04625| 0.05850| 0.06385| 1    |
|      | Std (IGD)  | 2.0358e−04| 6.64e−04| 1.34e−03| 9.67e−04| 5.12e−03| 5.34e−03|      |
| UF5  | Mean (IGD) | 0.0910| 0.01489| 0.2500| 0.16920| 0.56570| 0.18071| 1    |
|      | Std (IGD)  | 2.80e−02| 3.28e−03| 1.12e−03| 3.90e−03| 1.83e−01| 6.81e−02|      |
| UF6  | Mean (IGD) | 0.0067| 0.05918| 0.08650| 0.07338| 0.31032| 0.00587| 2    |
|      | Std (IGD)  | 2.2e−03| 1.06e−02| 5.54e−03| 2.45e−03| 1.91e−01| 1.71e−03|      |
| UF7  | Mean (IGD) | 0.0013| 0.04079| 0.05606| 0.03355| 0.02132| 0.00444| 1    |
|      | Std (IGD)  | 5.6231e−04| 1.44e−02| 1.83e−03| 1.74e−03| 1.95e−02| 1.17e−03|      |
Table 13 Comparison of best solution for this problem

| Variables | BEABC   | CNPSO   | SPA     | MPM     | IOD     | TGA     | CPSO   |
|-----------|---------|---------|---------|---------|---------|---------|--------|
| \( x_1 \) | 0.050000| 0.050000| 0.051480| 0.050000| 0.053396| 0.051989| 0.051728|
| \( x_2 \) | 0.317356| 0.607914| 0.351661| 0.315900| 0.399180| 0.363965| 0.357644|
| \( x_3 \) | 4.454718| 2.000000| 11.632201| 14.250000| 9.185400| 10.890522| 11.244543|
| \( g_1(x) \) | 1.797665E-07| -0.647405| -1.853135| -2.169570| -1.508228| -1.750986| -1.798383|
| \( g_2(x) \) | -1.721364E-04| 0.730448| -1.097012| -0.003782| -1.809986| -2.541595| -2.527414|
| \( g_3(x) \) | -14.652297| -8.501178| -4.026318| -3.938301| -4.123831| -4.061337| -4.051306|
| \( g_4(x) \) | -0.755096| -0.561390| -0.731239| -0.756066| -0.698282| -0.722697| -0.727085|
| \( f(x) \) | **0.005121**| 0.006076| 0.012704| 0.012833| 0.012730| 0.012681| 0.012674|

The optimal results are marked in bold;

Table 14 Comparison of statistical results for this problem

| Variables | BEABC       | CNPSO       | SPA     | MPM     | IOD     | TGA     | CPSO   |
|-----------|-------------|-------------|---------|---------|---------|---------|--------|
| Min       | **0.0051207**| 0.006076| 0.012704| 0.012833| 0.012730| 0.012681| 0.012674|
| Mean      | **0.0051209**| 0.006076| 0.012822| NA      | NA      | 0.012973| 0.012924|
| Std       | 1.581404E-07| **2.879554E-009**| 3.939000E-005| NA      | NA      | 5.900000E-005| 5.198500E-005|

The optimal results are marked in bold; NA represents the results are not given in Ref. [5]

Fig. 8 Design of a speed reducer gearing system

equal to those of DELC and DEDS, the Std is the smallest in all competitors. It is easy to draw the conclusion that BEABC is competent to search the best optimum solution and has a good robustness for this problem.

Table 15 Comparison of best solution for speed reducer design problem

| Variables | BEABC       | DELC       | DEDS     | HEAA     | PSO-DE   | MDE     | MBA    |
|-----------|-------------|------------|----------|----------|----------|---------|--------|
| \( x_1 \) | 3.500000    | 3.500000   | 3.500000 | 3.500022 | 3.500000 | 3.500010| 3.500000|
| \( x_2 \) | 0.700000    | 0.700000   | 0.700000 | 0.700000 | 0.700000 | 0.700000| 0.700000|
| \( x_3 \) | 17.000000   | 17.000000  | 17.000000| 17.000012| 17.000000| 17.000000| 17.000000|
| \( x_4 \) | 7.300000    | 7.300000   | 7.300000 | 7.300427 | 7.300000 | 7.300156| 7.300033|
| \( x_5 \) | 7.715319    | 7.715319   | 7.715319 | 7.715377 | 7.800000 | 7.800027| 7.715772|
| \( x_6 \) | 3.350214    | 3.350214   | 3.350214 | 3.350230 | 3.350214 | 3.350221| 3.350218|
| \( x_7 \) | 5.286654    | 5.286654   | 5.286654 | 5.286663 | 5.286683 | 5.286685| 5.286654|
| \( f(x) \) | **2994.471066** | **2994.471066** | **2994.471066** | 2994.499107 | 2996.348167 | 2996.356689 | 2994.482453 |

* The optimal results are marked in bold

Conclusion

A novel ABC algorithm named BEABC is presented in this paper. First, BEABC used a new probability based on Bayesian estimation instead of the fitness ratio for accelerating the convergence rate. Using the new probability selection mechanism, the onlooker bees prefer to choose the solutions that can produce good offsprings. Second, in onlooker bee phase, a directional guidance strategy was presented to tradeoff the balance between exploration and exploitation. Finally, the sensitivity test of MR, the experimental results of numerical functions and two practical optimization problems showed the availability and advancement of BEABC. Although BEABC outperforms its competitors, there is still room to improve its performance in solving the rotated and shifted problems. And in the future, we will further develop the potential of BEABC.
Table 16 Comparison of statistical results for speed reducer design problem

| Evaluation indicator | BEABC   | DELC    | DEDS    | HEAA    | PSO-DE  | MDE     | MBA     |
|----------------------|---------|---------|---------|---------|---------|---------|---------|
| Min                  | 2994.471066 | 2994.471066 | 2994.471066 | 2994.499107 | 2996.348167 | 2996.356689 | 2994.482453 |
| Mean                 | 2994.471066 | 2994.471066 | 2994.471066 | 2994.613368 | 2996.348174 | 2996.367220 | 2996.769019 |
| Std                  | 1.85E−12  | 1.90E−12 | 3.60E−12 | 7.00E−02  | 6.40E−06  | 8.20E−03  | 1.560000 |

The optimal results are marked in bold.

Acknowledgements The authors are grateful to the responsible editor and the anonymous referees for their valuable comments and suggestions, which have improved the earlier version of this paper. This study was funded by the National Natural Science Foundation NSFC (12071133); the Key Cultivation Project of Xianyang Normal University (XSYK21044); the Key Project of Henan Educational Committee (20A110021); the Basic Research Program of Natural Science in Shaanxi Province (2022JM-349).

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendix: Description of parameters and abbreviations

| Sets         | SN       | Set of the food sources |
|--------------|----------|-------------------------|
| D            |          | Set of the vector dimension |
| Variables    | xij      | The jth dimension of the food source i |
|              | xij      | The jth dimension of the neighbor solution n |
|              | vij      | The jth dimension of the candidate solution i |
|              | Pi       | The selection probability |
|              | Fit_i    | The fitness value of x_i |
|              | f(x_i)   | The function value of x_i |
|              | P(B = x_i) | The probability of x_i occurs |

| Parameters   | MR       | Modification rate |
|--------------|----------|-------------------|
|              | MRmax    | The upper bound of modification rate |
|              | xijmax   | The upper bound of the jth dimension of x_i |
|              | xijmin   | The lower bound of the jth dimension of x_i |
|              | r, rand, ψij | The uniform random number in [0, 1] |
|              | λ        | Adjust parameter |
|              | MaxDt    | The maximum iteration number |
|              | Dt       | The current iteration number |
|              | trial_i  | The counter of the food source i |
|              | limit    | The counter threshold |
|              | d(v, C)  | The minimum Euclidean distance between v and the points in C |

| Abbreviations | ABC      | Artificial bee colony |
|---------------|----------|-----------------------|
|               | ABC-AHC  | Adaptive heterogeneous competition augmented ABC |
|               | ABCM     | ABC with memory |
|               | ABCs     | Artificial bee colony variants |
|               | BEABC    | ABC based on Bayesian estimation |
|               | CNPSO    | PSO based on velocity updating mechanism |

| Abbreviations | COABC    | Converge onlookers ABC |
|---------------|----------|------------------------|
|               | CPSO     | Co-evolutionary PSO |
|               | dABC     | Directed ABC |
|               | DE       | Differential evolution |
|               | DEDS     | DE with dynamic stochastic selection |
|               | DELC     | DE with level comparison |
|               | DFS-ABC  | Depth-first search framework ABC |
|               | DPLABC   | ABC with dynamic penalty and Levy flight |
|               | EAs      | Evolutionary algorithms |
|               | FA       | Firefly algorithm |
|               | GA       | Genetic algorithm |
|               | GABC     | Gbest-guided ABC |
|               | HEAA     | Hybrid evolutionary algorithm |
References

1. Wang C, Liu K (2019) A randomly guided firefly algorithm based on elitist strategy and its applications. IEEE Access 7:130373–130387
2. Das AK, Pratihar DK (2019) A directional crossover (DX) operator for real parameter optimization using genetic algorithm. Appl Intel 49(5):1841–1865
3. Hu Z, Gao C, Su Q (2021) A novel evolutionary algorithm based on even difference grey model. Expert Syst Appl 176:114898
4. dos Santos Coelho L, Alotto P (2011) Gaussian artificial bee colony algorithm approach applied to Loney’s solenoid benchmark problem. IEEE Trans Magn 47(5):1326–1329
5. Wang C, Song W (2019) A modified particle swarm optimization algorithm based on velocity updating mechanism. Ain Shams Eng J 10(4):847–866
6. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department
7. Subramanyam G (2019) An improved artificial bee colony algorithm based harmonic control for multilevel inverter. J Control Eng Appl Inform 21(4):59–70
8. Luo H, Wang C, Zhi L, Yan S (2020) Prestack AVO inversion using the improved artificial bee colony algorithm based on exact Zoepritz equations. SEG technical program expanded abstracts 2020. Society of Exploration Geophysicists, pp 345–349
9. Li B, Gong L, Yang W (2014) An improved artificial bee colony algorithm based on balance evolution strategy for unmanned combat aerial vehicle path planning. Sci World J 2014:95–104
10. Jacob MS, Selvi Rajendran P (2021) Fuzzy artificial bee colony-based CNN-LSTM and semantic feature for fake product review classification. Pract Exp Concurr Comput 34:1–16
11. Zorarpci E, Örel SA (2021) Privacy preserving rule-based classifier using modified artificial bee colony algorithm. Expert Syst Appl 183:115437
12. Thilak KD, Amuthan A, Rajkamal S (2021) Mitigating DDoS attacks in VANETs using a variant artificial bee colony algorithm based on cellular automata. Soft Comput 25:12191–12201
13. De Jong K (2007) Parameter setting in EAs: a 30 year perspective. Springer, Berlin, pp 1–18
14. Akay B, Karaboga D (2009) Parameter tuning for the artificial bee colony algorithm. Springer, Berlin, pp 608–619
15. Akay B, Karaboga D (2012) A modified artificial bee colony algorithm for real-parameter optimization. Inf Sci 192:120–142
16. Kiran MS, Findik O (2015) A directed artificial bee colony algorithm. Appl Soft Comput 26:454–462
17. Durgut R, Aydin ME (2021) Adaptive binary artificial bee colony algorithm. Appl Soft Comput 101:107054
18. Wang H, Wang W, Zhou X, Zhao J, Wang Y, Xiao S, Xu M (2021) Artificial bee colony algorithm based on knowledge fusion. Complex Intell Syst 7(3):1139–1152
19. Cui L, Li G, Li Q, Du Z, Gao W, Chen J, Lu N (2016) A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation. Inf Sci 367:1012–1044
20. Liu F, Sun Y, Wang G, Wu T (2018) An artificial bee colony algorithm based on dynamic penalty and Levy flight for constrained optimization problems. Arab J Sci Eng 43(12):7189–7208
21. Chu X, Cai F, Gao D, Li L, Cui J, Xu S, Qin P (2020) An artificial bee colony algorithm with adaptive heterogeneous competition for global optimization problems. Appl Soft Comput 93:106391
22. Chaudhuri A, Sahu TP (2021) Feature weighting for nave Bayes using multi objective artificial bee colony algorithm. Int J Comput Sci Eng 24(1):74–88
23. Zhou X, Lu J, Huang J, Zhong M, Wang M (2021) Enhancing artificial bee colony algorithm with multi-elite guidance. Inf Sci 543:242–258
24. Zhu G, Kwong S (2010) Gbest-guided artificial bee colony algorithm for numerical function optimization. Appl Math Comput 217(7):3166–3173
25. Luo J, Wang Q, Xiao X (2013) A modified artificial bee colony algorithm based on converge-onlookers approach for global optimization. Appl Math Comput 219(20):10253–10262
26. Zhou X, Wu Z, Wang H, Rahmanayan S (2016) Gaussian bare-bones artificial bee colony algorithm. Soft Comput 20(3):907–924
27. Peng H, Deng C, Wu Z (2019) Best neighbor-guided artificial bee colony algorithm for continuous optimization problems. Soft Comput 23(18):8723–8740
28. Yu W, Li X, Cai H, Zeng Z, Li X (2018) An improved artificial bee colony algorithm based on factor library and dynamic search balance. Math Probl Eng 2018:1–16
29. Karaboga D, Gorkemli B, Ozturk C, Karaboga N (2014) A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artif Intell Rev 42(1):21–57
30. Liu J, Zhu H, Ma Q, Zhang L, Xu H (2015) An artificial bee colony algorithm with guide of global and local optima and asynchronous scaling factors for numerical optimization. Appl Soft Comput 37:608–618
31. Karaboga D, Gorkemli B (2014) A quick artificial bee colony (qABC) algorithm and its performance on optimization problems. Appl Soft Comput 23:227–238
32. Jadon SS, Bansal JC, Tiwari R, Sharma H (2018) Artificial bee colony algorithm with global and local neighborhoods. Int J Syst Assur Eng Manag 9(3):589–601
33. Al Mutairi AO (2018) Bayesian estimation using (Linex) for generalized power function distribution. Lobachevskii J Math 39(3):297–303
1. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41(2):113–127
2. Arora JS (2004) Introduction to optimum design. Elsevier, Amsterdam
3. Coello CAC, Montes EM (2002) Constraint-handling through the use of dominance-based tournament selection. Adv Eng Inform 16(3):193–203
4. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20(1):89–99
5. Ku KJ, Rao SS, Chen L (1998) Taguchi-aided search method for design optimization of engineering systems. Eng Optim 30(1):1–23
6. Wang L, Li L (2010) An effective differential evolution with level comparison for constrained engineering design. Struct Multidiscip Optim 41(6):947–963
7. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci 178(15):3043–3074
8. Wang Y, Cai Z, Zhou Y, Fan Z (2009) Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. Struct Multidiscip Optim 37(4):395–413
9. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl Soft Comput 10(2):629–640
10. Mezura-Montes E, Coello CAC, Velazquez-Reyes J (2006) Increasing successful offspring and diversity in differential evolution for engineering design. In: Proceedings of the seventh international conference on adaptive computing in design and manufacture, pp 131–139
11. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13(5):2592–2612

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.