Evolving Systems (2021) 12:191–205
https://doi.org/10.1007/s12530-019-09307-3

ORIGINAL PAPER

A complex-valued encoding satin bowerbird optimization algorithm for global optimization

Sen Zhang[1] · Yongquan Zhou[1,2] · Qifang Luo[1,2]

Received: 3 May 2019 / Accepted: 10 October 2019 / Published online: 3 November 2019

Abstract

The real-valued satin bowerbird optimization (SBO) is a novel metaheuristic bio-inspired algorithm which imitates the 'male-attracts-the-female for breeding' principle of the specialized stick structure mechanism of satin birds. SBO has achieved success in congestion management, accurate software development effort estimation. In this paper, to enhance the SBO algorithm global exploration ability, a complex-valued encoding satin bowerbird optimization algorithm (CSBO) is proposed. We use complex-valued encoding enhance the diversity of the population, and the global exploration ability of the SBO algorithm. The proposed CSBO optimization algorithm is compared to SBO and other state-of-art optimization algorithms using ten benchmark functions. Simulation results show that the proposed CSBO can significantly improve the convergence accuracy and convergence speed of the original algorithm.

Keywords Complex-valued encoding · Satin bowerbird optimization · Benchmark functions · Metaheuristic algorithm

1 Introduction

Metaheuristic algorithm has been become very popular over the past decade. This popularity is due to several key reasons: flexibility, lack of a gradient mechanism, simple structure and easy-to-understand. Which can convenient and quickly solve various problems. In Fister et al. (2013), the metaheuristic algorithms are divide into four major categories: swarm intelligence (SI) based, bio-inspired (but not SI-based), physics/chemistry-based, and others. The evolutionary algorithm simulates the concept of principle evolution, the most popular of which is the genetic algorithm (GA) (Goldberg and Holland 1988), which simulates Darwin's evolutionary theory, the principle is that optimization is a set of stochastic solutions for specific problems. In the case of the evaluation objective function, it updates the variables according to their fitness values. Similarly, there are many evolutionary algorithms, such as differential evolution (DE) (Rainer and Price 1997). The ecosystem simulation algorithms, such as, biogeography-based optimization (BBO) (Simon 2008). The swarm intelligence algorithms, including particle swarm optimization (PSO) (Kennedy 2011), ant colony optimization (ACO) (Dorigo et al. 2006). There are some other swarm intelligent algorithms, such as, artificial bee colony(ABC) (Karaboga and Basturk 2007), bat algorithm (BA) (Yang 2010), cuckoo search (CS) (Yang and Deb 2009), gray wolf optimization (GWO) (Mirjalili et al. 2014), whale optimization algorithm (WOA) (Mirjalili and Lewis 2016), dragonfly algorithm (DA) (Mirjalili 2016), flower pollination algorithm (FPA) (Yang 2012), and Satin bowerbird optimizer (SBO) (Moosavi and Bardsiri (2017)).

The neural network is a training algorithm to simulate the principle of human brain computation. In 1996, Huang made a systematic theoretical description of pattern recognition and neural network (Huang 1996). In 1999, Huang et al. proposed linear and nonlinear feedforward network classification (Huang and Ma 1999). In 2002, Huang research radial model of probabilistic neural network model and application (Huang 1999), in 2004 Huang proposed a constructive method to find arbitrary roots of polynomial through neural network (Huang 2004). Zhao et al. used radial basis probabilistic neural networks for genetic optimization (Shang et al. 2006). In 2005, Huang proposed a radial basis probability determination by recursive orthogonal least squares algorithm (Zhao et al. 2004). In 2006, Du et al. proposed a new

✉ Yongquan Zhou
yongquanzhou@126.com

[1] College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

[2] Key Laboratory of Guangxi High Schools Complex System and Computational Intelligence, Nanning 530006, China

radial basis probabilistic neural network full structure optimization algorithm (Huang and Zhao 2005). Shang et al. use the FastICA algorithm and radial basis probabilistic neural network for palmprint recognition (Du et al. 2007). In 2007, Zheng et al. MISEP method is proposed (Shang et al. 2006). In 2008, Huang et al. studied the structural hybrid structure optimization method of radial basis probabilistic neural network (Li et al. 2009). Han et al. proposed to encode a priori information into feedforward neural network. New Constraint Learning Algorithm for Function Approximation in (Huang and Du 2008), and propose modified constraint learning algorithms to incorporate additional functional constraints into neural networks (Han and Huang 2008). In 2018, Shayanfar et al. propsed a new metaheuristic farmland fertility algorithm for solving continuous optimization problems (Shayanfar and Gharehchopogh 2018). Klein et al. proposed a meerkats-inspired algorithm for global optimization problems (Klein and dos Santos Coelho 2018). Mortazavia et al. proposed a hybrid metaheuristic Interactive search algorithm optimization algorithm (Mortazavia et al. 2018). Pierezan et al. proposed a metaheuristic coyote optimization algorithm for global optimization problems (Pierezan and Dos Santos Coelho 2018). Klein et al. proposed a swarm Intelligence cheetah based optimization algorithm (Klein et al. 2018). In 2019, Segundo, et al. using falcon optimization algorithm design of heat exchangers (de Vasconcelos Segundo et al. 2019). A new approach to fuzzy optimization based on the generalization of Bellman-Zadeh's (BZ) concept is proposed by Angelov. It consists of a parametric generalization of intersection of fuzzy sets and a generalized defuzzification method (Angelov 1994). Shadravan et al. proposed a novel nature-inspired metaheuristic sailfish optimizer algorithm for solving constrained engineering optimization problems (Shadravan et al. 2019).

The Satin bower bird optimizer algorithm is new metaheuristic optimization algorithm is proposed by Moosavi and Bardsiri 2017. It has been proven that the SBO can provide relatively competitive accuracy results. Due to this optimization algorithm is flexible, efficient and simple and successfully applied to for software development effort estimation, et al. Seyyed et al. has applied the SBO to making a link between software development effort estimation problem, ANFIS and SBO, congestion management. El-Hay et al. applied the gardener bird algorithm to Steady-state and dynamic models of solid oxide fuel cells (Han et al. 2008). Zhao et al. use the neural networks committee to resolve Human face recognition (El-Hay et al. 2018).

The complex-valued method is applied to a single individual expressing the optimization algorithm, or to optimize the expression of the weights of the neural network Satin bowerbird optimizer algorithm adopts real number coding and the individual is real number coding. The concept of plural applications in the algorithm is proposed and the introduction of diploids to represent individuals. As each complex can express two-dimensional information, this method greatly expands the dimension of expressible feature space. This paper uses complex numbers to correspond to actual variables. Although the real part and the imaginary part of the complex number are still represented by real numbers, compared with the traditional encoding method, the algorithm uses the two parameters of real part and imaginary part to represent the independent variable, which greatly improves the diversification of personalities and further increases the SBO The convergence accuracy and convergence speed of the algorithm help to improve the search efficiency of Satin bowerbird optimizer algorithm. Another important feature is the new method of solving practical problems due to the multiplicity of forms of representation and operation itself, that is, the use of complex-valued CSBO algorithm. The original satin blue optimization (SBO) has the slow convergence speed and low convergence accuracy. To improve the shortcomings of the original algorithm, complex encoding is introduced to encode the nest individuals. The real coding of the nest becomes complex coding, which increases the individual diversity of the original algorithm, increases the convergence speed and accuracy, and avoids the algorithm falling into local optimum later.

The remainder of this paper is organized as follows: Sect. 2 gives a brief introduction to SBO; Sect. 3 the complex-valued encoding satin bower bird optimizer (CSBO) Algorithm is proposed. Section 4 tests the performance of the algorithm using standard test functions and discusses the experimental results of test functions. Section 5 uses the CSBO algorithm to solve motor parameter identification problem and compare it with other algorithms. Finally, Sect. 6 contains some conclusions and future works.

2 Satin Bowerbird optimizer (SBO)

The male satin bird establishes a dedicated gazebo, called the bower that attracts the female in the bower. The bower is decorated with some decorations (flowers, feathers). These decorations are important in attracting females. It is important to note that males build their bower with innate instincts and by imitating the way other males build. According to the principle of satin bird life, SBO algorithm is divided into the following stages.

2.1 Generating a set of random bower

Initial population includes a series of positions for bowers. Each position is defined as an n dimensional vector of parameters. These values are randomly initialized so that a coincident distribution is considered between the lower and upper limit parameters. The parameters of each bower are the same as variables in the optimization problem. The combination of parameters determines the attractiveness of bowers.

2.2 Calculating the probability of each population member

The attraction probability of male bird built bowers is a key factor in whether to attraction a mate. A female satin bower bird selects a bower (built) based on its probability. This probability calculated by Eq. (1)

$$prob_j = \frac{fit_n}{\sum_{m=1}^{nb} fit_m} \tag{1}$$

where the fit_n is the fitness of the *nth* solution, and *nb* is the number of bower. In this equation, the value of fit_n is achieved by Eq. (2)

$$fit_n = \begin{cases} \frac{1}{1+f(X_n)}, f(X_n) \geq 0 \\ 1 + \left|f(X_n)\right|, f(X_n) < 0 \end{cases} \tag{2}$$

where the $f(x_n)$ is the value of cost function in *nth* position (*nth* bower).

2.3 Upgrade bower position

Elitism is one of the important characteristics of evolutionary algorithms. Elitism allows the best solution (solutions) to be retained at every stage of the optimization process. All birds usually use natural instincts to build bower, and satin bower birds is no exception. The satin bower birds use his natural instinct to build his bower and decorate it. This means that all males use materials to decorate their bower, so there is an important factor in the attraction is the experience of the male birds. In each iteration, the best individual is preserved as elite of iteration. Elites have the highest fitness values and it should be affect other positions.

In each iteration of the algorithm, the changes of every bird's bower by Eq. (3)

$$X_{nk}^{new} = X_{nk}^{old} + \lambda_k \left(\left(\frac{X_{jk} + X_{elite,k}}{2} \right) - X_{nk}^{old} \right) \tag{3}$$

where the X_n is *nth* bower or solution vector and X_{nk} is *kth* member of vector. In Eq. (3), value *j* is calculated based on probabilities derived from positions. In fact, the value *j* is calculated by the roulette wheel procedure, which means that the solution having larger probability will have more chance to be selected as X_j,X_{elite} represents the location of the elite and is stored in each loop of the algorithm. In fact, the location of the elite is a bower position; the fitness of the current iteration is the highest.

The parameter λ_k determines the attractiveness of the target bower. λ_k determines the amount of steps to be calculated for each variable. This parameter is determined by Eq. (4)

$$\lambda_k = \frac{\alpha}{1 + p_j} \tag{4}$$

In Eq. (4), α is the greatest step size and p_j is the probability obtained by Eq. (1) using the goal bower. Since the obtained probability values are between 0 and 1, the denominator of this equation is collected by 1 to avoid 0 in the denominator of the Eq. (4). As is obvious from Eq. (4), the step size is inversely proportional to the probability of target position. In other words, when probability of the target position is greater (due to the constant α), movement to that position is more carefully done. The highest step size occurs when the probability of the target position is 0 while the step size will be α. On the other hands, the lowest step size occurs when the probability of target position is 1 and the step size will be $\alpha/2$.

2.4 Mutation

When men are busy building bower on the ground, they may be attacked by other animals or are completely ignored. In many cases, strong bird will steal materials from weak bird, and even destroy their bower. Thus, at the end of each cycle of the algorithm, a random variation is applied at a certain probability. The change is in accordance with the normal distribution, as in Eq. (5)

$$X_{nk}^{new} \sim N(X_{nk}^{old}, \sigma^2) \tag{5}$$

For ease of understanding, it is more intuitive and easy to program, and Eq. (5) can be expressed as Eq. (6)

$$X_{nk}^{new} = X_{nk}^{old} + (\sigma * N(0,1)) \tag{6}$$

In Eq. (6), σ is a proportion of space width, σ is calculated by Eq. (7)

$$\sigma = z * (\text{var}_{\max} - \text{var}_{\min}) \tag{7}$$

In Eq. (7), $\text{var}_{\max}$ is upper bound, $\text{var}_{\min}$ is lower bound. There are assigned to variables, respectively. z parameter is percent of the difference between the upper and lower limit which is variable.

3 Complex-valued Satin Bower bird Optimizer (CSBO)

3.1 The complex-valued encoding method

In the biological world, the chromosomes of complex biological tissues generally use double or multi-stranded structures. Due to the two-dimensional characteristic of

Table 1 Bower chromosome model

(R_1, I_1)	(R_2, I_2)	$\cdots$	(R_n, I_n)

complex-valued encoding, this paper naturally applies it to represent diploid. In references (El-Hay et al. 2018; Xiong et al. 2015; Rashid et al. 2016; Abdel-Baset et al. 2017; Sakthivel et al. 2010) that used complex encoding to represent individuals. This article is inspired by this. In particular, a plurality of alleles can be used to describe a pair of alleles in the chromosome pair, complex real and imaginary parts are called real genes and virtual genes. For a problem with n arguments, there are n complex numbers, then corresponding to the n complex bower is recorded as follows (8):

$$X_n = R_n + iI_n \tag{8}$$

The gene of the bower can be expressed as a diploid structure and recorded as $(R_n,\ L_n)$.Where R_n and I_n represent the real and imaginary parts of the complex number, respectively. Therefore, the *nth* bird's bower can be expressed as shown in Table 1.

3.2 Initialize the complex-valued encoding population

Let the range of the function argument be interval $[\mathrm{var}_{R\min}, \mathrm{var}_{R\max}]$. Randomly generate n complex modulus ρ_n and the amplitude θ_n, the resulting modulus vector satisfies as follows:

$$R_n + iL_n = \rho_n(\cos\theta_n + i\sin\theta_n) \tag{7}$$

$$\rho_n = \left[0,\ \frac{\mathrm{var}_{R\max} - \mathrm{var}_{R\min}}{2}\right],\quad \theta_n = \left[\mathrm{var}_{\mathrm{I}min}, \mathrm{var}_{\mathrm{I}max}\right] \tag{8}$$

where the $\mathrm{var}_{R\max}$ denotes the upper bound of the real part and $\mathrm{var}_{R\min}$ denotes the lower bound of the real part. $\mathrm{var}_{\mathrm{I}max}$ denotes the upper bound of the imaginary part, which is set as 2π, $\mathrm{var}_{\mathrm{I}min}$ denotes the lower bound of the imaginary part, and is set to -2π. The n real and imaginary parts are assigned to the real and virtual genes of the bower according to produce a bird's bower.

3.3 The updating method of CSBO

3.3.1 Upgrade position formula

$$X_{Rn}^{new} + i\,X_{In}^{new} = X_{Rn}^{old} + i\,X_{In}^{old} + \lambda_n\left(\left(\frac{X_{Rj} + X_{\mathrm{Relite}}}{2}\right) - X_{Rn}^{old}\right) + i\,\lambda_n\left(\left(\frac{X_{Ij} + X_{Ielite}}{2}\right) - X_{In}^{old}\right) \tag{9}$$

where the X_{Rn} is *nth* real parts bower or solution vector.X_{Relite} Represents the real parts position of the elite. The value X_{Rj} is calculated by the roulette wheel procedure.$\mathrm{var}_{R\max}$ and $\mathrm{var}_{R\min}$ mean the upper and lower bounds of the real part. x_{In} is *nth* imaginary parts bower or solution vector.x_{Ielite} Represents the imaginary parts position of the elite. The value x_{Ij} is calculated by the roulette wheel procedure.$\mathrm{var}_{\mathrm{I}max}$ and $\mathrm{var}_{\mathrm{I}min}$ mean the upper and lower bounds of the real part.

3.3.2 Bower mutation

$$X_{Rn}^{new} + i\,X_{In}^{new} = X_{Rn}^{old} + iX_{In}^{old} + (\sigma_R * N(0,1)) + i(\sigma_I * N(0,1)) \tag{10}$$

$$\sigma_R = z * (\mathrm{var}_{R\max} - \mathrm{var}_{R\min}) \tag{11}$$

$$\sigma_I = z * (\mathrm{var}_{\mathrm{I}max} - \mathrm{var}_{\mathrm{I}min}) \tag{12}$$

In Eq. (10), the real and imaginary parts of the bird's bower position obey normal distribution N. Use Eq. (12) to make a mutation update of the bower. $\mathrm{var}_{\mathrm{I}max}$ and $\mathrm{var}_{\mathrm{I}min}$ mean the upper and lower bounds of the real part.$\mathrm{var}_{\mathrm{I}max}$ and $\mathrm{var}_{\mathrm{I}min}$ mean the upper and lower bounds of the virtual part.

3.3.3 Fitness calculation

To solve the fitness function, the complex-valued of bower must be converted into a real number, the number of modulo as the size of the real number, the symbol determined by the amplitude. The specific approach is shown in Eqs. (17) and (18)

$$\rho_n = \sqrt{X_{Rn}^2 + X_{In}^2} \tag{13}$$

$$X_n = \rho_n \mathrm{sgn}\left(\sin\left(\frac{X_{In}}{\rho_n}\right)\right) + \frac{\mathrm{var}_{R\max} + \mathrm{var}_{R\min}}{2} \tag{14}$$

where ρ_n denotes the nth multidimensional modulus, X_{Rn} and X_{Ln} denote the real and imaginary parts of the complex modulus, respectively, and X_n is the transformed real number independent variable.

3.3.4 Calculating the probability of each population member

$$prob_j = \frac{fit_n}{\sum_{m=1}^{nb} fit_m} \tag{15}$$

$$fit_n = \begin{cases} \frac{1}{1+f(X_n)}, & f(X_n) \geq 0 \\ 1 + |f(X_n)|, & f(X_n) < 0 \end{cases} \tag{16}$$

$$\lambda_n = \frac{\alpha}{1 + p_j} \tag{17}$$

where α the maximum is step size, and p_j is the probability obtained by using the objective function by Eqs. (15) and (16).

3.4 CSBO algorithm

Using the modulus to correspond to the independent variable, the argument determines the sign of the independent variable. Although the two genes with the same amplitude and different angles are sometimes the same for the output of the objective function, but two variables of the real and imaginary parts are used to represent an independent variable, which can enhance the information of the bower and increase the diversity of the bower, reduce the chance of falling into the local optimum. The pseudo code of CSBO is as follows:

Algorithm 1. CSBO pseudo-code

```
Initialize the first population of bowers randomly :Let $\rho_n = \left[0, \frac{ver_{R\max} - ver_{R\min}}{2}\right]$, $\theta_L = [-2\pi, 2\pi]$
Set $\alpha = 0.94$, $z = 0.02$, $p = 0.05$;
Fitness evaluation.Calculate the cost of bowers .(Convert to real variables to calculate by Eq.(13) and Eq.(14))
Find the best bower and assume it as elite (elite bower position include Relite and Ielite)
While the end criterion is not satisfied
    Calculate the probability of bowers using Eqs. (17) and (18)
        For every bower
            Select a bower using roulette wheel
            Calculate $\lambda_n$ using Eq. (17)
            Update the position of bower using Eqs. (9 )and (10)
        End for
    Fitness evaluation. Calculate the cost of all bowers. (Convert to real variables by Eqs.(17) and(18))
    Update elite if a bower becomes fitter than the elite(elite bower position include Relite and Ielite)
End while
Return best bower
```

The CSBO algorithm flow chart is described as follow (Fig. 1).

4 Experimental results and discussion

4.1 Simulation platform

All the calculations are used Matlab R2016a. The CPU is Intel i5 processor. The operating system is Win 10.

4.2 Compare algorithm parameter settings

We have chosen five classical optimization algorithms to compare with CSBO, including the artificial bee colony optimization algorithm (Karaboga and Basturk 2007), bat algorithm (Yang 2010), dragonfly algorithm (Mirjalili 2016), Du Fu search optimizer (CS) (Yang and Deb 2009) and the original version Satin Bowerbird Optimization algorithm (Moosavi and Bardsiri 2017). These algorithms are initially set to Table 2.

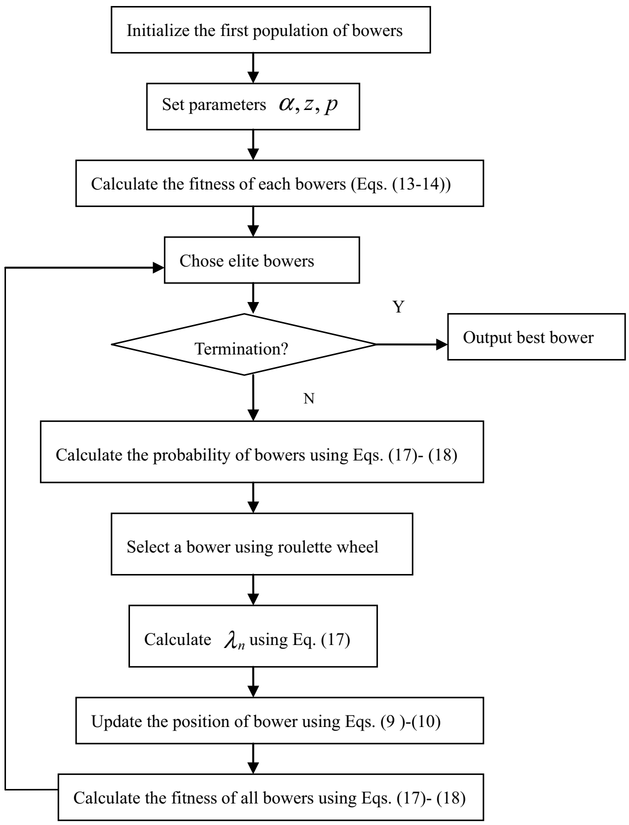
Initialize the first population of bowers
Set parameters α, z, p
Calculate the fitness of each bowers (Eqs. (13-14))
Chose elite bowers
Termination?
Y
Output best bower
N
Calculate the probability of bowers using Eqs. (17)- (18)
Select a bower using roulette wheel
Calculate λ_n using Eq. (17)
Update the position of bower using Eqs. (9)-(10)
Calculate the fitness of all bowers using Eqs. (17)- (18)

Fig. 1 The CSBO algorithm flow chart

4.3 Benchmark test functions

In this section, the CSBO algorithm is based on 20 benchmark test functions. These 20 reference functions are classical functions used by many researchers. In spite of the simplicity, we chose these test functions to compare our results with the current heuristic results. These benchmark functions are listed in Tables 3, 4, 5 and 6, where the *Dim* represents the dimension of the function, *range* is the boundary of the function search space, and $f_{\min}$ is optimal. To verify the performance of this paper, 20 classic benchmark functions were selected for simulation test. To compare each run and determine the significance of the results, the Wilcoxon rank sum test was also performed in this section. For statistical testing, CSBO selects each test function and compares it independently of other algorithms.

4.4 Benchmark functions

The results of the benchmark function test function are shown in Table 4, and the convergence of the algorithm is shown in Figs. 2, 3, 4, 5, 6, 7 and 8. The variance of this algorithm is shown in Figs. 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 and 21. The variance map is based on the results obtained by the program running 20 times in succession. The Wilcoxon rank-sum test is used to test the stability difference between the CCEO algorithm and the other six algorithms for $F_1 - F_{10}$. $p = 0.05$ was used as the boundary to distinguish between the test results: $p > 0.05$ indicates that the result is accidental and $p < 0.05$ indicates that the result is not accidental. The ranking in Table 5 is the average of 20 consecutive runs based on the CSBO algorithm. Examination of Table 5 showed that all p values were less than 0.05 (5% significance level).

4.5 Results discussion and analysis

The results of the Benchmark test functions are shown in Table 4. It can be seen that the accuracy of the CSBO algorithm exceeds the accuracy of other algorithms. It is proved that the use of complex coding to encode nest individuals effectively increases the amount of information of individuals, increases the diversity of populations, and increases the convergence accuracy. It can also be seen from the convergence graph that compared with other algorithms, the CSBO algorithm has the fastest convergence speed, benefiting from the individual's diploid idea, the individual information collector, and the convergence speed is improved. The variance diagram also shows that the CSBO algorithm has strong stability, and its variance value is more stable than other algorithms. Wilcoxon's statistical test results show that the results are statistically significant because most p values are less than 0.05. In summary, the experimental results of this paper show that the CSBO algorithm is very competitive in solving the 10 standard test functions.

Table 2 Algorithm initialization settings

ABC	MaxIteration = 1000 Population = 30 limit = 50;
BA	MaxIteration = 1000 Population = 30 $f \in [0, 1]$ $A = 0.9$, $r = 0.5$
DA	MaxIteration = 1000 Population = 30
CS	MaxIteration = 1000 Population = 30 pa = 0.25
SBO	MaxIteration = 1000 Population = 30 $\alpha = 0.94$ $z = 0.02$ mutation probability is 0.05
CSBO	MaxIteration = 1000 Population = 30 $\alpha = 0.94$ $z = 0.02$ mutation probability is 0.05

Table 3 Benchmark functions

Benchmark test functions	Dim	Range	Optimum
$F_1 = \sum_{i=1}^{D} x_i^2$	30	[− 100, 100]	0
$F_2 = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	30	[− 10, 10]	0
$F_3 = \sum_{i=1}^{D} (\sum_{j=1}^{D} x_j)^2$	30	[− 100, 100]	0
$F_4 = \max\{\lvert x_i \rvert, 1 \le i \le D\}$	30	[− 100, 100]	0
$F_5 = \sum_{i=1}^{D-1} [100\,(x_{i+1} - x_i^2)^2] + (x_i - 1)^2$	30	[− 30, 30]	0
$F_6 = \sum_{i=1}^{n} x_i^4 + random(0, 1)$	30	[− 5.12, 5.12]	0
$F_7 = \sum_{i=1}^{n} x_i \sin(x_i) + 0.1\, x_i$	30	[− 10, 10]	− 418*Dim
$F_8 = -20 \exp\left(-0.2\sqrt{\frac{1}{n}} \sum_{i=1}^{n} x_i^2 - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos 2\pi\, x_i\right)\right) + 20 + e$	30	[− 32, 32]	0
$F_9 = \frac{1}{4000} \sum_{i=1}^{n} (x_i^2) - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[− 600, 600]	0
$F_{10} = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 + 6)^2 + 10\left(1 - \frac{1}{8\pi}\right) \cos x_1 + 10$	2	[− 5, 5]	0.398

5 Identification problem

Induction motor parameters can not be measured directly. Therefore, the usual special identification methods are needed to estimate them. In this way, the behavior of the induction motor is simulated by the equivalent circuit. In general, they allow estimating the proper relationship of the motor parameters. In the process of identification, the parameter estimation is transformed into a multi-dimensional optimization problem, and the internal parameters of the induction motor are taken as decision variables. The approximate circuit model as shown below (Fig. 22).

Figure 22 illustrates the approximate circuit model. In the approximate circuit model, identification tasks can be expressed as the following optimization problems:

$$\text{Minimize } J(x),\ x = (R_1, R_2, X_1) \in R^4 \tag{18}$$

$$\text{Subject to:}\quad 0 \le R_1 \le 1,\ 0 \le R_2 \le 1,\ 0 \le X_1 \le 10,\ 0 < s < 1 \tag{19}$$

where

$$J(x) = (f_1(x))^2 + (f_2(x))^2 + (f_3(x))^2 \tag{20}$$

$$f_1(x) = \frac{\frac{K_t R_2}{s\left[\left(R_1 + {R_2}/{S}\right)^2 + X_1^2\right]} - T_{fl}}{T_{fl}} \tag{21}$$

$$f_2(x) = \frac{\frac{K_t R_2}{(R_1 + R_2)^2 + X_1^2} - T_{lr}}{T_{lr}} \tag{22}$$

$$f_3(x) = \frac{\frac{K_t}{2[R_1 + \sqrt{R_1^2 + X_1^2}]} - T_{\max}}{T_{\max}} \tag{23}$$

$$K_t = \frac{V_{ph}^2}{\omega_s} \tag{24}$$

The approximate circuit model uses the manufacturer data starting torque (T_{lr}), maximum torque ($T_{\max}$), full load torque (T_{f1}) and motor slip (s) to calculate the stator resistance (R_1) rotor resistance (R_2) and tator leakage reactance (X_1).

5.1 Initial parameter setting

We used two modes of induction motor to test the performance of the algorithm; the specific parameters are set as follows.

Table 4 Benchmark functions test results

Algorithms							
Functions	ABC	BA	DA	CS	SBO	CSBO	RANK
*F*1							
Best	9.8187E−08	1.1644E+04	4.0058E+02	5.0591E−02	5.4913E−03	1.3744E−81	1
Worst	5.3195E−06	2.4211E+04	3.1676E+03	1.1623E−01	2.4441E−02	6.7195E−68	
Mean	1.0789E−06	1.9021E+04	1.1439E+03	8.1576E−02	1.3218E−02	3.4266E−69	
Std.	1.2410E−06	3.4843E+03	6.7231E+02	1.6153E−02	4.3895E−03	1.5012E−68	
*F*2							
Best	1.1704E−05	1.3855E+02	2.7430E−01	1.5406E+00	2.0752E−02	8.3425E−26	1
Worst	1.0918E−05	5.6906E+08	2.5830E+01	3.8177E+00	5.3621E−02	3.8945E−24	
Mean	4.1282E−05	5.4628E+07	1.3649E+01	2.5392E+00	3.3815E−02	1.2284E−24	
Std.	2.4357E−05	1.3685E+08	6.3935E+00	6.4821E+00	7.6298E−03	1.1762E−24	
*F*3							
Best	1.3164E+04	1.0482E+04	2.3838E+03	1.2711E+00	1.4904E+02	1.9630E−73	1
Worst	2.9339E+05	6.2780E+04	3.1525E+04	4.0443E+00	5.3887E+02	1.7597E−57	
Mean	2.2475E+05	2.5207E+04	1.2756E+04	2.3825E+00	3.2879E+02	8.7990E−59	
Std.	4.1385E+04	1.1684E+04	7.7515E+03	7.8140E−01	1.2330E+02	3.9349E−58	
*F*4							
Best	5.6190E+01	5.4338E+01	1.1056E+01	3.2908E−01	3.6547E−01	1.2019E−37	1
Worst	7.5403E+01	6.7997E+01	4.0130E+01	9.8635E−01	7.3835E−01	7.8517E−33	
Mean	6.7805E+01	6.3147E+01	2.0258E+01	6.4784E−01	4.4821E−01	9.8410E−34	
Std.	4.4291E+00	4.3109E+00	6.7343E+00	1.7668E−01	8.5230E−02	2.1174E−33	
*F*5							
Best	1.1148E+01	2.6073E+01	7.7101E+04	3.4042E+01	2.3319E+01	2.8762E+01	1
Worst	8.7054E+01	2.4856E+02	7.8762E+05	1.1078E+02	6.2739E+02	2.8902E+01	
Mean	4.1911E+01	4.7610E+01	1.5845E+05	4.3729E+01	1.3161E+02	2.8802E+01	
Std.	2.2016E+01	5.4710E+01	1.7404E+05	1.6509E+01	1.5703E+02	3.5176E−02	
*F*6							
Best	3.9217E−01	1.2646E−02	7.5690E−02	1.8055E−01	3.9972E−02	1.3247E−04	1
Worst	1.2012E+00	4.5266E−02	7.2206E−01	6.2109E−01	1.0722E−01	7.4894E−04	
Mean	8.2435E−01	2.8219E−02	3.1307E−01	3.3770E−01	6.4263E−02	4.8193E−04	
Std.	1.9818E−01	7.8970E−02	1.7795E−01	1.3038E−01	16707E−02	1.3247E−04	
*F*7							
Best	−1.231E+04	−7.679E+03	−6.481E+03	−1.029E+02	−8.284E+03	−1.575E+05	1
Worst	−1.089E+04	−4.180E+03	−4.054E+03	−8.847E+01	−4.080E+03	−9.225E+04	
Mean	−1.166E+04	−5.593E+03	−5.561E+03	−9.596E+01	−5.653E+03	−1.356E+05	
Std.	3.9349E+02	8.8323E+02	5.5994E+02	3.9272E+00	9.1834E+02	1.5855E+04	
*F*8							
Best	5.1376E−05	1.8410E+01	6.0518E+00	2.5845E+00	2.4987E−02	8.8817E−16	1
Worst	3.1280E−03	1.9903E+01	1.1557E+01	3.6118E+00	1.0035E−01	7.9936E−16	
Mean	7.8133E−04	1.9022E+01	8.6257E+00	3.1210E+00	4.0233E−02	3.1974E−16	
Std.	8.0517E−04	3.3607E−01	1.3325E+00	2.9960E−01	2.0166E−02	2.0859E−15	
*F*9							
Best	4.1719E−07	3.9711E+02	4.9193E+00	3.7467E−03	1.3612E−02	0	1
Worst	4.2362E−02	6.1014E+02	2.2301E+01	9.8400E−03	9.7033E−01	0	
Mean	1.0850E−02	5.2028E+02	1.1843E+01	6.6411E−03	2.4836E−01	0	
Std.	1.5592E−02	5.2568E+01	4.9985E+00	1.5106E−03	2.5332E−01	0	

Table 4 (continued)

Algorithms							
Functions	ABC	BA	DA	CS	SBO	CSBO	RANK
*F*10							
Best	– 1.031E + 00	– 1.031E + 00	– 1.031E + 00	– 1.031E + 00	– 1.031E + 00	– 1.031E + 00	1
Worst	– 1.031E + 00	– 0.215E + 00	– 1.031E + 00	– 1.031E + 00	– 1.031E + 00	– 1.031E + 00	
Mean	– 1.031E + 00	– 0.950E + 00	– 1.031E + 00	– 1.031E + 00	– 1.031E + 00	– 1.031E + 00	
Std.	7.1558E – 04	0.2512E + 00	7.5208E – 05	2.4296E – 05	1.3875E – 05	7.5245E – 05	

Table 5 *p* Value test results

Algorithms					
Functions	DA vs CSBO	ABC vs CSBO	BA vs CSBO	CS vs CSBO	SBO vs CSBO
*F*1	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08
*F*2	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08
*F*3	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08
*F*4	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08
*F*5	6.7956E – 08	6.7956E – 08	6.3592E – 01	6.7956E – 08	1.6098E – 04
*F*6	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08
*F*7	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08	6.7956E – 08
*F*8	4.3719E – 08	4.3719E – 08	4.3719E – 08	4.3719E – 08	4.3719E – 08
*F*9	8.0065E – 09	8.0065E – 09	8.0065E – 09	8.0065E – 09	8.0065E – 09
*F*10	1.2921E – 04	4.3719E – 08	1.5997E – 05	3.9417E – 01	1.0645E – 07

Table 6 Manufacturer data of the motors used in the experiments

Modes		
Parameters	Mode 1	Mode 2
Power (*HP*)	5	5
Voltage (*V*)	400	400
Current (*A*)	8	45
Frequency (*Hz*)	50	50
No. poles	4	4
Full load slip (*s*)	0.07	0.09
Starting torque (T_{str})	15	260
Max. torque ($T_{\max}$)	42	370
Stator current	22	180
Full load torque (T_{f1})	25	190

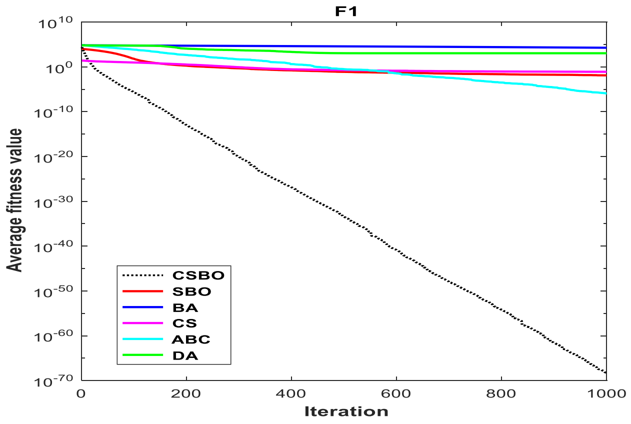

Fig. 2 Convergence curves of function F_1

5.2 Experimental results

We run continuously for 30 times by comparing ABC, DA, BA, PSO, GSA, SBO and CSBO with the Best, Worst, and Average and Variance values of the results. The experimental results are shown in the Table 7. The convergence graph is shown in Figs. 23 and 24, and the variance graph is shown in Figs. 25 and 26. The results of the *p* value in Table 8.

As can be seen from Table 7, the CSBO algorithm runs continuously for 30 times in each of the two modes, and its average is ranked first, which fully verifies the superiority of the CSBO in handling induction motors. From the Figs. 23 and 24, it can be seen that CSBO has the fastest convergence speed and convergence accuracy. Figures 25 and 26 also show that the CSBO algorithm results are relatively stable.

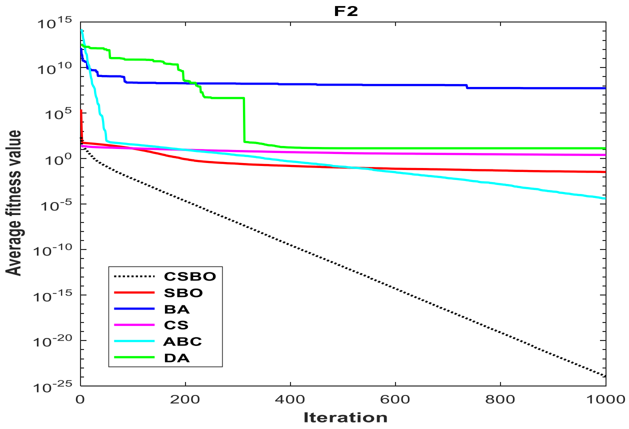

Fig. 3 Convergence curves of function F_2

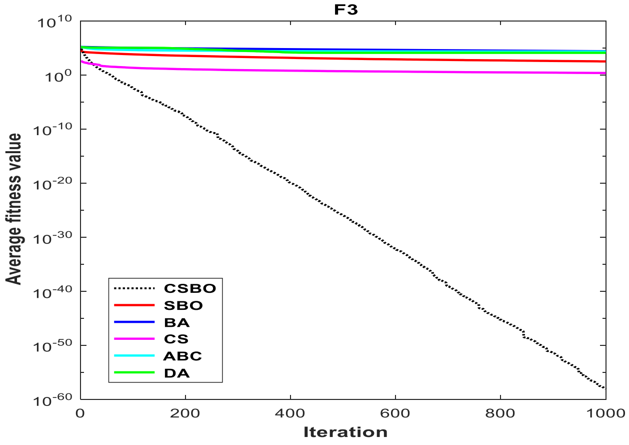

Fig. 4 Convergence curves of function F_3

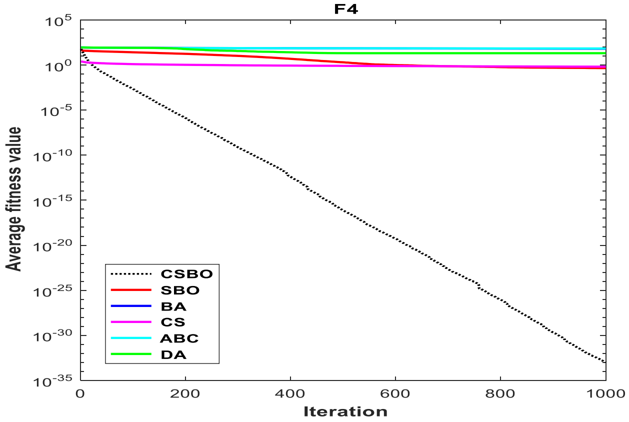

Fig. 5 Convergence curves of function F_4

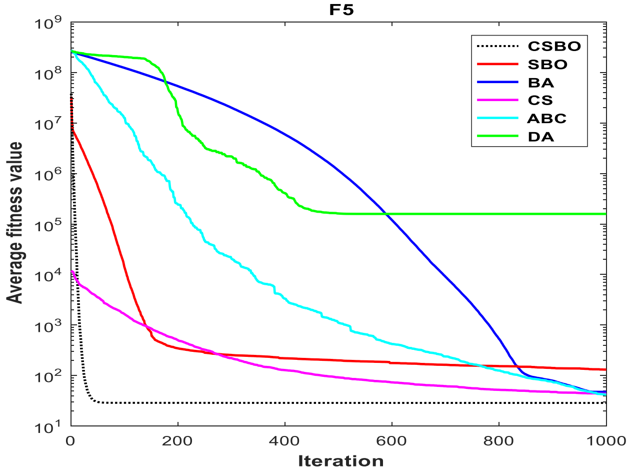

Fig. 6 Convergence curves of function F_5

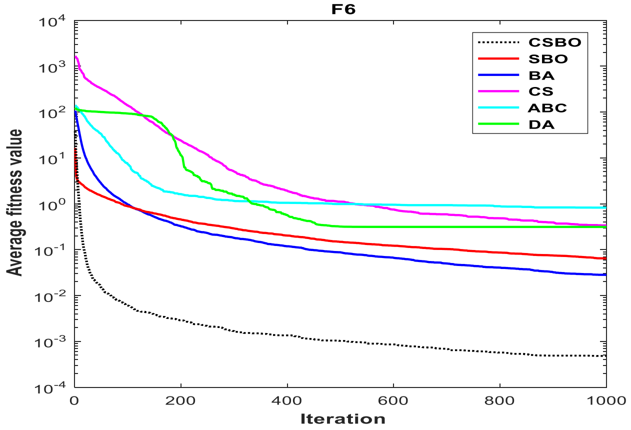

Fig. 7 Convergence curves of function F_6

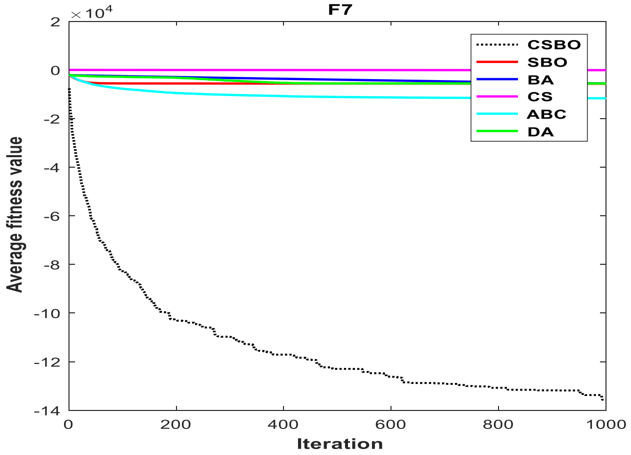

Fig. 8 Convergence curves of function F_7

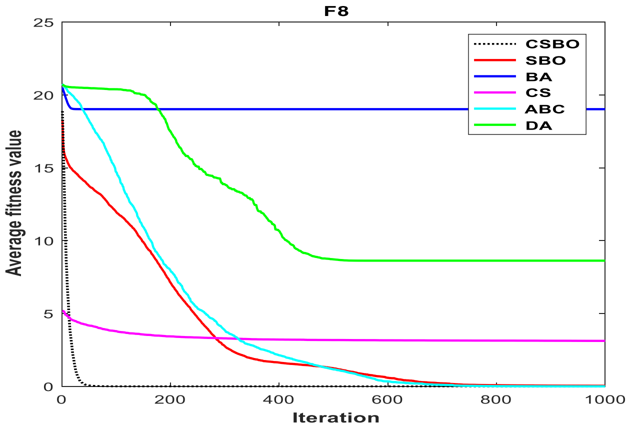

Fig. 9 Convergence curves of function F_8

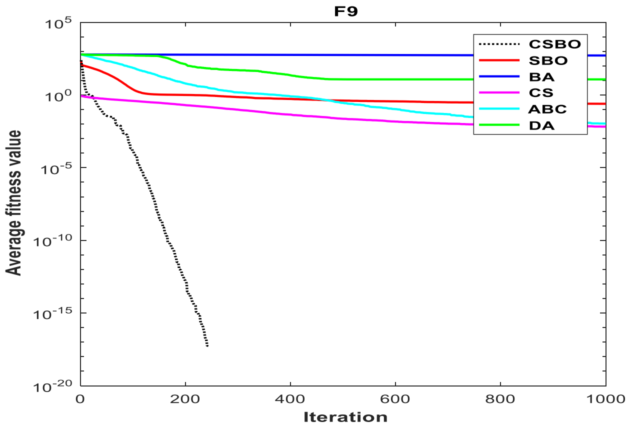

Fig. 10 Convergence curves of function F_9

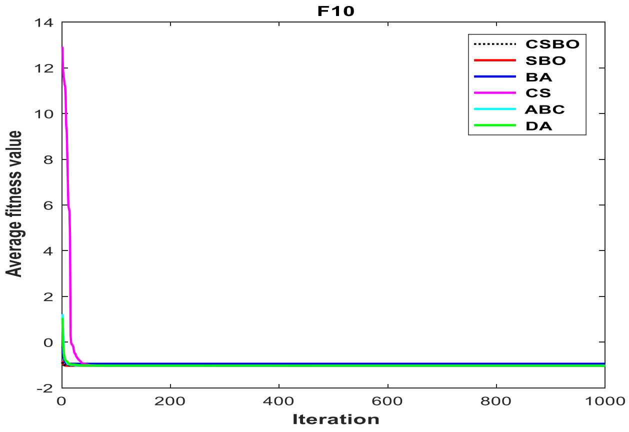

Fig. 11 Convergence curves of function F_{10}

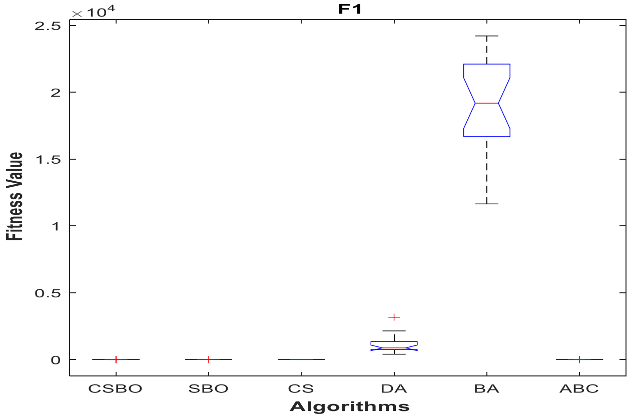

Fig. 12 ANOVA tests of the function F_1

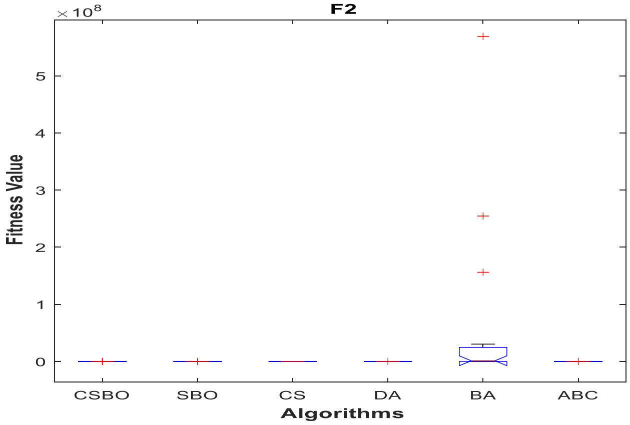

Fig. 13 ANOVA tests of the function F_2

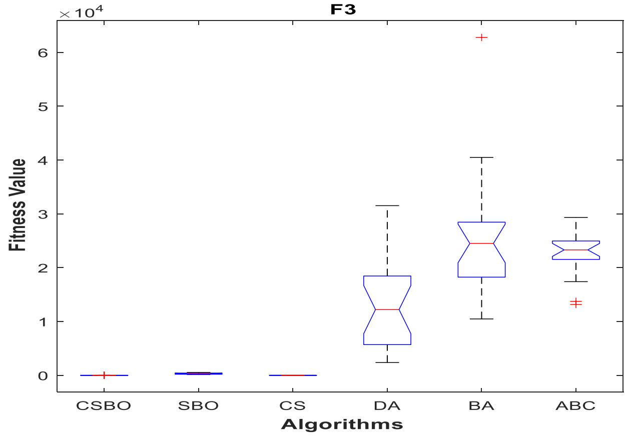

Fig. 14 ANOVA tests of the function F_3

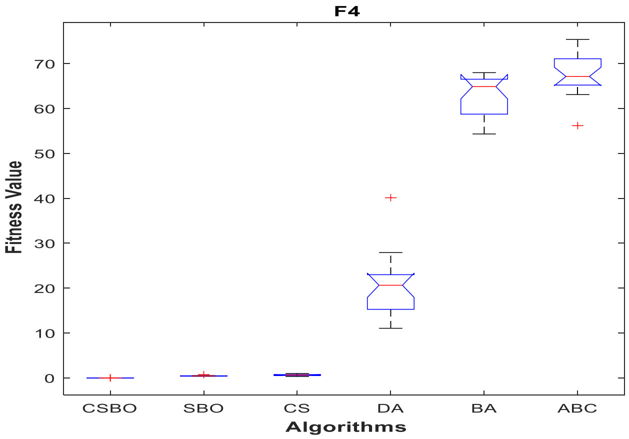

Fig. 15 ANOVA tests of the function F_4

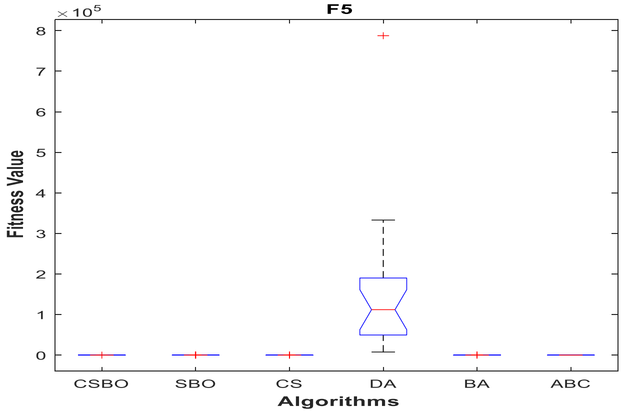

Fig. 16 ANOVA tests of the function F_5

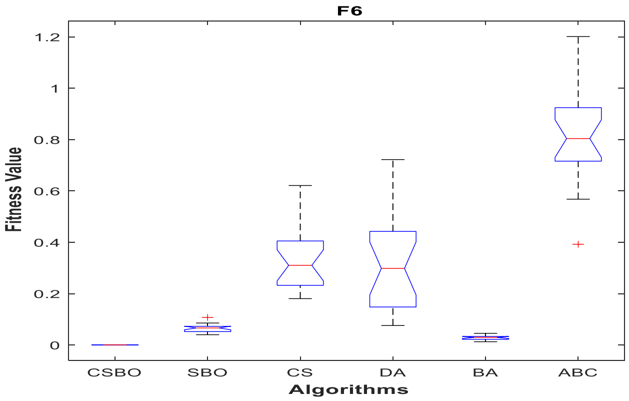

Fig. 17 ANOVA tests of the function F_6

5.3 Statistical analysis

In order to statistically analyze the results, a nonparametric test called Wilcoxon analysis was performed. It allows

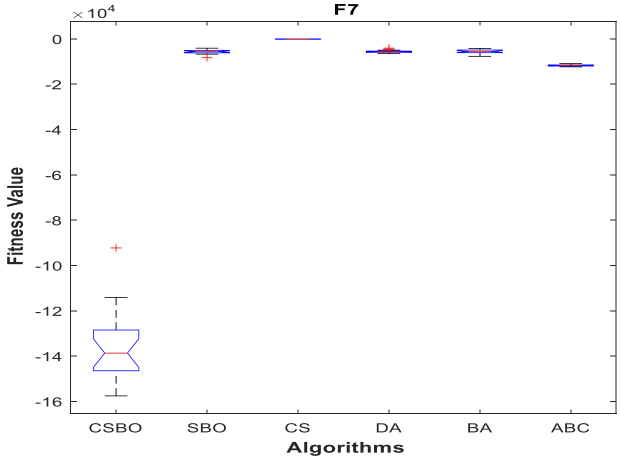

Fig. 18 ANOVA tests of the function F_7

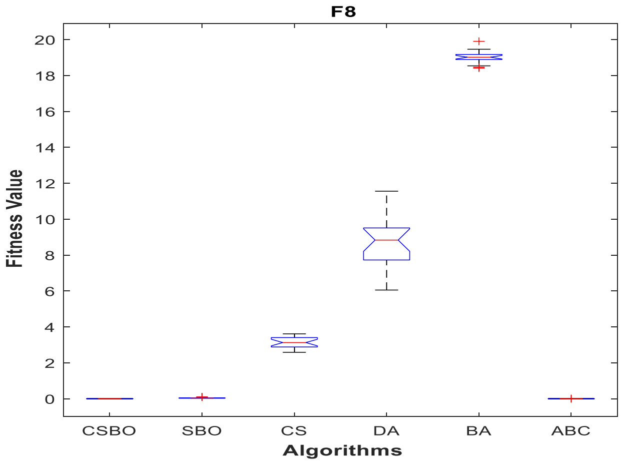

Fig. 19 ANOVA tests of the function F_8

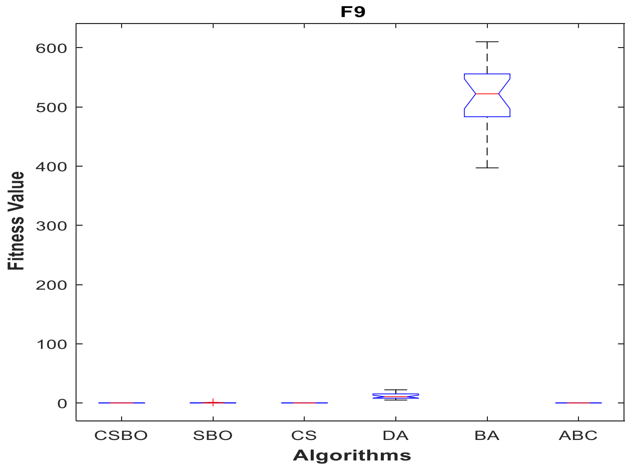

Fig. 20 ANOVA tests of the function F_9

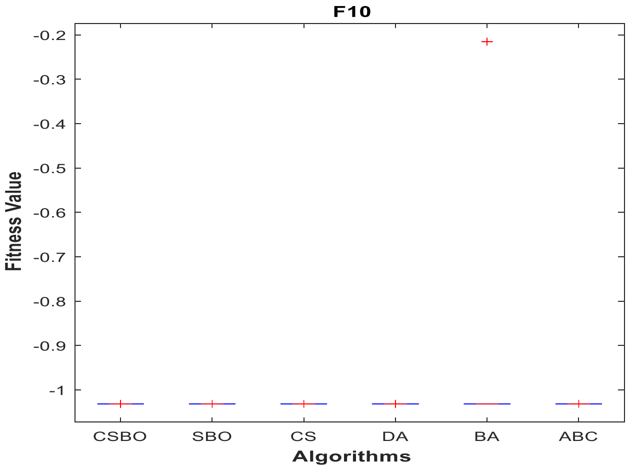

Fig. 21 ANOVA tests of the function F_{10}

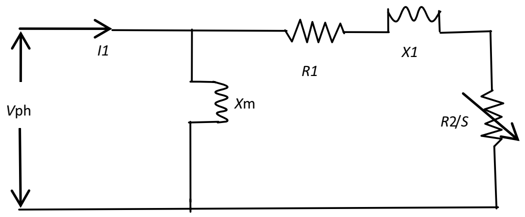

Fig. 22 Approximate circuit model

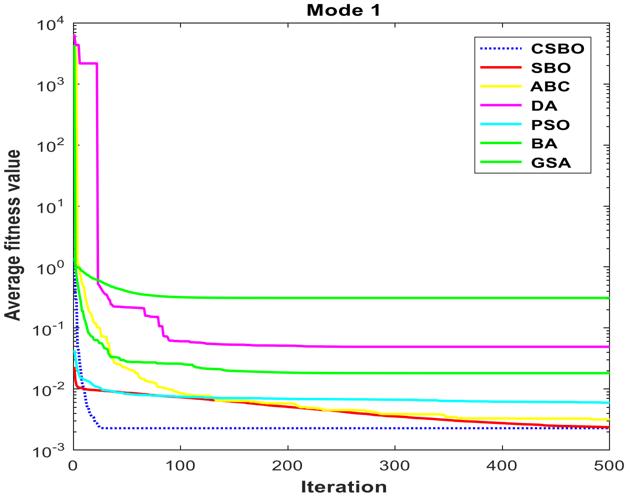

Fig. 23 Convergence curves of the mode1

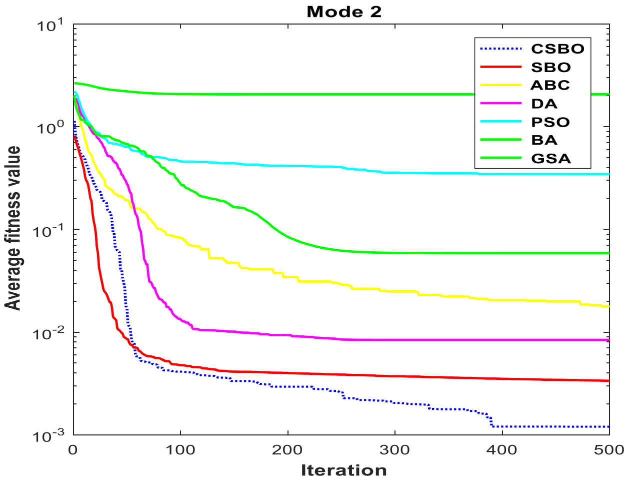

Fig. 24 Convergence curves of the mode2

assessing the differences between the two related methods. Table 8 reports the p values generated by the Wilcoxon analysis for pairwise comparisons between algorithms. In this case, there are six groups: CSBO vs SBO, CSBO vs ABC, CSBO vs DA.CSBO vs PSO, CSBO vs BA, CSBO vs GSA. Examination of Table 8 showed that all p values were less than 0.05 (5% significance level). This fact provides strong evidence against the null hypothesis, suggesting that the proposed method presents statistically better results than other algorithms.

Table 7 Induction motor experimental results

Algorithms								
Modes	ABC	DA	BA	PSO	GSA	SBO	CSBO	RANK
Mode1								
Best	0.00228	0.00227	0.00238	0.00029	0.00228	0.00227	0.00227	1
Worst	0.00733	0.06321	0.97989	0.01583	0.05449	0.00342	0.00227	
Mean	0.00309	0.00837	0.31114	0.00653	0.01818	0.00236	0.00227	
Std.	0.00117	0.01927	0.29201	0.00386	0.01602	0.00021	2.813E−13	
Mode2								
Best	0.00273	1.541E−06	0.00524	0.02726	2.176E−18	8.775E−06	0.00031	1
Worst	0.05150	0.01438	2.64491	0.91981	0.61605	0.01340	0.00197	
Mean	0.02335	0.00840	2.06631	0.38799	0.05863	0.00336	0.00120	
Std.	0.01530	0.00563	0.55145	0.28239	0.15857	0.00326	0.00061	

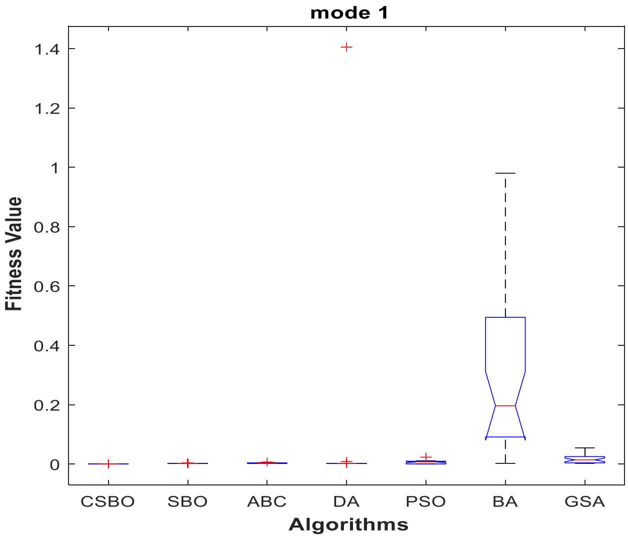

Fig. 25 Anova tests of the mode1

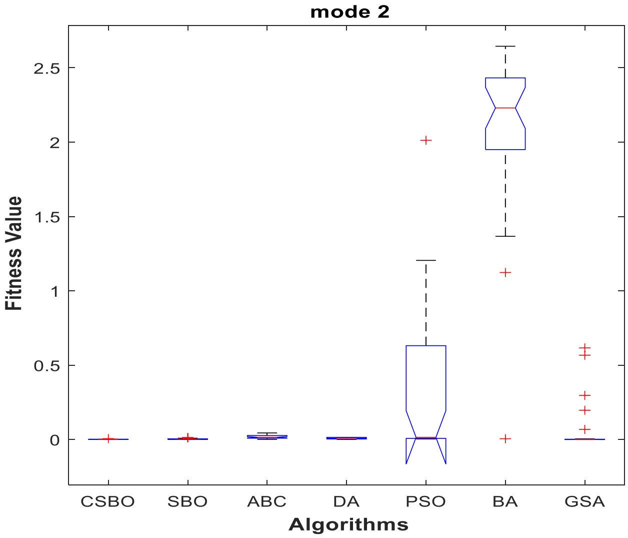

Fig. 26 Anova tests of the mode2

Table 8 *p* Value test results

Modes		
Algorithms	Mode 1	Mode 2
CSBO vs SBO	3.01985E – 11	2.70863E – 02
CSBO vs ABC	3.01985E – 11	1.69472E – 09
CSBO vs DA	3.01418E – 11	5.85867E – 06
CSBO vs PSO	1.68131E – 04	3.68972E – 11
CSBO vs BA	3.01985E – 11	3.33838E – 11
CSBO vs GSA	3.01985E – 11	3.91670E – 02

6 Conclusion and future work

This paper a complex-valued encoding satin bowerbird optimization algorithm (CSBO) is proposed. This algorithm introduces the idea of complex-valued coding and finds the optimal one by updating the real and imaginary parts value. The proposed CSBO optimization algorithm is compared with real-valued SBO and other optimization algorithms using 20 standard test functions including unimodal and multimodal functions, induction motor parameter identification and *p* value test. Results show that the proposed CSBO can significantly improve the performance. Future research focuses on the complex-valued satin bowerbird optimizer (CSBO) algorithm is used in combinatorial optimization, engineering optimization and other applied fields.

Acknowledgements This work is supported by National Science Foundation of China under Grants No. 61563008, and by Project of Guangxi Natural Science Foundation under Grant No. 2018GXNSFAA138146.

References

Abdel-Baset M, Wu H, Zhou Y (2017) A complex encoding flower pollination algorithm for constrained engineering optimisation problems. Int J Math Model Numer Optim 8(2):108–126

Angelov P (1994) A generalized approach to fuzzy optimization. Int J Intell Syst 9(3):261–268

de Vasconcelos Segundo EH, Mariani VC, dos SantosCoelho L (2019) Design of heat exchangers using Falcon Optimization Algorithm. Appl Thermal Eng 156:119–144

Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39

Du J-X, Huang DS, Wang X-F, Gu X (2007) Shape recognition based on neural networks trained by differential evolution algorithm. Neurocomputing 70(4–6):896–903

El-Hay EA, El-Hameed MA, El-Fergany AA (2018) Steady-state and dynamic models of solid oxide fuel cells based on Satin Bowerbird Optimizer. Int J Hydrogen Energy 43(31):14751–14761

Fister I Jr, Yang X-S, Fister I, Brest J, Fister D (2013) A brief review of nature-inspired algorithms for optimization. Elektrotehniski Vestnik 80(3):1–7

Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Mach Learn 3(2):95–99

Han F, Huang DS (2008) A new constrained learning algorithm for function approximation by encoding a priori information into feedforward neural networks. Neural Comput Appl 17(5–6):433–439

Han F, Ling Q-H, Huang DS (2008) Modified constrained learning algorithms incorporating additional functional constraints into neural networks. Inf Sci 178(3):907–919

Huang DS (1996) Systematic theory of neural networks for pattern recognition (in Chinese). Publishing House of Electronic Industry of China, Beijing

Huang DS (1999) Radial basis probabilistic neural networks: model and application. Int J Pattern Recognit Artif Intell 13(7):1083–1101

Huang DS (2004) A constructive approach for finding arbitrary roots of polynomials by neural networks. IEEE Trans Neural Networks 15(2):477–491

Huang DS, Du J-X (2008) A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. IEEE Trans Neural Netw 19(12):2099–2115

Huang DS, Ma SD (1999) Linear and nonlinear feedforward neural network classifiers: a comprehensive understanding. J Intell Syst 9(1):1–38

Huang DS, Zhao WB (2005) Determining the centers of radial basis probabilistic neural networks by recursive orthogonal least square algorithms. Appl Math Comput 162(1):461–473

Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471

Kennedy J (2011) Particle swarm optimization. Encyclopedia of machine learning. Springer, New York, pp 760–766

Klein CE, dos Santos Coelho L (2018) Meerkats-inspired algorithm for global optimization problems. ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25–27 April 2018

Klein CE, Mariani VC, dos Santos Coelho L (2018) Cheetah based optimization algorithm: a novel swarm intelligence paradigm. ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25–27 April 2018

Li B, Wang C, Huang DS (2009) Supe rvised feature extraction based on orthogonal discriminant projection. Neurocomputing 73(1–3):191–196

Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073

Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

Moosavi SHS, Bardsiri VK (2017) Satin bowerbird optimizer: a new optimization algorithm to optimize ANFIS for software development effort estimation. Eng Appl Artif Intell 60:1–15

Mortazavia Ali, Toğanb Vedat, Nuhoğluc Ayhan (2018) Interactive search algorithm: a new hybrid metaheuristic optimization algorithm. Eng Appl Artif Intell 71:275–292

Pierezan J, Dos Santos Coelho L (2018) Coyote optimization algorithm: a new metaheuristic for global optimization problems. IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018

Rainer S, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359

Rashid S, Saraswathi S, Kloczkowski A et al (2016) Protein secondary structure prediction using a small training set (compact model) combined with a complex-valued neural network approach. BMC Bioinform 17(1):362

Sakthivel VP, Bhuvaneswari R, Subramanian S (2010) Multi-objective parameter estimation of induction motor using particle swarm optimization. Eng Appl Artif Intell 23(3):302–312

Shadravan S, Naji HR, Bardsiri VK (2019) The Sailfish Optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Eng Appl Artif Intell 80:20–34

Shang L, Huang DS, Du J-X, Zheng C-H (2006) Palmprint recognition using FastICA algorithm and radial basis probabilistic neural network. Neurocomputing 69(13–15):1782–1786

Shayanfar H, Gharehchopogh FS (2018) Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems. Appl Soft Comput 71:728–746

Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713

Xiong T, Bao Y, Hu Z et al (2015) Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms. Inf Sci 305:77–92

Yang X-S (2010) A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, Berlin, pp 65–74

Yang X-S (2012) Flower pollination algorithm for global optimization. UCNC, Barasat

Yang X-S, Deb S (2009) Cuckoo search via Lévy flights. Nature & biologically inspired computing, 2009. NaBIC 2009. World Congress on. IEEE

Zhao WB, Huang DS, Du J-Y, Wang L-M (2004) Genetic optimization of radial basis probabilistic neural networks. Int J Pattern Recognit Artif Intell 18(8):1473–1500