Knowledge-Based Systems 212 (2021) 106546

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# Backtracking search algorithm with specular reflection learning for global optimization

Yiying Zhang

*School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, PR China*

ARTICLE INFO

*Article history:*
Received 18 June 2020
Received in revised form 8 October 2020
Accepted 18 October 2020
Available online 1 November 2020

*Keywords:*
Opposition-based learning
Specular reflection learning
Backtracking search algorithm
Global optimization

ABSTRACT

Benefiting from population, randomness and simple structures, metaheuristic methods show excellent performance for solving global optimization problems. However, in some cases, in order to get promising solutions, the existing metaheuristic methods usually need to be modified. This work reports a new technique, called specular reflection learning, for improving the optimization performance of metaheuristic methods. Specular reflection learning is motivated by specular reflection phenomenon in physics. Note that, there is a close relationship between opposition-based learning and specular reflection learning. Opposition-based learning can be seen as a special case of specular reflection learning. In order to investigate the effectiveness of specular reflection learning, specular reflection learning is employed to improve backtracking search algorithm (BSA). The performance of the proposed backtracking search algorithm with specular reflection learning is evaluated by 88 test functions extracted from the well-known CEC 2013, CEC 2014 and CEC 2017 test suites, and two constrained engineering design problems. Experimental results confirm that specular reflection learning is a more effective technique for improving BSA compared with opposition-based learning, which establishes the foundation for the applications of specular reflection learning on other metaheuristics. In addition, the source code of this work can be found from https://www.mathworks.com/matlabcentral/fileexchange/79030-bsa_srl.



## 1. Introduction

In the past twenty years, metaheuristic methods have been successfully used to solve a lot of optimization problems of various fields. Tan et al. [1] used enhanced particle swarm optimization for intelligent skin cancer detection. Elaziz et al. [2] presented a hybrid algorithm based on moth search algorithm and differential evolution for task scheduling in cloud computing. Zhou et al. [3] introduced symbiotic organism search algorithm for automatic data clustering. Shen et al. [4] adopted fruit fly optimization for medical data classification. Chen et al. [5] proposed enhanced bacterial foraging optimization for training kernel extreme learning machine. Feng et al. [6] reported a modified sine cosine algorithm for multiple hydropower reservoirs operation. The superiority of metaheuristic methods for optimization problems mainly comes from the following advantages:

- Population. For one metaheuristic method, its population consists of some individuals. The position of each individual denotes a potential solution to the optimization problem. During the optimization process, the position of each individual moves frequently based on some defined rules to search better solutions. Thus, metaheuristic methods also can be seen as population-based optimization methods. Note that, the individuals of the population can perform exploration or exploitation for the given search space simultaneously, which can significantly increase the chance of the population to find better solutions.
- Randomness. Randomness of metaheuristic methods is mainly shown in the following two aspects. On one hand, the individuals of the population are randomly initialized, which can weaken the relationship between initialization population and the obtained optimal solution. On the other hand, metaheuristic methods usually define some simple rules to simulate natural phenomena. Here, it should be pointed out that natural phenomena are often accompanied by randomness. Thus, in order to more accurately describe natural phenomena, random numbers, such as random numbers with uniform distribution, are usually introduced to the defined rules, which can further weaken the relationship the initialization population and the obtained optimal solution.
- Simplicity. Metaheuristic methods do not rely on complex math principles and only use some simple rules to simulate natural phenomena. In addition, metaheuristics usually have simple structures. Thus, metaheuristic methods are usually

*E-mail address:* zhangyiying@tju.edu.cn.

https://doi.org/10.1016/j.knosys.2020.106546
0950-7051/

easy to implement and have great potential to be used for different types of optimization problems.

A major driver for promoting the development of metaheuristic methods is "No-Free-Lunch" (NFL) theorem [7]. According to NFL theorem, it is not possible for one algorithm to beat other algorithms for all optimization problems [8,9]. The main reason of supporting NFL theorem is that each optimization problem usually has its features, such as the number of variables, the constrained conditions and the characteristic of objective function. Thus, with more and more optimization problems with different features found in the real world, it is very essential for researchers to do more efforts for developing metaheuristic methods. At present, the development of metaheuristic methods has two branches, i.e. developing metaheuristic methods by new inspired sources and modifying the reported metaheuristic methods by some improved techniques.

Backtracking search algorithm (BSA) is a recently reported metaheuristic method [10]. BSA has the following features: (1) it only has a single control parameter called mix rate; (2) it has a very simple structure and is easy to implement; (3) the search direction of BSA is guided by the differential vectors between historical population and current population. Given these features, BSA has been used to solve various types of optimization problems, such as modeling beach realignment [11], parameter identification of photovoltaic models [12], short-term electricity price forecasting [13], and cast heat treatment charge plan problem [14]. However, the strategy of updating population in BSA highly depends on the differential vectors between historical population and current population, which cannot take advantage of population information. With the increasing of iterations, the differences between historical population and current population become small gradually, which will weaken the exploration of BSA. Thus, once BSA gets trapped in a local minimum, it is very difficult for BSA to escape from it.

As an improved technique for metaheuristic methods, opposition-based learning has been widely used for many metaheuristic methods, such as opposition-based differential evolution [15], grasshopper optimization algorithm based on opposition-based learning [16], crow search algorithm based on opposition-based learning [17], opposition-based sine cosine approach with local search [18], opposition-based gray-wolf optimization algorithm [19], dynamic opposite learning enhanced teaching–learning-based optimization [20], opposition-based emperor penguin optimization [21] and opposition-based moth flame optimization [22]. The basic idea of opposition-based learning can be described as follows. The opposite solution of one solution may be closer to the optimal solution. Thus it is possible that a fitter solution can be selected by simultaneously checking one solution and its opposite solution to accelerate convergence speed of metaheuristics [15].

Motivated by opposition-based learning and specular reflection phenomena in physics, this work presents a new variant of BSA, called backtracking search algorithm with specular reflection learning (BSA_SRL), for global optimization. In the proposed method, a new technique for improving metaheuristic methods is introduced, which can be called specular reflection learning inspired by specular reflection phenomena. According to opposition-based learning, there is a one-to-one correspondence between one solution and its opposite solution. However, for specular reflection learning, there is a corresponding relationship between one solution and one neighborhood of its opposite solution. That is, a fitter solution can be selected by simultaneously checking one solution and another solution from this neighborhood. Obviously, opposition-based learning can be seen as a special case of specular reflection learning. In addition, like opposition-based learning, when used to metaheuristic methods, specular reflection learning is also as a mutation operator with jumping rate. The main contributions of this work can be summarized as follows.

(1) Specular reflection learning is presented for improving metaheuristic methods.
(2) A novel variant of BSA, called BSA_SRL, is proposed for global optimization.
(3) The performance of BSA_SRL is tested by CEC 2013, CEC 2014 and CEC 2017 test suites.
(4) BSA_SRL is employed for solving constrained engineering optimization problems.

The rest of this paper is organized as follows. Section 2 introduces two basic components, i.e. backtracking search algorithm and opposition-based learning. Section 3 shows the built specular reflection learning model and the proposed BSA_SRL. The performance of BSA_SRL for unconstrained numerical optimization problems including CEC 2013, CEC 2014 and CEC 2017, is checked in Section 4. Section 5 gives the experimental results of BSA_SRL for engineering design problems. In Section 6, managerial insights are presented. Lastly, conclusion and future directions are made in Section 7.

## 2. Basic knowledge

### 2.1. Opposition-based learning

Opposition-based learning is based on opposite number and opposite point [15].

- Opposite number. Assume that the range of a real number $x$ is from $V_{\mathrm{L}}$ to $V_{\mathrm{U}}$, i.e. $x \in [V_{\mathrm{L}}, V_{\mathrm{U}}]$. The opposite number $\breve{x}$ of $x$ can be defined by

$$\breve{x} = V_{\mathrm{L}} + V_{\mathrm{U}} - x \tag{1}$$

- Opposite point. Assume that $P$ is a point in $D$-dimensional space, i.e. $P = (x_1, x_2, \ldots, x_D)$, where $x_i$ is a real number and $x_i \in \left[V_{\mathrm{L},i}, V_{\mathrm{U},i}\right], \forall i = \{1, 2, \ldots, D\}$. The opposite point $\breve{P} = (\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_D)$ of the point $P$ can be defined by

$$\breve{x}_i = V_{\mathrm{L},i} + V_{\mathrm{U},i} - x_i, i = 1, 2, \ldots, D \tag{2}$$

As mentioned previously, for one metaheuristic method, each individual of its population denotes a candidate solution. In fact, each individual also can be seen a point in $D$-dimensional space. Here, $D$ means the number of variables of an optimization problem. Based on the concept of opposite point, opposition-based learning used to the metaheuristic method can be described as follows. Assume that $f(\cdot)$ is objective function of a minimum optimization problem, which is used for measuring the quality of a candidate solution. When opposition-based learning is performed, the point $P$ and its opposite point $\breve{P}$ are evaluated at the same time. If $f(\breve{P}) \leq f(P)$ is met, $\breve{P}$ is a better candidate solution than $P$ and $P$ will be replaced with $\breve{P}$. In addition, as a mutation operator, opposition-based learning is controlled by jumping rate $J_{\mathrm{r}}$.$J_{\mathrm{r}}$ is often set to 0.3 [15,23].

### 2.2. Backtracking search algorithm

As shown in Fig. 1, backtracking search algorithm has a very simple structure, which has five components, i.e. initialization, selection-I, mutation, crossover and selection-II.

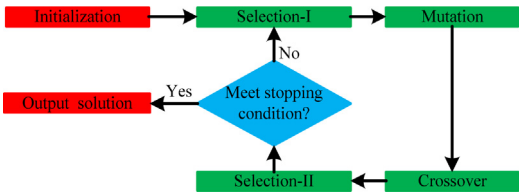


**Fig. 1.** The framework of backtracking search algorithm.

- Initialization. This phase is to initialize population $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ and historical population $\boldsymbol{X}_{\text{old}} = \{\boldsymbol{x}_{\text{old},1}, \boldsymbol{x}_{\text{old},2}, \ldots, \boldsymbol{x}_{\text{old},N}\}$, where $N$ is population size.$\boldsymbol{X}$ can be initialized by

$$x_{i,j} = l_j + (u_j - l_j) \times \xi, i = 1, 2, \ldots, N, j = 1, 2, \ldots, D \quad (3)$$

  where $x_{i,j}$ is the value of the $j$th variable of $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$, $l_j$ is the lower limit of the $j$th variable of $\boldsymbol{x}_i$, $u_j$ is the upper limit of the $j$th variable of $\boldsymbol{x}_i$ and $\xi$ is a random number between 0 and 1. In addition, $\boldsymbol{X}_{\text{old}}$ is initialized by the same method with $\boldsymbol{X}$.
- Selection-I.$\boldsymbol{X}_{\text{old}}$ plays a very important role, which is used to guide the search direction of $\boldsymbol{X}$. This phase is to update $\boldsymbol{X}_{\text{old}}$, which can be expressed as

$$\boldsymbol{X}_{\text{old}} = \begin{cases} \boldsymbol{X}, \rho_1 > \rho_2 \\ \boldsymbol{X}_{\text{old}}, \text{otherwise} \end{cases} \quad (4)$$

  where $\rho_1$ and $\rho_2$ are two random numbers between 0 and 1 with uniform distribution. In order to enhance the randomness, $\boldsymbol{X}_{\text{old}}$ is further processed by

$$\boldsymbol{X}_{\text{old}} = \text{permuting}\,(\boldsymbol{X}_{\text{old}}) \quad (5)$$

  where permuting is a random shuffling function that is used to randomly sort for the individuals of $\boldsymbol{X}_{\text{old}}$ [10].
- Mutation. Mutation operator is to produce a basic solution $\boldsymbol{\psi}_i = (\psi_{i,1}, \psi_{i,2}, \ldots, \psi_{i,D})$ for updating $\boldsymbol{x}_i$, which can be represented as

$$\boldsymbol{\psi}_i = \boldsymbol{x}_i + F \times (\boldsymbol{x}_{\text{old},i} - \boldsymbol{x}_i) \quad (6)$$

  where $F$ is called scale factor that is set to $3m$ in the basic backtracking search algorithm [10]. In addition, $m$ is random number with standard normal distribution.
- Crossover. Crossover operator is to generate the final solution $\boldsymbol{\chi}_i = (\chi_{i,1}, \chi_{i,2}, \ldots, \chi_{i,D})$, which is performed by a binary integer-valued matrix $\boldsymbol{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N\}$ of size $N \times D$ and can be denoted by

$$\chi_{i,j} = \begin{cases} x_{i,j}, \text{if } e_{i,j} = 1 \\ \psi_{i,j}, \text{if } e_{i,j} = 0 \end{cases}, i = 1, 2, \ldots, N, j = 1, 2, \ldots, D \quad (7)$$

  where $\boldsymbol{e}_i = (e_{i,1}, e_{i,2}, \ldots, e_{i,D})$ is the integer-valued vector of the $i$th individual. In addition, $\boldsymbol{E}$ is controlled by a parameter called mix rate $M_{\text{rate}}$ that is set to 1 according to the authors of backtracking search algorithm [10]. The detailed way to generate $\boldsymbol{E}$ can be found the basic backtracking search algorithm [10].
- Selection-II. This phase is to select the fitter solution between $\boldsymbol{\chi}_i$ and $\boldsymbol{x}_i$ into the next generation population, which can be computed by

$$\boldsymbol{x}_i = \begin{cases} \boldsymbol{\chi}_i, \text{if } f(\boldsymbol{\chi}_i) \le f(\boldsymbol{x}_i) \\ \boldsymbol{x}_i, \text{otherwise} \end{cases} \quad (8)$$

## 3. The proposed method

This section is to present the proposed method, which first shows the built specular reflection learning model and then gives the implementation of the proposed BSA_SRL.

### 3.1. The built specular reflection learning model

Fig. 2 presents the schematic diagram of the built specular reflection learning model. As shown in Fig. 2(a), specular reflection is a very common physical phenomenon: the light that reflects off objects with a shiny surface. In Fig. 2(a), $Ao$ and $oB$ are incident light and reflect light, respectively. The angle $\alpha$ between $Ao$ and normal is called angle of incidence. The angle $\beta$ between $oB$ and normal is called angle of reflection. According to the law of reflection, $\alpha$ is equal to $\beta$.

Fig. 2(b) shows the built specular reflection learning model. In Fig. 2(b), $o = (x_0, 0)$ is the midpoint of $[V_L, V_U]$ and $x = (a, 0)$ is a random variable within $[V_L, V_U]$. Thus, $\bar{x} = (2x_0 - a, 0)$ can be denoted as the opposite number of $x$ obtained by Eq. (1). In the proposed specular reflection learning, assume the obtained opposite number of $x$ is $\bar{x} = (b, 0)$. From Fig. 2(b), the following formula can be met,

$$\tan(\alpha) = \frac{x_0 - a}{A_0} \quad (9)$$

$$\tan(\beta) = \frac{b - x_0}{B_0} \quad (10)$$

As mentioned previously, $\alpha$ is equal to $\beta$. Thus, the relationship between Eqs. (9) and (10) can be represented by

$$\frac{x_0 - a}{A_0} = \frac{b - x_0}{B_0} \quad (11)$$

Based on Eq. (11), $b$ can be computed by

$$b = \frac{B_0\,(x_0 - a)}{A_0} + x_0 \quad (12)$$

Assume $B_0 = \lambda A_0$, where $\lambda(\lambda > 0)$ is a real number. Thus, Eq. (12) can be rewritten as

$$\begin{aligned} b &= \lambda\,(x_0 - a) + x_0 \\ &= (\lambda + 1)\,x_0 - \lambda a \\ &= (0.5\lambda + 0.5) \times 2x_0 - \lambda a \\ &= (0.5\lambda + 0.5) \times (V_L + V_U) - \lambda a \end{aligned} \quad (13)$$

From Eq. (13), when $\lambda$ is set to different value, the value of $b$ can be computed by

$$b = \begin{cases} b_1, 0 < \lambda < 1 \\ 2x_0 - a, \lambda = 1 \\ b_2, \lambda > 1 \end{cases} \quad (14)$$

where $b_1$ and $b_2$ meet $b_1 \in (x_0, 2x_0 - a)$ and $b_2 \in (2x_0 - a, V_U]$, respectively. As can be seen from Eq. (14), when $\lambda$ is set to 1, which means specular reflection learning will be opposition-based learning. That is, opposition-based learning is a special case of specular reflection learning. In order to follow the basic idea of opposition-based learning, the possible position of the opposite number $b$ obtained by specular reflection learning should be a small neighborhood of $2x_0 - a$. In the designed specular reflection learning, $\lambda$ can be computed by

$$\lambda = \begin{cases} 1 + \varphi R_0, \text{if } \kappa_1 > \kappa_2 \\ 1 - \varphi R_0, \text{otherwise} \end{cases} \quad (15)$$

where $\kappa_1$ and $\kappa_2$ are two random numbers between 0 and 1 with uniform distribution, $R_0$ can be called neighborhood radius, and

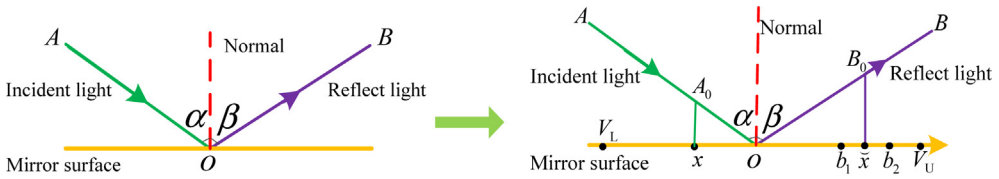


**Fig. 2.** The schematic diagram of specular reflection learning.

$\varphi$ can be called elastic factor. In addition, $\varphi$ is set to [0, 1], which is to obtain all possible values within the neighborhood radius $R_0$. From Eq. (15), if $\kappa_1 > \kappa_2$ is met, the position of the opposite number $b$ will appear at the right neighborhood of $2x_0 - a$; otherwise, $\bar{x}$ will appear at the left neighborhood of $2x_0 - a$. Here, note that, if $R_0$ is equal to 1, the maximum value and the minimum value of $b$ are $3x_0 - 2a$ and $x_0$, respectively. Note that $3x_0 - 2a$ and $x_0$ are symmetrical about $2x_0 - a$ that is the opposite number of $x$ obtained by opposition-based learning. From viewpoint of symmetry, when $R_0$ is more than 1, the obtained maximum value of $b$ is larger than $2x_0 - a$; the obtained minimum value of $b$ is smaller than $x_0$. In other words, when $R_0$ is more than 1, the number $x$ and its opposite number $b$ may be appear the same side of the midpoint $x_0$. Obviously, this is contrary to the basic idea of opposition-based learning. Thus, in order to follow the basic idea of opposition-based learning, $R_0$ is set to [0, 1].

### 3.2. The proposed BSA_SRL

Based on the built specular reflection learning model, opposite number in Eq. (1) can be rewritten as

$$\bar{x} = (0.5\lambda + 0.5) \times (V_L + V_U) - \lambda x \tag{16}$$

In addition, opposite point in Eq. (2) also can be rewritten as

$$\bar{x}_i = (0.5\lambda + 0.5) \times \left(V_{L,i} + V_{U,i}\right) - \lambda x_i, i = 1, 2, \ldots, D \tag{17}$$

Based on Eqs. (16)–(17), when specular reflection learning is used for improving the performance of metaheuristics, the opposite solution $\bar{P} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_D)$ of the solution $P = (x_1, x_2, \ldots, x_D)$ obtained by specular reflection can be denoted as

$$\bar{x}_i = (0.5\lambda + 0.5) \times (U_{max,i} + L_{min,i}) - \lambda x_i, i = 1, 2, \ldots, D \tag{18}$$

where $\boldsymbol{U}_{\max} = \left(U_{max,1}, U_{max,2}, \ldots, U_{max,D}\right)$ and $\boldsymbol{L}_{\min} = \left(L_{min,1}, L_{min,2}, \ldots, L_{min,D}\right)$ are the upper limits and lower limits of the current population, respectively.

Like opposition-based learning, specular reflection learning is also as a mutation operator used for improving metaheuristic methods by jumping rate $J_r$. In addition, some elements produced by Eq. (18) may be out of boundary, which is handled by

$$\bar{x}_i = \begin{cases} l_i, \text{ if } \bar{x}_i < l_i \\ u_i, \text{ if } \bar{x}_i > u_i \end{cases} \tag{19}$$

Fig. 3 presents the pseudocode of the proposed BSA_SRL. From Fig. 3, BSA_SRL has a very simple structure and is easy to work.

## 4. BSA_SRL for unconstrained numerical optimization

The proposed specular reflection learning is closely related with opposition-based learning. In addition, opposition-based learning can be as a special case of specular reflection learning. Thus, it is very essential for comparing the performance of opposition-based learning and specular reflection learning used for improving metaheuristic methods. In this section, the designed experiments focus on investigating the performance differences among BSA, BSA_SRL and backtracking search algorithm with opposition-based learning (BSA_OBL) for various types of unconstrained optimization problems. The unconstrained numerical optimization problems are extracted from three well-known test suites, i.e. CEC 2013 [24], CEC2014 [25,26] and CEC 2017 [27,28]. CEC 2013 test suite includes five unimodal functions, 15 simple multimodal functions, and eight composition functions. CEC 2014 test suite consists of three unimodal functions, 13 simple multimodal functions, six hybrid functions and eight composition functions. There are three unimodal functions, seven simple multimodal functions, 10 hybrid functions and 10 composition functions in CEC 2017 test suite. Compared with unimodal functions, multimodal functions are more complex. Thus, the used test problems are very suitable for checking the ability of BSA, BSA_SRL and BSA_OBL for solving complex problems. The definitions of CEC 2013, CEC 2014 and CEC 2017 test suites have been listed in Appendix.

In order to make a fair comparison, for BSA, BSA_OBL and BSA_SRL, population size and the number of independent runs were set to 50 and 50 for each test problems, respectively. In addition, the maximum number of function evaluations was seen as the terminal condition, which was set to 150,000. Besides, jumping rate $J_r$ for BSA_OBL and BSA_SRL was set to 0.3 [15]. In addition, when $\lambda$ is set to 1 (see line 19 in Fig. 3), Fig. 3 can be seen as the pseudocode of BSA_OBL algorithm. Thus, the pseudocode of BSA_OBL is not given again. Besides, during the search process, the obtained solution may be out of boundary, which is processed by

$$x_{i,j} = \begin{cases} l_j, \text{ if } x_{i,j} < l_j \\ u_j, \text{ if } x_{i,j} > u_j \end{cases} \tag{20}$$

The experimental results obtained by BSA, BSA_OBL and BSA_SRL for CEC 2013, CEC 2014 and CEC 2017 test suites are shown in Table 1, Table 3 and Table 5, respectively. In addition, in the three tables, the optimal solutions were marked in bold type; "Mean" and "Std" stand for mean value and standard variance of the best solutions from 50 independent runs, respectively. Besides, in order to test the significant difference of the solutions achieved by the applied algorithms, Wilcoxon signed ranks test [29] with a significant level $\alpha = 0.05$ is executed for the mean solutions of 50 independent runs obtained by BSA, BSA_OBL and BSA_SRL. The results of Wilcoxon signed ranks test are presented in Table 2, Tables 4 and 6. In Table 2, Tables 4 and 6, "R+" means the sum of ranks for the test function in which BSA_SRL outperforms the compared algorithm, and "R-" is the sum of ranks for the opposite [29]; "+", "=" and "−" indicate that BSA_SRL is superior, equal and inferior to the compared algorithm, respectively.

### 4.1. Experimental results for CEC 2013

Fig. 4 shows the comparison results among BSA, BSA_OBL and BSA_SRL for CEC 2013 test suite. According to Fig. 4, BSA_SRL can offer better solutions than BSA and BSA_OBL on more than

**BSA_SRL Algorithm**

**Input:** Initialize mix_rate $M_{\text{rate}}$, jumping rate $J_{\text{r}}$, population size $N$, the number of variables $D$, the objective function $f(\cdot)$, the current number of function evaluations $T_{\text{C}}$ and the maximum number of function evaluations $T_{\max}$

01: **Begin**
02: Initialize population $\boldsymbol{X}$ and population $\boldsymbol{X}_{\text{old}}$ by Eq. (3) // *Initialization population*
03: Calculate the fitness value of each solution and select the optimal solution $\boldsymbol{x}_{\text{Best}}$ // *Population evaluation*
04: Update the current number of function evaluations $T_{\text{C}}$ by $T_{\text{C}}=T_{\text{C}}+N$
05: **Repeat** // *Main loop*
06: Perform Selection-I operator to update historical population $\boldsymbol{X}_{\text{old}}$ by Eqs. (4-5) // *Selection-I operator*
07: Calculate scale factor $F$ and update the matrix $\boldsymbol{E}$ // Update parameters of backtracking search algorithm
08: **For** each individual $i \in N$ **do**
09: Perform mutation operator to obtain $\boldsymbol{\psi}_i$ by Eq. (6) // *Mutation operator*
10: Perform crossover operator to obtain $\boldsymbol{\chi}_i$ by Eq. (7) // *Crossover operator*
11: Perform Selection-II operator to obtain $\boldsymbol{x}_i$ by Eq. (8) // *Selection-II operator*
12: Update the current number of function evaluations $T_{\text{C}}$ by $T_{\text{C}}=T_{\text{C}}+1$
13: **End for**
14: Generate a random number $\eta$ between 0 and 1
15: **If** $\eta < J_{\text{r}}$ // Condition for performing specular reflection learning
16: Calculate the upper limits $\boldsymbol{U}_{\max}$ and lower limits $\boldsymbol{L}_{\min}$ of $\boldsymbol{X}$ // *Start (specular reflection learning)*
17: **For** each individual $i \in N$ **do**
18: Generate neighborhood radius $R_0$ and elastic factor $\varphi$
19: Generate $\lambda$ by Eq. (15)
20: Obtain the opposite solution of $\boldsymbol{x}_i$ by Eq. (18)
21: Check the boundary for the opposite solution of $\boldsymbol{x}_i$ by Eq. (19)
22: Calculate the fitness value of the opposite solution of $\boldsymbol{x}_i$
23: Update the current number of function evaluations $T_{\text{C}}$ by $T_{\text{C}}=T_{\text{C}}+1$
24: **End for**
25: Select the first $N$ best individuals from population $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N\}$ and $\boldsymbol{P}=\{\boldsymbol{p}_1,\boldsymbol{p}_2,\ldots,\boldsymbol{p}_N\}$ as the next generation population $\boldsymbol{X}$ // *End (specular reflection learning)*
26: **End if**
27: Update the optimal solution $\boldsymbol{x}_{\text{Best}}$
28: **Until** $(T_{\text{C}} > T_{\max})$ // *Terminal condition*
29: **End**

**Output:** The optimal solution $\boldsymbol{x}_{\text{Best}}$

**Fig. 3.** The pseudocode of BSA_SRL algorithm.

sixty percent of test functions with 10-dimensional and 30-dimensional.

From Table 1, for test functions with 10-dimensional, BSA_SRL can outperform BSA and BSA_OBL on 17 (i.e. F3, F6, F7, F9, F12, F13, F15, F16, F17, F18, F20, F22, F23, F24, F25, F26 and F27) and 18 (i.e. F2, F3, F4, F6, F7, F11, F13, F14, F15, F16, F17, F18, F20, F22, F23, F25, F26 and 27) test functions, respectively. In addition, BSA_SRL can get the same solutions with BSA and BSA_OBL on 5 (i.e. F1, F5, F8, F10 and F11) and 5 (i.e. F1, F5, F8, F9 and F10) functions, respectively. However, BSA and BSA_OBL only can beat BSA_SRL on 6 (i.e. F2, F4, F11, F14, F19 and F28) and 5 (i.e. F12, F19, F21, F24 and F28) functions, respectively.

As can be seen from Table 1, for test functions with 30-dimensional, BSA, BSA_OBL and BSA_SRL can offer the same solutions on F1, F5 and F8. In addition, BSA and BSA_OBL are superior to BSA_SRL on 6 (i.e. F12, F13, F14, F18, F21 and F22) and 7 (i.e. F3, F12, F13, F14, F21, F22 and F24) functions, respectively. But BSA and BSA_OBL cannot compete with BSA_SRL on 19 (i.e. F2, F3, F4, F6, F7, F9, F10, F11, F15, F16, F17, F19, F20, F23, F24, F25, F26, F27 and F28) and 18 (i.e. F2, F4, F6, F7, F9, F10, F11, F15, F16, F17, F18, F19, F20, F23, F25, F26, F27 and 28) functions, respectively.

By observing the statistical results of Wilcoxon signed ranks test displayed in Table 2, BSA_SRL is superior to BSA on 12 functions with 10-dimensional (i.e. F3, F6, F7, F8, F9, F13, F15, F16, F17, F18, F20 and F27) and 16 functions with 30-dimensional (i.e. F1, F2, F3, F5, F6, F8, F9, F10, F11, F15, F16, F17, F19, F20, F23 and F28). BSA_SRL can achieve better solutions than BSA_OBL on eight functions with 10-dimensional (i.e. F3, F6, F7, F8, F15, F16, F18 and F27) and 11 functions with 30-dimensional (i.e. F2, F5, F7, F8, F10, F15, F19, F20, F23, F25 and F28). However, BSA only can beat BSA_SRL on four functions with 10-dimensional (i.e. F2, F4, F10 and F14) and three functions with 30-dimensional (i.e. F13, F14 and F22). In addition, BSA_OBL only outperforms BSA_SRL on F28 with 10-dimensional and has no advantages over BSA_SRL on any functions with 30-dimensional.

### 4.2. Experimental results for CEC 2014

The comparison results among BSA, BSA_OBL and BSA_SRL for CEC 2014 test suite are presented in Fig. 5. From Fig. 5, BSA_SRL outperforms BSA and BSA_OBL on more than forty percent of test functions with 10-dimensional and 30-dimensional.

As presented in Table 3, for test functions with 10-dimensional, BSA_SRL can give better solutions than BSA and BSA_OBL on 13 (i.e. F29, F32, F33, F34, F35, F36, F38, F45, F52, F53, F55, F56

**Table 1**
Experimental results obtained by BSA, BSA_OBL and BSA_SRL for CEC 2013 test suite.

| No. | Indicator | 10-dimensional | | | 30-dimensional | | |
|---|---|---|---|---|---|---|---|
| | | BSA | BSA_OBL | BSA_SRL | BSA | BSA_OBL | BSA_SRL |
| F1 | Mean | **−1.400E+03** | **−1.400E+03** | **−1.400E+03** | **−1.400E+03** | **−1.400E+03** | **−1.400E+03** |
| | Std | **0.000E+00** | **0.000E+00** | **0.000E+00** | 1.455E−09 | **2.698E−13** | 3.264E−13 |
| F2 | Mean | **3.858E+04** | 5.538E+04 | 5.257E+04 | 4.239E+06 | 3.579E+06 | **1.377E+06** |
| | Std | 6.110E+04 | **5.501E+04** | 5.736E+04 | 2.352E+06 | 2.126E+06 | **9.054E+05** |
| F3 | Mean | 3.329E+08 | 3.851E+08 | **1.502E+04** | 1.872E+09 | **9.371E+07** | 1.153E+09 |
| | Std | 2.354E+09 | 2.500E+09 | **9.117E+04** | 1.107E+10 | **1.126E+08** | 6.246E+09 |
| F4 | Mean | **7.969E+02** | 1.758E+03 | 1.488E+03 | 2.255E+04 | 2.362E+04 | **2.157E+04** |
| | Std | **6.228E+02** | 1.340E+03 | 1.503E+03 | 6.201E+03 | 6.508E+03 | **5.617E+03** |
| F5 | Mean | **−1.000E+03** | **−1.000E+03** | **−1.000E+03** | **−1.000E+03** | **−1.000E+03** | **−1.000E+03** |
| | Std | 3.248E−14 | **2.297E−14** | 4.297E−14 | 6.119E−06 | 3.414E−03 | **3.867E−13** |
| F6 | Mean | −8.931E+02 | −8.942E+02 | **−8.949E+02** | −8.327E+02 | −8.595E+02 | **−8.673E+02** |
| | Std | **4.498E+00** | 4.759E+00 | 4.949E+00 | **1.839E+01** | 2.715E+01 | 2.821E+01 |
| F7 | Mean | −7.904E+02 | −7.923E+02 | **−7.977E+02** | −7.117E+02 | −7.133E+02 | **−7.304E+02** |
| | Std | 1.420E+01 | 1.763E+01 | **5.570E+00** | 4.278E+01 | 3.871E+01 | **2.311E+01** |
| F8 | Mean | **−6.796E+02** | **−6.796E+02** | **−6.796E+02** | **−6.790E+02** | **−6.790E+02** | **−6.790E+02** |
| | Std | **6.390E−02** | 7.654E−02 | 8.324E−02 | 7.150E−02 | 8.048E−02 | **6.545E−02** |
| F9 | Mean | −5.953E+02 | **−5.974E+02** | **−5.974E+02** | −5.693E+02 | −5.711E+02 | **−5.720E+02** |
| | Std | 1.687E+00 | 1.462E+00 | **1.143E+00** | 4.445E+00 | **3.961E+00** | 4.585E+00 |
| F10 | Mean | **−4.999E+02** | **−4.999E+02** | **−4.999E+02** | −4.980E+02 | −4.997E+02 | **−4.999E+02** |
| | Std | **4.496E−02** | 7.266E−02 | 4.587E−02 | 8.366E−01 | 1.282E−01 | **5.899E−02** |
| F11 | Mean | **−3.999E+02** | −3.997E+02 | −3.998E+02 | −3.756E+02 | −3.782E+02 | **−3.794E+02** |
| | Std | **4.025E−01** | 5.966E−01 | 4.197E−01 | **8.312E+00** | 1.032E+01 | 9.949E+00 |
| F12 | Mean | −2.929E+02 | **−2.932E+02** | −2.931E+02 | **−1.888E+02** | −1.832E+02 | −1.782E+02 |
| | Std | 3.034E+00 | 3.167E+00 | **2.645E+00** | **2.705E+01** | 3.456E+01 | 3.373E+01 |
| F13 | Mean | −1.845E+02 | −1.879E+02 | **−1.885E+02** | **−2.180E+01** | −1.196E+01 | 1.049E+00 |
| | Std | 6.787E+00 | 6.294E+00 | **6.172E+00** | 2.097E+01 | **2.076E+01** | 2.587E+01 |
| F14 | Mean | **−8.536E+01** | −5.091E+01 | −6.757E+01 | **1.886E+03** | 2.310E+03 | 2.362E+03 |
| | Std | **9.605E+00** | 2.984E+01 | 2.617E+01 | **3.033E+02** | 4.333E+02 | 3.518E+02 |
| F15 | Mean | 1.085E+03 | 1.065E+03 | **6.027E+02** | 6.104E+03 | 5.230E+03 | **4.617E+03** |
| | Std | 3.466E+02 | 2.580E+02 | **1.819E+02** | **7.785E+02** | 9.208E+02 | 8.496E+02 |
| F16 | Mean | 2.012E+02 | 2.011E+02 | **2.007E+02** | 2.024E+02 | 2.018E+02 | **2.016E+02** |
| | Std | 5.134E−01 | 5.904E−01 | **3.808E−01** | 7.677E−01 | 7.519E−01 | **7.422E−01** |
| F17 | Mean | 3.117E+02 | 3.112E+02 | **3.109E+02** | 3.893E+02 | 3.876E+02 | **3.841E+02** |
| | Std | 7.138E−01 | 1.064E+00 | **3.631E−01** | 1.137E+01 | **1.037E+01** | 1.128E+01 |
| F18 | Mean | 4.244E+02 | 4.228E+02 | **4.201E+02** | **5.906E+02** | 5.913E+02 | 5.907E+02 |
| | Std | **3.230E+00** | 3.953E+00 | 3.332E+00 | **2.851E+01** | 3.759E+01 | 3.213E+01 |
| F19 | Mean | **5.005E+02** | **5.005E+02** | 5.006E+02 | 5.055E+02 | 5.052E+02 | **5.045E+02** |
| | Std | 1.185E−01 | 1.475E−01 | **1.084E−01** | 1.781E+00 | **1.462E+00** | 1.519E+00 |
| F20 | Mean | 6.034E+02 | 6.033E+02 | **6.032E+02** | 6.130E+02 | 6.130E+02 | **6.125E+02** |
| | Std | 2.789E−01 | **2.401E−01** | 3.756E−01 | **4.475E−01** | 4.481E−01 | 6.126E−01 |
| F21 | Mean | 1.000E+03 | **9.880E+02** | 1.000E+03 | **1.014E+03** | 1.023E+03 | 1.040E+03 |
| | Std | 3.248E−14 | 5.938E+01 | **1.624E−14** | **4.338E+01** | 6.967E+01 | 8.698E+01 |
| F22 | Mean | 9.935E+02 | 9.827E+02 | **9.750E+02** | **2.763E+03** | 3.536E+03 | 3.648E+03 |
| | Std | 5.937E+01 | 5.575E+01 | **3.730E+01** | **3.571E+02** | 7.052E+02 | 6.676E+02 |
| F23 | Mean | 2.003E+03 | 2.021E+03 | **1.957E+03** | 7.340E+03 | 6.832E+03 | **6.264E+03** |
| | Std | 2.954E+02 | 2.988E+02 | **2.187E+02** | 1.081E+03 | 9.012E+02 | **8.639E+02** |
| F24 | Mean | 1.216E+03 | **1.204E+03** | 1.205E+03 | 1.285E+03 | **1.283E+03** | 1.284E+03 |
| | Std | **7.953E+00** | 3.135E+01 | 3.000E+01 | 1.262E+01 | 1.517E+01 | **1.215E+01** |
| F25 | Mean | 1.319E+03 | 1.319E+03 | **1.317E+03** | 1.401E+03 | 1.402E+03 | **1.397E+03** |
| | Std | **2.556E+00** | 4.393E+00 | 1.386E+01 | **1.311E+01** | 1.418E+01 | 1.374E+01 |
| F26 | Mean | 1.393E+03 | 1.388E+03 | **1.377E+03** | 1.457E+03 | 1.463E+03 | **1.455E+03** |
| | Std | **1.910E+01** | 2.898E+01 | 3.911E+01 | **8.381E+01** | 8.658E+01 | 8.445E+01 |
| F27 | Mean | 1.844E+03 | 1.815E+03 | **1.680E+03** | 2.440E+03 | 2.405E+03 | **2.391E+03** |
| | Std | **3.325E+01** | 6.803E+01 | 9.442E+01 | 1.322E+02 | 1.276E+02 | **1.013E+02** |
| F28 | Mean | 1.676E+03 | **1.644E+03** | 1.680E+03 | 1.696E+03 | 1.718E+03 | **1.692E+03** |
| | Std | **6.001E+01** | 9.071E+01 | 6.061E+01 | **2.828E+01** | 1.566E+02 | 3.959E+01 |

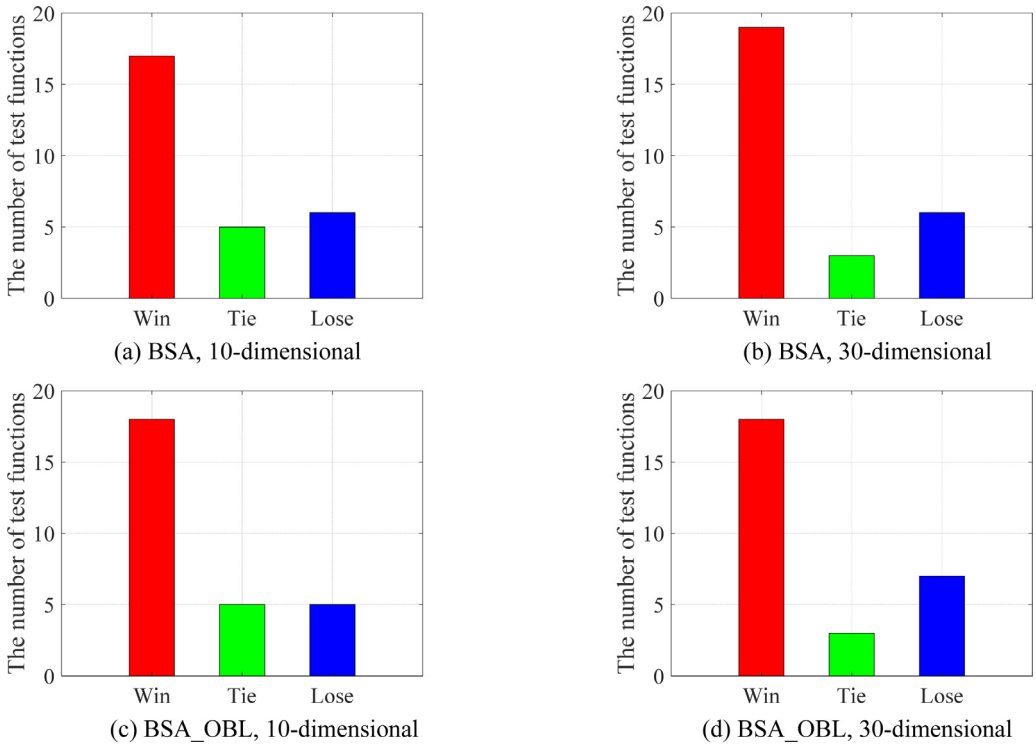


**Fig. 4.** The comparison results among BSA, BSA_OBL and BSA_SRL for CEC 2013 test suite. "Win", "Tie" and "Lose" means the number of test functions that BSA_SRL is superior, equal and inferior to the compared algorithms, respectively.

**Table 2**
Results produced by Wilcoxon signed ranks test for CEC 2013 test suite.

| No. | 10-dimensional | | | | | | | | 30-dimensional | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BSA_SRL vs. BSA | | | | BSA_SRL vs. BSA_OBL | | | | BSA_SRL vs. BSA | | | | BSA_SRL vs. BSA_OBL | | | |
| | R+ | R- | *P*-value | S | R+ | R- | *P*-value | S | R+ | R- | *P*-value | S | R+ | R- | *P*-value | S |
| F1 | 637.5 | 637.5 | 1.00E+0 | = | 637.5 | 637.5 | 1.00E+0 | = | 1275 | 0 | 7.56E−10 | + | 637 | 638 | 1.00E+0 | = |
| F2 | 418 | 857 | 3.41E−2 | − | 700 | 575 | 5.46E−1 | = | 1248 | 27 | 3.78E−9 | + | 1208 | 67 | 3.65E−8 | + |
| F3 | 952 | 323 | 2.40E−3 | + | 1026 | 249 | 1.77E−4 | + | 935 | 340 | 4.08E−3 | + | 700 | 575 | 5.46E−1 | = |
| F4 | 426 | 849 | 4.12E−2 | − | 763 | 512 | 2.26E−1 | = | 691 | 584 | 6.06E−1 | = | 798 | 477 | 1.21E−1 | = |
| F5 | 637.5 | 637.5 | 1.00E+0 | = | 637.5 | 637.5 | 1.00E+0 | = | 1275 | 0 | 7.56E−10 | + | 826.5 | 448.5 | 5.59E−6 | + |
| F6 | 1039 | 236 | 1.06E−4 | + | 755 | 520 | 3.58E−2 | + | 1184 | 91 | 1.32E−7 | + | 824 | 451 | 7.18E−2 | = |
| F7 | 983 | 292 | 8.52E−4 | + | 897 | 378 | 1.22E−2 | + | 827 | 448 | 6.74E−2 | = | 895 | 380 | 1.29E−2 | + |
| F8 | 936.5 | 338.5 | 4.84E−4 | + | 853 | 422 | 3.21E−2 | + | 1046 | 229 | 8.03E−5 | + | 995 | 280 | 5.58E−4 | + |
| F9 | 1165 | 110 | 3.54E−7 | + | 621 | 654 | 8.73E−1 | = | 918 | 357 | 6.77E−3 | + | 757 | 518 | 2.49E−1 | = |
| F10 | 417 | 858 | 3.33E−2 | − | 543 | 732 | 3.62E−1 | = | 1275 | 0 | 7.56E−10 | + | 1226 | 49 | 1.34E−8 | + |
| F11 | 635 | 640 | 9.65E−1 | = | 651.5 | 623.5 | 2.75E−1 | = | 912 | 363 | 8.05E−3 | + | 716 | 559 | 4.49E−1 | = |
| F12 | 706 | 569 | 5.08E−1 | = | 589 | 686 | 6.40E−1 | = | 510 | 765 | 2.18E−1 | = | 587 | 688 | 6.26E−1 | = |
| F13 | 963 | 312 | 1.68E−3 | + | 687 | 588 | 6.33E−1 | = | 382 | 893 | 1.36E−2 | − | 497 | 778 | 1.75E−1 | = |
| F14 | 256 | 1019 | 2.31E−4 | − | 836 | 439 | 5.53E−2 | = | 95 | 1180 | 1.63E−7 | − | 538 | 737 | 3.37E−1 | = |
| F15 | 1202 | 73 | 5.06E−8 | + | 1274 | 1 | 8.03E−10 | + | 1217 | 58 | 2.22E−8 | + | 1020 | 255 | 2.22E−4 | + |
| F16 | 1152 | 123 | 6.81E−7 | + | 1053 | 222 | 6.05E−5 | + | 1101 | 174 | 7.67E−6 | + | 755 | 520 | 2.57E−1 | = |
| F17 | 1195 | 80 | 7.38E−8 | + | 752 | 523 | 2.69E−1 | = | 866 | 409 | 2.74E−2 | + | 817 | 458 | 8.31E−2 | = |
| F18 | 1172 | 103 | 2.47E−7 | + | 949 | 326 | 2.64E−3 | + | 607 | 668 | 7.68E−1 | = | 632 | 643 | 9.58E−1 | = |
| F19 | 547 | 728 | 3.82E−1 | = | 555 | 720 | 4.26E−1 | = | 961 | 314 | 1.79E−3 | + | 863 | 412 | 2.95E−2 | + |
| F20 | 981 | 294 | 9.13E−4 | + | 750 | 525 | 2.77E−1 | = | 1041 | 234 | 9.82E−5 | + | 1009 | 266 | 3.36E−4 | + |
| F21 | 637.5 | 637.5 | 1.00E+0 | = | 636 | 639 | 5.00E−1 | = | 618 | 657 | 8.51E−1 | = | 692.5 | 582.5 | 5.60E−1 | = |
| F22 | 730 | 545 | 3.72E−1 | = | 680 | 595 | 6.82E−1 | = | 89 | 1186 | 1.19E−7 | − | 537 | 738 | 3.32E−1 | = |
| F23 | 715 | 560 | 4.54E−1 | = | 770 | 505 | 2.01E−1 | = | 1058 | 217 | 4.92E−5 | + | 913 | 362 | 7.83E−3 | + |
| F24 | 831 | 444 | 6.18E−2 | = | 655 | 620 | 8.66E−1 | = | 648 | 627 | 9.19E−1 | = | 598 | 677 | 7.03E−1 | = |
| F25 | 800 | 475 | 1.17E−1 | = | 762 | 513 | 2.29E−1 | = | 749 | 526 | 2.82E−1 | = | 844 | 431 | 4.62E−2 | + |
| F26 | 739 | 536 | 3.27E−1 | = | 757 | 518 | 2.49E−1 | = | 715 | 560 | 4.54E−1 | = | 774 | 501 | 1.88E−1 | = |
| F27 | 1259 | 16 | 1.98E−9 | + | 1242 | 33 | 5.37E−9 | + | 808 | 467 | 9.98E−2 | = | 716 | 559 | 4.49E−1 | = |
| F28 | 638.5 | 636.5 | 9.44E−1 | = | 606 | 669 | 2.25E−2 | − | 1227 | 48 | 1.27E−8 | + | 1228 | 47 | 1.19E−8 | + |
| +/= /− | 12/12/4 | | | | 8/19/1 | | | | 16/9/3 | | | | 11/17/0 | | | |

and F57) and 13 (i.e. F29, F32, F34, F36, F37, F38, F39, F52, F53, F55, F56, F57 and F58) test functions, respectively. In addition, BSA_SRL, BSA and BSA_OBL can share the same solutions on 12 (i.e. F30, F31, F37, F40, F41, F42, F43, F44, F47, F48, F51 and F54) and 13 (i.e. F30, F31, F33, F35, F40, F41, F42, F43, F44, F47, F50, F51 and F54) functions, respectively. Note that, BSA and BSA_OBL only are superior to BSA_SRL on 5 (i.e. F39, F46, F49, F50 and F58) and 4 (i.e. F45, F46, F48 and F49) functions, respectively.

Based on Table 3, for test functions with 30-dimensional, BSA and BSA_OBL only can beat BSA_SRL on four (i.e. F38, F45, F48

**Table 3**
Experimental results obtained by BSA, BSA_OBL and BSA_SRL for CEC 2014 test suite.

| No. | Indicator | 10-dimensional | | | 30-dimensional | | |
|---|---|---|---|---|---|---|---|
| | | BSA | BSA_OBL | BSA_SRL | BSA | BSA_OBL | BSA_SRL |
| F29 | Mean | 1.654E+02 | 1.299E+02 | **1.022E+02** | 1.690E+06 | 1.894E+06 | **4.478E+05** |
| | Std | 1.317E+02 | 1.219E+02 | **4.704E+00** | 1.027E+06 | 1.154E+06 | **3.414E+05** |
| F30 | Mean | **2.000E+02** | **2.000E+02** | **2.000E+02** | 8.475E+03 | 9.813E+03 | **2.000E+02** |
| | Std | 2.503E−14 | 1.573E−14 | **1.347E−14** | 5.784E+03 | 9.085E+03 | **2.287E−06** |
| F31 | Mean | **3.000E+02** | **3.000E+02** | **3.000E+02** | 6.914E+02 | 5.659E+02 | **3.172E+02** |
| | Std | **8.120E−15** | 3.046E−13 | 5.006E−14 | 7.083E+02 | 7.838E+02 | **2.941E+01** |
| F32 | Mean | 4.229E+02 | 4.248E+02 | **4.219E+02** | 4.984E+02 | 4.809E+02 | **4.512E+02** |
| | Std | 1.607E+01 | **1.546E+01** | 1.664E+01 | **2.843E+01** | 3.646E+01 | 4.125E+01 |
| F33 | Mean | 5.202E+02 | **5.201E+02** | **5.201E+02** | 5.207E+02 | **5.205E+02** | **5.205E+02** |
| | Std | 6.562E−02 | **4.697E−02** | **4.504E−02** | 9.641E−02 | 1.309E−01 | **1.268E−01** |
| F34 | Mean | 6.019E+02 | 6.019E+02 | **6.004E+02** | 6.201E+02 | 6.153E+02 | **6.109E+02** |
| | Std | 1.868E+00 | 2.003E+00 | **1.060E+00** | 7.600E+00 | 6.909E+00 | **4.567E+00** |
| F35 | Mean | **7.003E+02** | **7.000E+02** | **7.000E+02** | **7.000E+02** | **7.000E+02** | **7.000E+02** |
| | Std | 2.209E+00 | 2.423E−02 | **1.849E−02** | **1.273E−02** | 1.625E−02 | 2.474E−02 |
| F36 | Mean | 8.002E+02 | 8.004E+02 | **8.001E+02** | 8.187E+02 | 8.184E+02 | **8.157E+02** |
| | Std | 4.654E−01 | 7.564E−01 | **2.727E−01** | 8.283E+00 | 6.955E+00 | **5.969E+00** |
| F37 | Mean | **9.042E+02** | 9.050E+02 | **9.042E+02** | 9.727E+02 | **9.666E+02** | 9.683E+02 |
| | Std | **1.577E+00** | 1.982E+00 | 1.784E+00 | 1.752E+01 | **1.573E+01** | 1.850E+01 |
| F38 | Mean | 1.104E+03 | 1.057E+03 | **1.052E+03** | **2.275E+03** | 2.353E+03 | 2.376E+03 |
| | Std | 8.062E+01 | 4.171E+01 | **1.348E+01** | 4.402E+02 | **3.584E+02** | 4.227E+02 |
| F39 | Mean | **1.422E+03** | 1.521E+03 | 1.473E+03 | 6.036E+03 | **4.275E+03** | 4.982E+03 |
| | Std | 1.819E+02 | 2.189E+02 | **1.651E+02** | 1.177E+03 | **6.855E+02** | 8.971E+02 |
| F40 | Mean | **1.200E+03** | **1.200E+03** | **1.200E+03** | **1.201E+03** | **1.201E+03** | **1.201E+03** |
| | Std | **8.727E−02** | 1.025E−01 | 1.214E−01 | 3.728E−01 | 3.737E−01 | **3.049E−01** |
| F41 | Mean | **1.300E+03** | **1.300E+03** | **1.300E+03** | **1.300E+03** | **1.300E+03** | **1.300E+03** |
| | Std | **3.471E−02** | 6.582E−02 | 4.664E−02 | **5.983E−02** | 7.224E−02 | 8.112E−02 |
| F42 | Mean | **1.400E+03** | **1.400E+03** | **1.400E+03** | **1.400E+03** | **1.400E+03** | **1.400E+03** |
| | Std | **5.180E−02** | 6.821E−02 | 6.378E−02 | **3.967E−02** | 5.535E−02 | 5.204E−02 |
| F43 | Mean | **1.501E+03** | **1.501E+03** | **1.501E+03** | 1.510E+03 | **1.509E+03** | **1.509E+03** |
| | Std | **1.791E−01** | 1.845E−01 | 1.881E−01 | 2.965E+00 | **2.657E+00** | 2.824E+00 |
| F44 | Mean | **1.603E+03** | **1.603E+03** | **1.603E+03** | **1.612E+03** | **1.612E+03** | **1.612E+03** |
| | Std | 2.680E−01 | **2.659E−01** | 2.805E−01 | 5.071E−01 | 3.594E−01 | **3.085E−01** |
| F45 | Mean | 1.724E+03 | **1.719E+03** | 1.722E+03 | **3.943E+04** | 6.574E+04 | 5.696E+04 |
| | Std | 3.106E+01 | **1.132E+01** | 1.549E+01 | **3.060E+04** | 6.107E+04 | 5.923E+04 |
| F46 | Mean | **1.801E+03** | **1.801E+03** | 1.802E+03 | 7.434E+03 | **4.406E+03** | 6.015E+03 |
| | Std | 7.530E−01 | **6.857E−01** | 1.619E+00 | 7.501E+03 | **3.778E+03** | 5.478E+03 |
| F47 | Mean | **1.901E+03** | **1.901E+03** | **1.901E+03** | 1.908E+03 | 1.912E+03 | **1.906E+03** |
| | Std | 9.388E−01 | 3.485E−01 | **3.334E−01** | 8.193E+00 | 1.782E+01 | **1.439E+00** |
| F48 | Mean | 2.001E+03 | **2.000E+03** | 2.001E+03 | **2.575E+03** | 2.863E+03 | 2.643E+03 |
| | Std | 6.126E−01 | **4.906E−01** | 5.290E−01 | 7.239E+02 | 1.140E+03 | **6.436E+02** |
| F49 | Mean | **2.101E+03** | **2.101E+03** | 2.102E+03 | 1.239E+04 | 1.332E+04 | **1.046E+04** |
| | Std | 2.393E+00 | 2.433E+00 | **3.392E+00** | 1.066E+04 | 1.531E+04 | **8.215E+03** |
| F50 | Mean | **2.203E+03** | 2.204E+03 | 2.204E+03 | **2.407E+03** | 2.472E+03 | 2.465E+03 |
| | Std | **5.571E+00** | 6.656E+00 | 5.901E+00 | 1.183E+02 | 1.236E+02 | **1.069E+02** |
| F51 | Mean | **2.629E+03** | **2.629E+03** | **2.629E+03** | **2.615E+03** | **2.615E+03** | **2.615E+03** |
| | Std | 9.636E−13 | **9.526E−13** | **9.526E−13** | 1.949E−06 | 1.017E−06 | **2.550E−12** |
| F52 | Mean | 2.512E+03 | 2.513E+03 | **2.511E+03** | 2.626E+03 | 2.625E+03 | **2.613E+03** |
| | Std | **2.300E+00** | 3.015E+00 | 3.307E+00 | **2.621E+00** | 3.152E+00 | 1.004E+01 |
| F53 | Mean | 2.656E+03 | 2.641E+03 | **2.628E+03** | 2.710E+03 | 2.704E+03 | **2.702E+03** |
| | Std | 3.594E+01 | 2.591E+01 | **2.263E+01** | 3.177E+00 | 4.591E+00 | **3.110E+00** |
| F54 | Mean | **2.700E+03** | **2.700E+03** | **2.700E+03** | 2.701E+03 | 2.701E+03 | **2.700E+03** |
| | Std | 3.983E−02 | 5.321E−02 | **3.930E−02** | 1.226E+00 | 6.954E−01 | **2.464E−01** |
| F55 | Mean | 2.960E+03 | 2.977E+03 | **2.871E+03** | 3.405E+03 | 3.283E+03 | **3.181E+03** |
| | Std | **1.534E+02** | 1.747E+02 | 1.862E+02 | 3.403E+02 | 2.683E+02 | **1.999E+02** |
| F56 | Mean | 3.217E+03 | 3.214E+03 | **3.180E+03** | 4.044E+03 | 4.032E+03 | **3.917E+03** |
| | Std | **2.623E+01** | 2.631E+01 | 4.017E+01 | **3.069E+02** | 3.587E+02 | 3.318E+02 |
| F57 | Mean | 7.235E+04 | 3.726E+04 | **3.173E+03** | 7.676E+06 | 8.027E+06 | **5.541E+06** |
| | Std | 3.413E+05 | 2.401E+05 | **7.334E+01** | 1.641E+06 | **1.172E+06** | 4.039E+06 |
| F58 | Mean | **3.524E+03** | 3.638E+03 | 3.535E+03 | 7.471E+04 | 5.685E+04 | **2.423E+04** |
| | Std | **1.240E+02** | 5.199E+02 | 1.390E+02 | 7.727E+04 | 6.339E+04 | **3.561E+04** |

and F50) and four (i.e. F37, F38, F39, F46) functions, respectively. However, BSA_SRL outperforms BSA and BSA_OBL on 20 (i.e. F29, F30, F31, F32, F33, F34, F36, F37, F39, F43, F46, F47, F49, F52, F53, F54, F55, F56, F57 and F58) and 18 (i.e. F29, F30, F31, F32, F34, F36, 45, F47, F48, F49, F50, F52, F53, F54, F55, F56, F57 and F58) functions, respectively. In addition, BSA_SRL can find the same solutions with BSA and BSA_OBL on six (i.e. F35, F40, F41, F42, F44 and F51) and eight (F33, F35, F40, F41, F42, F43, F44 and F51) functions, respectively.

As can be seen from the statistical results of Wilcoxon signed ranks test presented in Table 4, BSA_SRL outperforms BSA on nine functions with 10-dimensional (i.e. F29, F33, F34, F38, F42, F53,

**Table 4**
Results produced by Wilcoxon signed ranks test for CEC 2014 test suite.

| No. | 10-dimensional | | | | | | | | 30-dimensional | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BSA_SRL vs. BSA | | | | BSA_SRL vs. BSA_OBL | | | | BSA_SRL vs. BSA | | | | BSA_SRLvs. BSA_OBL | | | |
| | R+ | R- | *P*-value | S | R+ | R- | *P*-value | S | R+ | R- | *P*-value | S | R+ | R- | *P*-value | S |
| F29 | 1199 | 76 | 5.95E−8 | + | 865 | 410 | 2.81E−2 | + | 1254 | 21 | 2.66E−9 | + | 1263 | 12 | 1.56E−9 | + |
| F30 | 638 | 637 | 1.00E+0 | = | 637.5 | 637.5 | 1.00E+0 | = | 1275 | 0 | 7.56E−10 | + | 1275 | 0 | 7.56E−10 | + |
| F31 | 636 | 639 | 5.00E−1 | = | 637.5 | 637.5 | 1.00E+0 | = | 1106 | 169 | 6.11E−6 | + | 1065 | 210 | 3.68E−5 | + |
| F32 | 698.5 | 576.5 | 2.32E−1 | = | 705.5 | 569.5 | 1.02E−1 | = | 1189 | 86 | 1.02E−7 | + | 1010 | 265 | 3.23E−4 | + |
| F33 | 1087 | 188 | 1.43E−5 | + | 804 | 471 | 1.08E−1 | = | 1196 | 79 | 6.99E−8 | + | 880 | 395 | 1.92E−2 | + |
| F34 | 1049 | 226 | 3.40E−6 | + | 998.5 | 276.5 | 6.36E−6 | + | 1206 | 69 | 4.07E−8 | + | 956 | 319 | 2.11E−3 | + |
| F35 | 473 | 802 | 1.02E−1 | = | 697 | 578 | 5.66E−1 | = | 351 | 924 | 5.68E−3 | − | 545 | 730 | 3.72E−1 | = |
| F36 | 647 | 628 | 3.54E−1 | = | 700.5 | 574.5 | 3.67E−3 | + | 824 | 451 | 7.18E−2 | = | 855 | 420 | 3.58E−2 | + |
| F37 | 682.5 | 592.5 | 6.54E−1 | = | 890 | 385 | 1.20E−2 | + | 748 | 527 | 2.86E−1 | = | 589 | 686 | 6.40E−1 | = |
| F38 | 1202 | 73 | 5.06E−8 | + | 515 | 760 | 2.37E−1 | = | 503 | 772 | 1.94E−1 | = | 602 | 673 | 7.32E−1 | = |
| F39 | 471 | 804 | 1.08E−1 | = | 729 | 546 | 3.77E−1 | = | 1031 | 244 | 1.46E−4 | + | 272 | 1003 | 4.18E−4 | − |
| F40 | 684 | 591 | 6.54E−1 | = | 496 | 779 | 1.72E−1 | = | 775 | 500 | 1.84E−1 | = | 405 | 870 | 2.48E−2 | − |
| F41 | 313 | 962 | 1.73E−3 | − | 1020 | 255 | 2.22E−4 | + | 72 | 1203 | 4.79E−8 | − | 223 | 1052 | 6.30E−5 | − |
| F42 | 1141 | 134 | 1.17E−6 | + | 1081 | 194 | 1.86E−5 | + | 324 | 951 | 2.48E−3 | − | 426 | 849 | 4.12E−2 | − |
| F43 | 575 | 700 | 5.46E−1 | = | 772 | 503 | 1.94E−1 | = | 869 | 406 | 2.54E−2 | + | 681 | 594 | 6.75E−1 | = |
| F44 | 736 | 539 | 3.42E−1 | = | 803 | 472 | 1.10E−1 | = | 912 | 363 | 8.05E−3 | + | 644 | 631 | 9.50E−1 | = |
| F45 | 555 | 720 | 4.26E−1 | = | 589 | 686 | 6.40E−1 | = | 535 | 740 | 3.22E−1 | = | 697 | 578 | 5.66E−1 | = |
| F46 | 550 | 725 | 3.98E−1 | = | 576 | 699 | 5.53E−1 | = | 707 | 568 | 5.02E−1 | = | 456 | 819 | 7.98E−2 | = |
| F47 | 452 | 823 | 7.33E−2 | = | 747 | 528 | 2.90E−1 | = | 832 | 443 | 6.04E−2 | = | 724 | 551 | 4.04E−1 | = |
| F48 | 539 | 736 | 3.42E−1 | = | 367 | 908 | 9.02E−3 | − | 495 | 780 | 1.69E−1 | = | 700 | 575 | 5.46E−1 | = |
| F49 | 341 | 934 | 4.21E−3 | − | 582 | 693 | 5.92E−1 | = | 676 | 599 | 7.10E−1 | = | 665 | 610 | 7.91E−1 | = |
| F50 | 372 | 903 | 1.04E−2 | − | 588 | 687 | 6.33E−1 | = | 348 | 927 | 5.20E−3 | − | 644 | 631 | 9.50E−1 | = |
| F51 | 637.5 | 637.5 | 1.00E+0 | = | 637.5 | 637.5 | 1.00E+0 | = | 1275 | 0 | 7.56E−10 | + | 1275 | 0 | 7.56E−10 | + |
| F52 | 830 | 445 | 6.31E−2 | = | 903 | 372 | 1.04E−2 | + | 1222 | 53 | 1.68E−8 | + | 1190 | 85 | 9.64E−8 | + |
| F53 | 1085 | 190 | 1.56E−5 | + | 925 | 350 | 5.51E−3 | + | 1266 | 9 | 1.30E−9 | + | 913 | 362 | 1.86E−3 | + |
| F54 | 602 | 673 | 7.32E−1 | = | 1012 | 263 | 3.00E−4 | + | 554 | 721 | 4.20E−1 | = | 562 | 713 | 4.66E−1 | = |
| F55 | 844 | 431 | 4.62E−2 | + | 947 | 328 | 2.81E−3 | + | 1097 | 178 | 9.18E−6 | + | 976 | 299 | 1.08E−3 | + |
| F56 | 1184 | 91 | 1.32E−7 | + | 1115 | 160 | 4.04E−6 | + | 754 | 521 | 2.61E−1 | = | 776 | 499 | 1.81E−1 | = |
| F57 | 1004 | 271 | 4.03E−4 | + | 988 | 287 | 7.16E−4 | + | 866 | 409 | 2.30E−2 | + | 905 | 370 | 2.53E−3 | + |
| F58 | 594 | 681 | 6.75E−1 | = | 753 | 522 | 2.65E−1 | = | 1055 | 220 | 5.57E−5 | + | 957 | 318 | 2.04E−3 | + |
| +/=/- | 9/18/3 | | | | 12/17/1 | | | | 15/11/4 | | | | 13/13/4 | | | |

F55, F56 and F57) and 15 functions with 30-dimensional (i.e. F29, F30, F31, F32, F33, F34, F39, F43, F44, F51, F52, F53, F55, F57 and F58). BSA only can give better solutions than BSA_SRL on three functions with 10-dimensional (i.e. F41, F49 and F50) and four functions with 30-dimensional (i.e. F35, F41, F42 and F50). In addition, BSA_OBL cannot compete with BSA_SRL on 12 functions with 10-dimensional (i.e. F29, F34, F36, F37, F41, F42, F52, F53, F54, F55, F56 and F57) and 13 functions with 30-dimensional (i.e. F29, F30, F31, F32, F33, F34, F36, F51, F52, F53, F55, F57 and F58). Note that, BSA_SRL only is inferior to BSA_OBL on F48 with 10-dimensional and four functions with 30-dimensional (i.e. F39, F40, F41 and F42).

### 4.3. Experimental results for CEC 2017

The comparison results among BSA, BSA_OBL and BSA_SRL for CEC 2017 test suite are given in Fig. 6. According to Fig. 6, BSA_SRL outperforms BSA and BSA_OBL on more than sixty percent of test functions with 10-dimensional and 30-dimensional.

By observing Table 5, for test functions with 10-dimensional, BSA and BSA_OBL can offer the same solution with BSA_SRL on eight (i.e. F59, F60, F61, F64, F67, F72, F76 and F77) and 12 (i.e. F59, F60, F61, F64, F67, F69, F71, F72, F73, F77, F78 and F87) functions, respectively. In addition, BSA_SRL is only inferior to BSA and BSA_OBL on five (i.e. F69, F73, F75, F83 and F87) and four (i.e. F70, F76, F83 and F84) functions, respectively. Note that, BSA_SRL outperforms BSA and BSA_SRL on the rest functions.

As presented in Table 5, for test functions with 30-dimensional, BSA_SRL cannot beat BSA on only seven functions (i.e. F60, F66, F67, F74, F76, F77 and F79) while BSA_SRL is superior to BSA on 22 functions (i.e. F59, F61, F62, F63, F65, F68, F69, F70, F71, F72, F73, F75, F78, F80, F81, F82, F83, F84, F85, F86, F87 and F88). Although BSA_OBL shows better performance than BSA_SRL on 11 functions (i.e. F60, F63, F65, F67, F69, F71, F72, F74, F76, F79 and F83), BSA_OBL cannot compete with BSA_SRL on more than half of functions, including F59, F61, F62, F66, F68, F70, F73, F75, F77, F78, F80, F81, F82, F84, F85, F86, F87 and F88.

From the statistical results of Wilcoxon signed ranks test shown in Table 6, BSA_SRL can give better solutions than BSA on 16 functions with 10-dimensional (i.e. F59, F60, F61, F62, F63, F65, F66, F68, F78, F79, F81, F82, F84, F85, F86 and F88) and 13 functions with 30-dimensional (i.e. F59, F60, F62, F64, F68, F69, F70, F71, F78, F80, F82, F86 and F87). However, BSA is only superior to BSA_SRL on four functions with 10-dimensional (i.e. F69, F73, F85 and F87) and four functions with 30-dimensional (i.e. F67, F74, F76 and F77), respectively. In addition, BSA_OBL only can win BSA_SRL on two functions with 10-dimensional (i.e. F73 and F83) and F67 with 30-dimensional. But BSA_SRL outperforms BSA_OBL on seven functions with 10-dimensional (i.e. F62, F63, F65, F66, F82, F85 and F88) and 11 functions with 30-dimensional (i.e. F59, F60, F62, F64, F68, F70, F77, F80, F86, F87 and F88).

### 4.4. Discussion for the effectiveness of specular reflection learning

This section discusses the effectiveness of specular reflection learning introduced to BSA based on the experimental results shown in Sections 4.1–4.3.

As mentioned previously, opposition-based learning can be regarded as one special case of specular reflection learning. Thus, specular reflection learning also can be called generalized opposition-based learning. The only difference between opposition-based learning and specular reflection learning is the way of achieving opposite solutions. More specifically, according to Eq. (15), $\lambda$ is a random number between 0 and 2 in specular reflection learning. In addition, if $\lambda$ is equal to 1, specular reflection learning will turn to opposition-based learning. For a solution $\boldsymbol{p}$, its opposite solution $\breve{\boldsymbol{p}}$ is "static" in the opposition-based learning while its opposite solution $\breve{\boldsymbol{p}}$ is "dynamic" in the specular reflection learning. Further, "static" means only one candidate solution and "dynamic" indicates more than one candidate solutions. In other words, specular reflection learning has stronger local search ability compared with opposition-based learning, which is helpful for specular reflection learning to improve population diversity. This advantage of specular reflection learning has been proven in the designed experiments. The used test functions include 11 unimodal functions and 77 multimodal functions (including simple multimodal functions, hybrid functions and composition

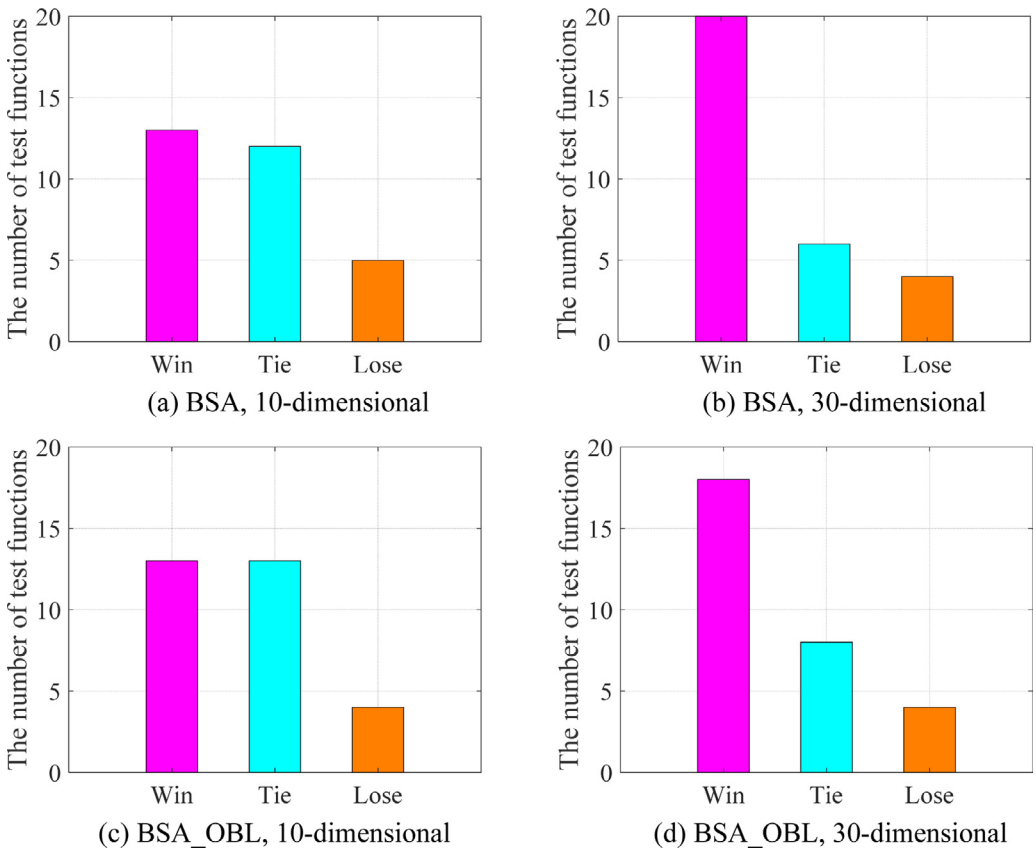


**Fig. 5.** The comparison results among BSA, BSA_OBL and BSA_SRL for CEC 2014 test suite. "Win", "Tie" and "Lose" means the number of test functions that BSA_SRL is superior, equal and inferior to the compared algorithm, respectively.

functions). According to Table 1, Table 3 and Table 5, Table 7 displays the statistical results obtained by BSA, BSA_OBL and BSA_SRL for all test functions. From the last column of Table 7, for each test dimension, BSA_SRL all can offer better solutions than BSA and BSA_OBL on more than fifty percent of test functions. That is, specular reflection learning is more effective than opposition-based learning for improving BSA.

## 5. BSA_SRL for real-world constrained engineering optimization

In this section, the proposed BSA_SRL is used to solve two challenging real-world constrained engineering optimization problems including rolling element bearing design problem and pressure vessel design problem. The constraint handling method is based on the penalty function approach as done in [30]. In addition, population size and the maximum number of function evaluations were set to 50 and 50,000, respectively. 50 independent runs were executed for each test problem and the optimal results were recorded. Besides, to prove the competitiveness of BSA_SRL, the solutions obtained by BSA_SRL were compared with some recently reported solutions.

### 5.1. Rolling element bearing design problem

The objective of this problem is to maximize the dynamic load carting capacity of a rolling element bearing. The detailed discussion about this problem can be found in [31]. 10 design variables $\boldsymbol{x} = [D_m, D_b, Z, f_i, f_o, K_{Dmin}, K_{Dmax}, \varepsilon, e, \zeta]$ are considered for this problem where$D_m$is the pitch diameter,$D_b$is the ball diameter,$Z$is the number of balls,$f_i$is the inner raceway curvature coefficients, $f_o$is the outer raceway curvature coefficients, $K_{Dmin}$is the minimum ball diameter limiter, $K_{Dmax}$is the maximum ball diameter limiter, $\varepsilon$is the parameter for outer ring strength consideration,$e$is the parameter for mobility condition, $\zeta$ is the bearing width limiter. Note that $Z$ is a discrete design variable and the rest variables are continuous design variables. Nine subjected constraints are imposed based on kinematic and manufacturing considerations. This problem can be defined as

$$\text{Maximum } C_d = \begin{cases} f_c Z^{2/3} D_b^{1.8}, & \text{if } D \le 25.4 \text{ mm} \\ 3.647 f_c Z^{2/3} D_b^{1.4}, & \text{if } D > 25.4 \text{ mm} \end{cases} \tag{21}$$

Subject to:

$$g_1(\boldsymbol{x}) = \frac{\phi_0}{2\sin^{-1}(D_b/D_m)} - Z + 1 \ge 0 \tag{22}$$

$$g_2(\boldsymbol{x}) = 2D_b - K_{D\min}(D-d) \ge 0 \tag{23}$$

$$g_3(\boldsymbol{x}) = K_{D\max}(D-d) - 2D_b \ge 0 \tag{24}$$

$$g_4(\boldsymbol{x}) = \zeta B_w - D_b \le 0 \tag{25}$$

$$g_5(\boldsymbol{x}) = D_m - 0.5(D+d) \ge 0 \tag{26}$$

$$g_6(\boldsymbol{x}) = (0.5+e)(D+d) - D_m \ge 0 \tag{27}$$

$$g_7(\boldsymbol{x}) = 0.5(D - D_m - D_b) - \varepsilon D_b \ge 0 \tag{28}$$

$$g_8(\boldsymbol{x}) = f_i \ge 0.515 \tag{29}$$

$$g_9(\boldsymbol{x}) = f_o \ge 0.515 \tag{30}$$

where

$$\gamma = \frac{D_b}{D_m}, f_i = \frac{r_i}{D_b}, f_o = \frac{r_o}{D_b}, T = D - d - 2D_b, D = 160,$$
$$d = 90, B_w = 30, D = 160, r_i = r_o = 11.033$$
$$0.5(D+d) \le D_m \le 0.6(D+d),$$

**Table 5**
Experimental results obtained by BSA, BSA_OBL and BSA_SRL for CEC 2017 test suite.

| No. | Indicator | 10-dimensional | | | 30-dimensional | | |
|---|---|---|---|---|---|---|---|
| | | BSA | BSA_OBL | BSA_SRL | BSA | BSA_OBL | BSA_SRL |
| F59 | Mean | **1.000E+02** | **1.000E+02** | **1.000E+02** | 2.405E+03 | 4.709E+03 | **1.383E+03** |
| | Std | 2.456E−11 | 3.004E−07 | **8.370E−15** | **2.365E+03** | 4.991E+03 | 2.951E+03 |
| F60 | Mean | **2.000E+02** | **2.000E+02** | **2.000E+02** | **7.258E+13** | 7.383E+21 | 1.119E+22 |
| | Std | 1.201E−08 | 5.076E−08 | **2.353E−09** | **3.890E+14** | 4.972E+22 | 7.834E+22 |
| F61 | Mean | **3.000E+02** | **3.000E+02** | **3.000E+02** | 1.053E+04 | 1.081E+04 | **1.034E+04** |
| | Std | 1.467E−11 | 1.816E−14 | **1.989E−14** | **4.070E+03** | 4.949E+03 | 4.511E+03 |
| F62 | Mean | 4.004E+02 | 4.002E+02 | **4.000E+02** | 4.900E+02 | 4.897E+02 | **4.643E+02** |
| | Std | 2.741E−01 | 1.078E−01 | **2.297E−03** | 2.466E+01 | **2.209E+01** | 2.673E+01 |
| F63 | Mean | 5.052E+02 | 5.048E+02 | **5.040E+02** | 5.678E+02 | **5.658E+02** | 5.667E+02 |
| | Std | 2.904E+00 | **1.836E+00** | 2.576E+00 | 1.755E+01 | **1.519E+01** | 2.049E+01 |
| F64 | Mean | **6.000E+02** | **6.000E+02** | **6.000E+02** | **6.000E+02** | **6.000E+02** | **6.000E+02** |
| | Std | **5.387E−14** | 6.890E−14 | 7.956E−14 | 8.411E−02 | **7.608E−03** | 1.671E−02 |
| F65 | Mean | 7.147E+02 | 7.152E+02 | **7.139E+02** | 8.237E+02 | **8.107E+02** | 8.167E+02 |
| | Std | 1.853E+00 | 2.039E+00 | **1.796E+00** | 2.341E+01 | 1**.619E+01** | 2.356E+01 |
| F66 | Mean | 8.048E+02 | 8.049E+02 | **8.036E+02** | **8.586E+02** | 8.633E+02 | 8.605E+02 |
| | Std | **1.634E+00** | 1.999E+00 | 1.745E+00 | **1.365E+01** | 1.768E+01 | 1.418E+01 |
| F67 | Mean | **9.000E+02** | **9.000E+02** | **9.000E+02** | **9.892E+02** | 1.000E+03 | 1.101E+03 |
| | Std | **0.000E+00** | **0.000E+00** | 2.570E−01 | **8.410E+01** | 1.361E+02 | 1.891E+02 |
| F68 | Mean | 1.667E+03 | 1.583E+03 | **1.522E+03** | 6.813E+03 | 6.249E+03 | **5.301E+03** |
| | Std | 2.692E+02 | 2.167E+02 | **1.517E+02** | 9.408E+02 | 1.203E+03 | **8.636E+02** |
| F69 | Mean | **1.101E+03** | 1.102E+03 | 1.102E+03 | 1.170E+03 | 1.151E+03 | **1.154E+03** |
| | Std | 1.409E+00 | **1.224E+00** | 1.927E+00 | 3.121E+01 | 3.116E+01 | **2.543E+01** |
| F70 | Mean | 1.996E+03 | **1.364E+03** | 1.422E+03 | 8.379E+04 | 7.323E+04 | **4.158E+04** |
| | Std | 4.286E+03 | **9.977E+01** | 2.438E+02 | 8.736E+04 | 5.317E+04 | **2.102E+04** |
| F71 | Mean | 1.306E+03 | **1.305E+03** | **1.305E+03** | 2.400E+04 | **1.481E+04** | 1.505E+04 |
| | Std | **1.900E+00** | 2.577E+00 | 3.069E+00 | 1.543E+04 | 1.578E+04 | **1.507E+04** |
| F72 | Mean | **1.401E+03** | **1.401E+03** | **1.401E+03** | 1.464E+03 | **1.461E+03** | 1.462E+03 |
| | Std | 1.132E+00 | 1.105E+00 | **8.268E−01** | **2.049E+01** | 2.375E+01 | 2.139E+01 |
| F73 | Mean | **1.500E+03** | 1.501E+03 | 1.501E+03 | 2.102E+03 | 2.625E+03 | **2.101E+03** |
| | Std | **4.874E−01** | 5.495E−01 | 8.139E−01 | **9.515E+02** | 2.059E+03 | 9.863E+02 |
| F74 | Mean | 1.606E+03 | 1.606E+03 | **1.604E+03** | 2.131E+03 | **2.182E+03** | 2.224E+03 |
| | Std | 1.568E+01 | 1.755E+01 | **9.311E+00** | 1.991E+02 | 2.135E+02 | **1.753E+02** |
| F75 | Mean | **1.707E+03** | 1.714E+03 | 1.710E+03 | 1.847E+03 | 1.830E+03 | **1.821E+03** |
| | Std | **5.770E+00** | 1.061E+01 | 6.907E+00 | 9.160E+01 | 9.012E+01 | **8.735E+01** |
| F76 | Mean | 1.801E+03 | **1.800E+03** | 1.801E+03 | **1.478E+04** | 1.499E+04 | 1.819E+04 |
| | Std | **5.287E−01** | 5.893E−01 | 8.905E−01 | 3.139E+04 | 1.614E+04 | **1.529E+04** |
| F77 | Mean | **1.900E+03** | **1.900E+03** | **1.900E+03** | **2.006E+03** | 3.836E+03 | 2.911E+03 |
| | Std | 2.442E−01 | **3.976E−02** | 2.749E−01 | **1.472E+02** | 3.197E+03 | 2.071E+03 |
| F78 | Mean | 2.002E+03 | **2.001E+03** | **2.001E+03** | 2.403E+03 | 2.318E+03 | **2.261E+03** |
| | Std | 4.064E+00 | **1.024E+00** | 1.058E+00 | 2.756E+02 | **1.472E+02** | 1.615E+02 |
| F79 | Mean | 2.271E+03 | 2.205E+03 | **2.201E+03** | **2.347E+03** | 2.350E+03 | 2.351E+03 |
| | Std | 4.974E+01 | 2.959E+01 | **1.297E+00** | **1.317E+01** | 1.393E+01 | 2.528E+01 |
| F80 | Mean | 2.300E+03 | 2.291E+03 | **2.289E+03** | 7.384E+03 | 5.459E+03 | **2.372E+03** |
| | Std | **6.065E+00** | 2.704E+01 | 2.953E+01 | 1.481E+03 | 2.481E+03 | **5.082E+02** |
| F81 | Mean | 2.615E+03 | 2.613E+03 | **2.611E+03** | 2.714E+03 | 2.704E+03 | **2.698E+03** |
| | Std | 8.400E+00 | **4.788E+00** | 5.429E+00 | 5.959E+01 | 1.574E+01 | **1.508E+01** |
| F82 | Mean | 2.737E+03 | 2.722E+03 | **2.701E+03** | 2.885E+03 | 2.878E+03 | **2.870E+03** |
| | Std | **3.463E+01** | 6.623E+01 | 8.200E+01 | 3.252E+01 | 2.302E+01 | **1.415E+01** |
| F83 | Mean | 2.920E+03 | **2.914E+03** | 2.928E+03 | 2.896E+03 | **2.889E+03** | 2.893E+03 |
| | Std | 2.311E+01 | **2.219E+01** | 2.246E+01 | 1.537E+01 | 3.520E+00 | **1.258E+01** |
| F84 | Mean | 2.918E+03 | **2.895E+03** | 2.902E+03 | 4.481E+03 | 4.350E+03 | **4.338E+03** |
| | Std | 2.856E+01 | 4.308E+01 | **9.333E+00** | 3.887E+02 | **2.527E+02** | 3.857E+02 |
| F85 | Mean | 3.091E+03 | 3.092E+03 | **3.090E+03** | 3.235E+03 | 3.232E+03 | **3.228E+03** |
| | Std | 2.857E+00 | 3.138E+00 | **1.439E+00** | 3.383E+01 | 4.718E+01 | **2.297E+01** |
| F86 | Mean | 3.254E+03 | 3.158E+03 | **3.132E+03** | 3.697E+03 | 3.404E+03 | **3.166E+03** |
| | Std | 1.242E+02 | **8.213E+01** | 8.487E+01 | 1.118E+03 | 7.950E+02 | **5.373E+01** |
| F87 | Mean | **3.139E+03** | 3.149E+03 | 3.149E+03 | 4.091E+03 | 3.909E+03 | **3.673E+03** |
| | Std | **5.455E+00** | 1.434E+01 | 1.090E+01 | 6.124E+02 | 5.581E+02 | **2.940E+02** |
| F88 | Mean | 7.088E+05 | 7.430E+05 | **3.635E+03** | 7.549E+03 | 8.651E+03 | **7.189E+03** |
| | Std | 4.532E+05 | 2.975E+05 | **1.893E+02** | 2.233E+03 | 2.725E+03 | **1.518E+03** |

$$0.15\,(D-d) \le D_b \le 0.45\,(D-d)\,,\ 4 \le Z \le 50,$$

$$0.515 \le f_i \le 0.6,\ 0.515 \le f_o \le 0.6$$

$$0.4 \le K_{D\min} \le 0.5, 0.6 \le K_{D\max} \le 0.7, 0.3 \le e \le 0.4,$$

$$0.02 \le \varepsilon \le 0.1, 0.6 \le \zeta \le 0.85$$

$$f_c = 37.91\left[1+\left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i\,(2f_o-1)}{f_o\,(2f_i-1)}\right)^{0.41}\right\}^{10/3}\right]^{-0.3} \times\left(\frac{\gamma^{0.3}\,(1-\gamma)^{1.39}}{(1+\gamma)^{1./3}}\right)\left[\frac{2f_i}{2f_i-1}\right]^{0.41}$$

**Table 6**
Results produced by Wilcoxon signed ranks test for CEC 2017 test suite.

| No. | 10-dimensional | | | | | | | | 30-dimensional | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BSA_SRL vs. BSA | | | | BSA_SRL vs. BSA_OBL | | | | BSA_SRL vs. BSA | | | | BSA_SRL vs. BSA_OBL | | | |
| | R+ | R- | *P*-value | S | R+ | R- | *P*-value | S | R+ | R- | *P*-value | S | R+ | R- | *P*-value | S |
| F59 | 1068 | 207 | 2.35E−8 | + | 639 | 636 | 5.00E−1 | = | 1014 | 261 | 2.79E−4 | + | 1068 | 207 | 3.24E−5 | + |
| F60 | 896 | 379 | 1.26E−2 | + | 687 | 588 | 6.33E−1 | = | 876 | 399 | 2.13E−2 | + | 1049 | 226 | 7.12E−5 | + |
| F61 | 826.5 | 448.5 | 2.49E−6 | + | 637.5 | 637.5 | 1.00E+0 | = | 674 | 601 | 7.25E−1 | = | 668 | 607 | 7.68E−1 | = |
| F62 | 1274 | 1 | 8.03E−10 | + | 1275 | 0 | 7.56E−10 | + | 1066 | 209 | 3.53E−5 | + | 1103 | 172 | 7.00E−6 | + |
| F63 | 869.5 | 405.5 | 2.51E−2 | + | 861 | 414 | 3.10E−2 | + | 694 | 581 | 5.85E−1 | = | 596 | 679 | 6.89E−1 | = |
| F64 | 637.5 | 637.5 | 1.00E+0 | = | 637.5 | 637.5 | 1.00E+0 | = | 1069 | 206 | 3.11E−5 | + | 935 | 340 | 4.08E−3 | + |
| F65 | 876 | 399 | 2.13E−2 | + | 953 | 322 | 2.32E−3 | + | 773 | 502 | 1.91E−1 | = | 519 | 756 | 2.53E−1 | = |
| F66 | 925 | 350 | 5.51E−3 | + | 895 | 380 | 1.29E−2 | + | 550 | 725 | 3.98E−1 | = | 721 | 554 | 4.20E−1 | = |
| F67 | 637 | 638 | 1.00E+0 | = | 637 | 638 | 1.00E+0 | = | 240 | 1035 | 1.24E−4 | − | 348 | 927 | 5.20E−3 | − |
| F68 | 895 | 380 | 1.29E−2 | + | 760 | 515 | 2.23E−1 | = | 1219 | 56 | 1.98E−8 | + | 1062 | 213 | 4.17E−5 | + |
| F69 | 392.5 | 882.5 | 1.80E−2 | − | 563 | 712 | 4.59E−1 | = | 929 | 346 | 4.89E−3 | + | 560 | 715 | 4.54E−1 | = |
| F70 | 682 | 593 | 6.68E−1 | = | 540 | 735 | 3.47E−1 | = | 1001 | 274 | 4.50E−4 | + | 1008 | 267 | 3.48E−4 | + |
| F71 | 833 | 442 | 5.91E−2 | = | 695 | 580 | 5.79E−1 | = | 908 | 367 | 9.02E−3 | + | 630 | 645 | 9.42E−1 | = |
| F72 | 751.5 | 523.5 | 1.22E−1 | = | 703 | 572 | 3.41E−1 | = | 645 | 630 | 9.42E−1 | = | 617 | 658 | 8.43E−1 | = |
| F73 | 374 | 901 | 1.10E−2 | − | 401 | 874 | 2.24E−2 | − | 689 | 586 | 6.19E−1 | = | 746 | 529 | 2.95E−1 | = |
| F74 | 667 | 608 | 7.76E−1 | = | 747 | 528 | 2.90E−1 | = | 394 | 881 | 1.87E−2 | − | 499 | 776 | 1.81E−1 | = |
| F75 | 426 | 849 | 4.12E−2 | − | 812 | 463 | 9.21E−2 | = | 798 | 477 | 1.21E−1 | = | 698 | 577 | 5.59E−1 | = |
| F76 | 698 | 577 | 5.59E−1 | = | 578 | 697 | 5.66E−1 | = | 336 | 939 | 3.61E−3 | − | 517 | 758 | 2.45E−1 | = |
| F77 | 595 | 680 | 6.82E−1 | = | 517 | 758 | 2.45E−1 | = | 296 | 979 | 9.79E−4 | − | 860 | 415 | 3.17E−2 | + |
| F78 | 892 | 383 | 1.14E−2 | + | 679 | 596 | 6.70E−1 | = | 943 | 332 | 3.19E−3 | + | 827 | 448 | 6.74E−2 | = |
| F79 | 1265 | 10 | 1.38E−9 | + | 629.5 | 645.5 | 7.95E−1 | = | 453 | 822 | 7.49E−2 | = | 473 | 802 | 1.12E−1 | = |
| F80 | 533 | 742 | 3.13E−1 | = | 487 | 788 | 1.46E−1 | = | 1273 | 2 | 8.53E−10 | + | 1233 | 42 | 9.00E−9 | + |
| F81 | 892 | 383 | 1.40E−2 | + | 808 | 467 | 9.98E−2 | = | 833 | 442 | 5.91E−2 | = | 834 | 441 | 5.78E−2 | = |
| F82 | 1209 | 66 | 3.45E−8 | + | 1073 | 202 | 2.62E−5 | + | 1009 | 266 | 3.36E−4 | + | 791 | 484 | 1.38E−1 | = |
| F83 | 461 | 814 | 8.84E−2 | = | 335 | 940 | 3.50E−3 | − | 681 | 594 | 6.75E−1 | = | 513 | 762 | 2.29E−1 | = |
| F84 | 709 | 566 | 1.12E−3 | + | 634.5 | 640.5 | 5.00E−1 | = | 826 | 449 | 6.88E−2 | = | 538 | 737 | 3.37E−1 | = |
| F85 | 903 | 372 | 2.73E−3 | + | 1031 | 244 | 5.44E−5 | + | 775 | 500 | 1.84E−1 | = | 632 | 643 | 9.58E−1 | = |
| F86 | 1115.5 | 159.5 | 9.33E−7 | + | 701.5 | 573.5 | 8.31E−2 | = | 1223 | 52 | 1.59E−8 | + | 1082 | 193 | 1.78E−5 | + |
| F87 | 139 | 1136 | 1.49E−6 | − | 623 | 652 | 8.89E−1 | = | 1020 | 255 | 2.22E−4 | + | 887 | 388 | 1.60E−2 | + |
| F88 | 1265 | 10 | 1.38E−9 | + | 1272 | 3 | 9.07E−10 | + | 675 | 600 | 7.17E−1 | = | 927 | 348 | 5.20E−3 | + |
| +/=/- | 16/10/4 | | | | 7/21/2 | | | | 13/13/4 | | | | 11/18/1 | | | |

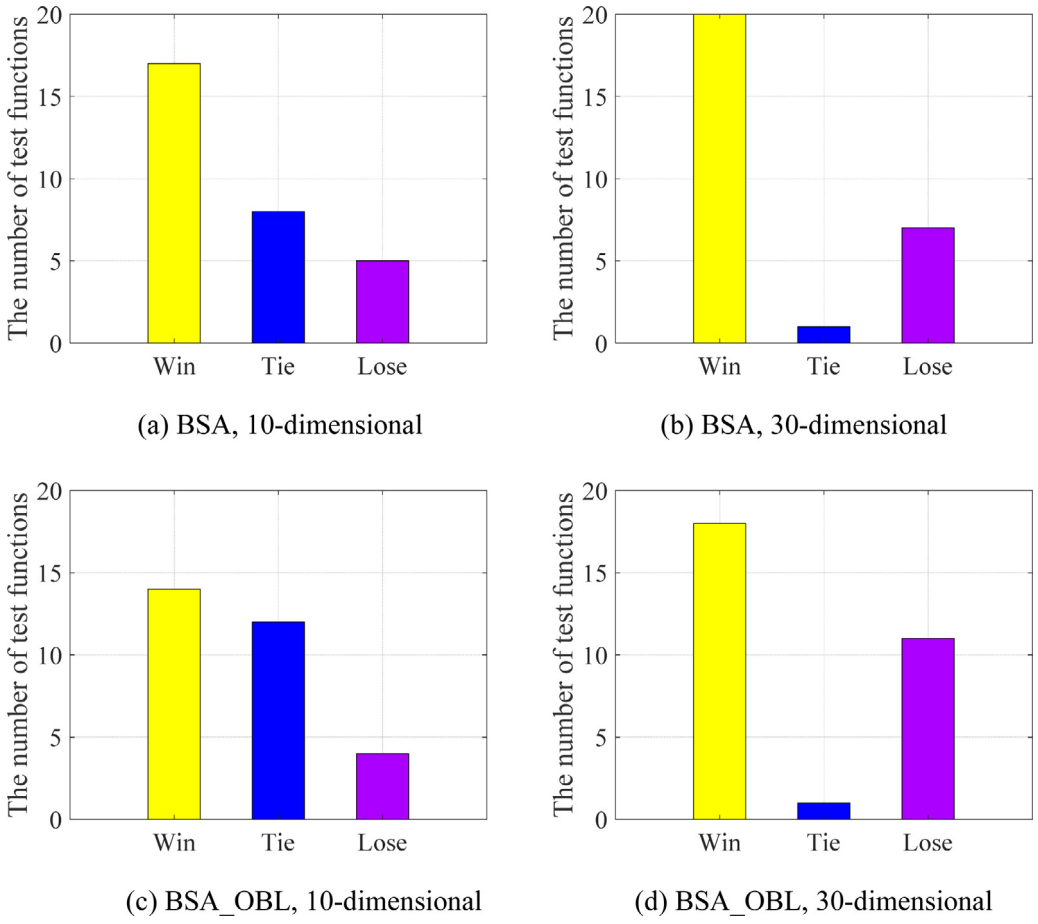


**Fig. 6.** The comparison results among BSA, BSA_OBL and BSA_SRL for CEC 2017 test suite. "Win", "Tie" and "Lose" means the number of test functions that BSA_SRL is superior, equal and inferior to the compared algorithm, respectively.

**Table 7**
The statistical results obtained by BSA, BSA_OBL and BSA_SRL for all test functions. "Win", "Tie" and "Lose" means the number of test functions that BSA_SRL is superior, equal and inferior to the compared algorithms, respectively.

| Case | BSA_SRL vs. BSA | | | | | | BSA_SRL vs. BSA_OBL | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10-dimensional | | | 30-dimensional | | | 10-dimensional | | | 30-dimensional | | |
| | Win | Tie | Lose | Win | Tie | Lose | Win | Tie | Lose | Win | Tie | Lose |
| CEC 2013 | 17 | 5 | 6 | 19 | 3 | 6 | 18 | 5 | 5 | 18 | 3 | 7 |
| CEC 2014 | 13 | 12 | 5 | 20 | 6 | 4 | 13 | 13 | 4 | 18 | 8 | 4 |
| CEC 2017 | 17 | 8 | 5 | 22 | 1 | 7 | 14 | 12 | 4 | 18 | 1 | 11 |
| Total | 47 | 25 | 16 | 61 | 10 | 17 | 45 | 30 | 13 | 54 | 12 | 22 |

**Table 8**
The optimal solutions obtained by BSA_SRL and the compared algorithms for rolling element bearing design problem.

| Algorithm | TLBO | EPO | SOA | WCA | MBA | SHO | BSA_SRL |
|---|---|---|---|---|---|---|---|
| $D_m$ | 125.7191 | 125 | 125 | 125.721167 | 125.7153 | 125 | 125.71906 |
| $D_b$ | 21.42559 | 21.41890 | 21.41892 | 21.423300 | 21.423300 | 21.04732 | 21.425588 |
| $Z$ | 11 | 10.94113 | 10.94123 | 11.001030 | 11 | 10.93268 | 11.000000 |
| $f_i$ | 0.515 | 0.515 | 0.515 | 0.515000 | 0.515 | 0.515 | 0.5150000 |
| $f_o$ | 0.515 | 0.515 | 0.515 | 0.515000 | 0.515 | 0.515 | 0.5150000 |
| $K_{Dmin}$ | 0.424266 | 0.4 | 0.4 | 0.401514 | 0.488805 | 0.4 | 0.4000000 |
| $K_{Dmax}$ | 0.633948 | 0.7 | 0.7 | 0.659047 | 0.627829 | 0.7 | 0.7000000 |
| $\varepsilon$ | 0.3 | 0.3 | 0.3 | 0.300032 | 0.300149 | 0.3 | 0.3000000 |
| $e$ | 0.068858 | 0.02 | 0.02 | 0.040045 | 0.097305 | 0.02 | 0.1000000 |
| $\zeta$ | 0.799498 | 0.6 | 0.6 | 0.600000 | 0.646095 | 0.6 | 0.6261572 |
| Opt. cost | 81859.74 | 85067.983. | 85068.052 | 85538.48 | 85535.9611 | 85054.032 | **85549.223** |

$$\phi_0 = 2\pi - 2\cos^{-1} \times \frac{\left[\{(D-d)/2 - 3(T/4)\}^2 + \{D/2 - (T/4) - D_b\}^2 - \{d/2 + (T/4)\}^2\right]}{2\{(D-d)/2 - 3(T/4)\}\{D/2 - (T/4) - D_b\}}$$

Teaching–learning-based optimization (TLBO) [32], emperor penguin optimizer [33], Seagull optimization algorithm (SOA) [34], water cycle algorithm (WCA) [35], mine blast algorithm (MBA) [36], and spotted hyena optimizer (SHO) [37] have been employed to solve this problem. Table 8 shows the optimal solutions obtained by BSA_SRL and the compared algorithms. From Table 8, BSA_SRL can get the optimal cost, i.e. 85549.223. In addition, WCA and MBA show strong competitiveness, whose solutions are very close to that of BSA_SRL.

### 5.2. Pressure vessel design problem

This problem was proposed in [38], whose goal is to minimize the total cost consisting of material, forming, and welding of a cylindrical vessel. Four design variables $\boldsymbol{x} = [T_s, T_h, R, L]$ are included in this problem where$T_s$is the thickness of the shell, $T_h$is the thickness of the head,$R$is the inner radius, and $L$ is the length of the cylindrical section without considering the head. In addition, four subjected constraints need to be considered. This problem can be expressed as

$$\text{Minimize} f(\boldsymbol{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (31)$$

Subject to

$$g_1(\boldsymbol{x}) = -x_1 + 0.0193x_3 \le 0 \quad (32)$$

$$g_2(\boldsymbol{x}) = -x_2 + 0.00954x_3 \le 0 \quad (33)$$

$$g_3(\boldsymbol{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296,000 \le 0 \quad (34)$$

$$g_4(\boldsymbol{x}) = x_4 - 240 \le 0 \quad (35)$$

where $0 \le x_i \le 100, i = 1, 2$; $10 \le x_i \le 200, i = 3, 4$.

This problem has been solved by several algorithms including SHO, moth flame optimization (MFO) [39], sine cosine algorithm (SCA) [40], gray wolf optimizer (GWO) [41], multi-verse optimizer (MVO) [42], hybrid GSA-GA algorithm (GSA-GA) [43], and crow search algorithm (CSA) [44] . Table 9 presents the obtained optimal solution by BSA_SRL and the compared algorithms. From Table 9, BSA_SRL can get the optimal cost, i.e. 5885.33277. In addition, SHO and GSA-GA can find solutions that are very close to the solution of BSA_SRL.

## 6. Managerial insights

Specular reflection learning is a very simple technique for improving metaheuristic methods, which has great potential to be applied to a lot of metaheuristic methods. When specular reflection learning is used for improving one metaheuristic method, an individual with better fitness value has more chance to be selected into the next generation population. Thus specular reflection learning may not be able to find promising results for metaheuristic methods with multiple evaluation indicators, such as particle swarm optimization and neural network algorithm. For particle swarm optimization, its each individual has two evaluation indicators, i.e. the current fitness value and the historical best fitness value; for neural network algorithm, its each individual also has two evaluation indicators, i.e. the current fitness value and the weight vector. Obviously, when specular reflection learning is employed by metaheuristic methods with multiple evaluation indicators, there may be confused among multiple indicators of each individual in these methods, which is not helpful for these methods to find global optimal solutions. Note that most reported metaheuristic methods only have an evaluation indicator, which means specular reflection learning still has a great applicative value in the field of metaheuristics.

In this work, BSA_SRL shows excellent performance on the two classical engineering design problems. However, the superiority of BSA_SRL on more complex engineering problems needs to be confirmed. In recent years, solving optimization problems about supply chains has been a hot and challenge topic. In short, optimization problems about supply chains are to find optimal solutions with the limited resources and the constraints. Given the excellent performance of BSA_SRL, one of our future projects is to use BSA_SRL to solve optimization problems about supply chains, such as the total cost of three-level supply chains [45], the optimal supply chain batch-sizing policy that collectively embodies green policies and a vendor-managed inventory with consignment stock agreement [46], the selective maintenance model

**Table 9**
The best solutions obtained by BSA_SRL and the compared algorithms for pressure vessel design problem.

| Algorithm | Variables | | | | Opt. cost |
|---|---|---|---|---|---|
| | $T_s$ | $T_h$ | $R$ | $L$ | |
| SHO | 0.778210 | 0.384889 | 40.315040 | 200.0000 | 5885.5773 |
| MFO | 0.835241 | 0.409854 | 43.578621 | 152.21520 | 6055.6378 |
| MVO | 0.845719 | 0.418564 | 43.816270 | 156.38164 | 6011.5148 |
| GWO | 0.779035 | 0.384660 | 40.327793 | 199.65029 | 5889.3689 |
| SCA | 0.817577 | 0.417932 | 41.74939 | 183.57270 | 6137.3724 |
| GSA-GA | 0.77819652 | 0.3846644 | 40.3210580446 | 199.9799646 | 5885.38533633 |
| CSA | 0.8125 | 0.4375 | 42.0984456 | 176.6365958 | 6059.7143348 |
| BSA_SRL | 0.7781686 | 0.3846492 | 40.319618754 | 199.99999959 | **5885.33277** |

**Table A.1**
The definition for CEC2013 test suite.

| No. | Type | Name | Dimension | Range | Optimum |
|---|---|---|---|---|---|
| F1 | Unimodal functions | Sphere function | 10/30 | [−100,100] | −1400 |
| F2 | | Rotated high conditioned elliptic function | 10/30 | [−100,100] | −1300 |
| F3 | | Rotated bent cigar function | 10/30 | [−100,100] | −1200 |
| F4 | | Rotated discus function | 10/30 | [−100,100] | −1100 |
| F5 | | Different powers function | 10/30 | [−100,100] | −1000 |
| F6 | Basic multimodal functions | Rotated Rosenbrock's function | 10/30 | [−100,100] | −900 |
| F7 | | Rotated Schaffers F7 function | 10/30 | [−100,100] | −800 |
| F8 | | Rotated Ackley's function | 10/30 | [−100,100] | −700 |
| F9 | | Rotated Weierstrass function | 10/30 | [−100,100] | −600 |
| F10 | | Rotated Griewank's function | 10/30 | [−100,100] | −500 |
| F11 | | Rastrigin's function | 10/30 | [−100,100] | −400 |
| F12 | | Rotated Rastrigin's function | 10/30 | [−100,100] | −300 |
| F13 | | Non-continuous rotated Rastrigin's function | 10/30 | [−100,100] | −200 |
| F14 | | Schwefel's function | 10/30 | [−100,100] | −100 |
| F15 | | Rotated Schwefel's function | 10/30 | [−100,100] | 100 |
| F16 | | Rotated Katsuura function | 10/30 | [−100,100] | 200 |
| F17 | | Lunacek bi_rastrigin function | 10/30 | [−100,100] | 300 |
| F18 | | Rotated lunacek bi_rastrigin function | 10/30 | [−100,100] | 400 |
| F19 | | Expanded Griewank's plus Rosenbrock's function | 10/30 | [−100,100] | 500 |
| F20 | | Expanded Scaffer's F6 function | 10/30 | [−100,100] | 600 |
| F21 | Composition functions | Composition function 1 (n = 5, rotated) | 10/30 | [−100,100] | 700 |
| F22 | | Composition function 2 (n = 3, unrotated) | 10/30 | [−100,100] | 800 |
| F23 | | Composition function 3 (n = 3, rotated) | 10/30 | [−100,100] | 900 |
| F24 | | Composition function 4 (n = 3, rotated) | 10/30 | [−100,100] | 1000 |
| F25 | | Composition function 5 (n = 3, rotated) | 10/30 | [−100,100] | 1100 |
| F26 | | Composition function 6 (n = 5, rotated) | 10/30 | [−100,100] | 1200 |
| F27 | | Composition function 7 (n = 5, rotated) | 10/30 | [−100,100] | 1300 |
| F28 | | Composition function 8 (n = 5, rotated) | 10/30 | [−100,100] | 1400 |

with stochastic maintenance quality for multi-component systems [47], the multi-objective multi-period sustainable location–allocation supply chain network model [48], the impact of emission costs on the replenishment order sizes and the total profit of a buyer in an imperfect supply process [49], the impact of ordering cost reduction and process quality improvement on an integrated vendor–buyer inventory system with variable lead time in just-in-time manufacturing environment [50], the supply chain model consisting of a single supplier and a single retailer [51], the joint economic lot-sizing problem [52], the game theoretic model for supply chain coordination problem [53], the model of supply chain consisting of a single manufacturer, a single distributor and a single retailer [54], the bi-objective multi-product constrained and integrated economic production quantity model [55], the closed-loop supply chain model with two suppliers, one manufacturer and one retailer [56], and an economic production quantity model of the replenishment [57].

## 7. Conclusion and future directions

This paper presents a novel technique, named specular reflection learning, for improving metaheuristic methods. Specular reflection learning is inspired by specular reflection phenomenon. The proposed specular reflection learning is closely related to opposition-based learning. Note that, opposition-based learning can be regarded as a special case of specular reflection learning. In order to examine the performance of specular reflection learning, the designed experiments focus on the performance differences of specular reflection learning and opposition-based learning used for improving backtracking search algorithm. Experimental results prove that specular reflection learning significantly outperforms opposition-based learning in improving the global search ability of backtracking search algorithm by solving CEC 2013, CEC 2014 and CEC 2017 test suites. In addition, the solutions of backtracking search algorithm with specular reflection learning are superior to the reported solutions on two real-world constrained engineering design problems. This demonstrates the effectiveness of specular reflection learning introduced to backtracking search algorithm.

Future research will focus on investigating the performance of specular reflection learning on other metaheuristic methods and evaluating the effectiveness of backtracking search algorithm with specular reflection learning on practical engineering problems about supply chains.

**Table A.2**
The definition for CEC2014 test suite.

| No. | Type | Name | Dimension | Range | Optimum |
|---|---|---|---|---|---|
| F29 | Unimodal functions | Rotated high conditioned elliptic function | 10/30 | [−100,100] | 100 |
| F30 | | Rotated bent cigar function | 10/30 | [−100,100] | 200 |
| F31 | | Rotated discus function | 10/30 | [−100,100] | 300 |
| F32 | Basic multimodal functions | Shifted and rotated Rosenbrock's function | 10/30 | [−100,100] | 400 |
| F33 | | Shifted and rotated Ackleys function | 10/30 | [−100,100] | 500 |
| F34 | | Shifted and rotated Weierstrass function | 10/30 | [−100,100] | 600 |
| F35 | | Shifted and rotated Griewank's function | 10/30 | [−100,100] | 700 |
| F36 | | Shifted Rastrigin's function | 10/30 | [−100,100] | 800 |
| F37 | | Six Hump Camel Back | 10/30 | [−100,100] | 900 |
| F38 | | Shifted and rotated Rastrigin's function | 10/30 | [−100,100] | 1000 |
| F39 | | Shifted and rotated Schwefel's function | 10/30 | [−100,100] | 1100 |
| F40 | | Shifted and rotated Katsuura function | 10/30 | [−100,100] | 1200 |
| F41 | | Shifted and rotated HappyCat function | 10/30 | [−100,100] | 1300 |
| F42 | | Shifted and rotated HGBat function | 10/30 | [−100,100] | 1400 |
| F43 | | Shifted and rotated Expanded Griewank's plus Rosenbrock's function | 10/30 | [−100,100] | 1500 |
| F44 | | Shifted and rotated Expanded Scaffer's F6 function | 10/30 | [−100,100] | 1600 |
| F45 | Hybrid functions | Hybrid function 1 (m= 3) | 10/30 | [−100,100] | 1700 |
| F46 | | Hybrid function 2 (m = 3) | 10/30 | [−100,100] | 1800 |
| F47 | | Hybrid function 3 (m = 4) | 10/30 | [−100,100] | 1900 |
| F48 | | Hybrid function 4 (m = 4) | 10/30 | [−100,100] | 2000 |
| F49 | | Hybrid function 5 (m= 5) | 10/30 | [−100,100] | 2100 |
| F50 | | Hybrid function 6 (m = 5) | 10/30 | [−100,100] | 2200 |
| F51 | Composition functions | Composition function 1 (m=5) | 10/30 | [−100,100] | 2300 |
| F52 | | Composition function 2 (m=3) | 10/30 | [−100,100] | 2400 |
| F53 | | Composition function 3 (m=3) | 10/30 | [−100,100] | 2500 |
| F54 | | Composition function 4 (m=5) | 10/30 | [−100,100] | 2600 |
| F55 | | Composition function 5 (m=5) | 10/30 | [−100,100] | 2700 |
| F56 | | Composition function 6 (m=5) | 10/30 | [−100,100] | 2800 |
| F57 | | Composition function 7 (m=3) | 10/30 | [−100,100] | 2900 |
| F58 | | Composition function 8 (m=3) | 10/30 | [−100,100] | 3000 |

**Table A.3**
The definition for CEC2017 test suite.

| No. | Type | Name | Dimension | Range | Optimum |
|---|---|---|---|---|---|
| F59 | Unimodal functions | Shifted and rotated bent cigar function | 10/30 | [−100,100] | 100 |
| F60 | | Shifted and rotated sum of different powers function | 10/30 | [−100,100] | 200 |
| F61 | | Shifted and rotated Zakharov function | 10/30 | [−100,100] | 300 |
| F62 | Basic multimodal functions | Shifted and rotated Rosenbrock's function | 10/30 | [−100,100] | 400 |
| F63 | | Shifted and rotated Rastrigin's function | 10/30 | [−100,100] | 500 |
| F64 | | Shifted and rotated expanded Scaffer's F6 function | 10/30 | [−100,100] | 600 |
| F65 | | Shifted and rotated Lunacek bi_rastrigin function | 10/30 | [−100,100] | 700 |
| F66 | | Shifted and rotated non-continuous Rastrigin's function | 10/30 | [−100,100] | 800 |
| F67 | | Shifted and rotated levy function | 10/30 | [−100,100] | 900 |
| F68 | | Shifted and rotated Schwefel's function | 10/30 | [−100,100] | 1000 |
| F69 | Hybrid functions | Hybrid function 1 (N = 3) | 10/30 | [−100,100] | 1100 |
| F70 | | Hybrid function 2 (N = 3) | 10/30 | [−100,100] | 1200 |
| F71 | | Hybrid function 3 (N = 3) | 10/30 | [−100,100] | 1300 |
| F72 | | Hybrid function 4 (N = 4) | 10/30 | [−100,100] | 1400 |
| F73 | | Hybrid function 5 (N = 4) | 10/30 | [−100,100] | 1500 |
| F74 | | Hybrid function 6 (N = 4) | 10/30 | [−100,100] | 1600 |
| F75 | | Hybrid function 7 (N = 5) | 10/30 | [−100,100] | 1700 |
| F76 | | Hybrid function 8 (N = 5) | 10/30 | [−100,100] | 1800 |
| F77 | | Hybrid function 9 (N = 5) | 10/30 | [−100,100] | 1900 |
| F78 | | Hybrid function 10 (N = 6) | 10/30 | [−100,100] | 2000 |
| F79 | Composition functions | Composition function 1 (N = 3) | 10/30 | [−100,100] | 2100 |
| F80 | | Composition function 2 (N = 3) | 10/30 | [−100,100] | 2200 |
| F81 | | Composition function 3 (N = 4) | 10/30 | [−100,100] | 2300 |
| F82 | | Composition function 4 (N = 4) | 10/30 | [−100,100] | 2400 |
| F83 | | Composition function 5 (N = 5) | 10/30 | [−100,100] | 2500 |
| F84 | | Composition function 6 (N = 5) | 10/30 | [−100,100] | 2600 |
| F85 | | Composition function 7 (N = 6) | 10/30 | [−100,100] | 2700 |
| F86 | | Composition function 8 (N = 6) | 10/30 | [−100,100] | 2800 |
| F87 | | Composition function 9 (N = 3) | 10/30 | [−100,100] | 2900 |
| F88 | | Composition function 10 (N = 3) | 10/30 | [−100,100] | 3000 |

## CRediT authorship contribution statement

**Yiying Zhang:** Conceptualization, Methodology, Software, Validation, Writing - original draft, Writing - review & editing.

## Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## Appendix

See Tables A.1–A.3.

## References

[1] T.Y. Tan, L. Zhang, S.C. Neoh, C.P. Lim, Intelligent skin cancer detection using enhanced particle swarm optimization, Knowl.-Based Syst. 158 (2018) 118–135, http://dx.doi.org/10.1016/j.knosys.2018.05.042.
[2] M.A. Elaziz, S. Xiong, K.P.N. Jayasena, L. Li, Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution, Knowl.-Based Syst. 169 (2019) 39–52, http://dx.doi.org/10.1016/j.knosys.2019.01.023.
[3] Y. Zhou, H. Wu, Q. Luo, M. Abdel-Baset, Automatic data clustering using nature-inspired symbiotic organism search algorithm, Knowl.-Based Syst. 163 (2019) 546–557, http://dx.doi.org/10.1016/j.knosys.2018.09.013.
[4] L. Shen, H. Chen, Z. Yu, W. Kang, B. Zhang, H. Li, B. Yang, D. Liu, Evolving support vector machines using fruit fly optimization for medical data classification, Knowl.-Based Syst. 96 (2016) 61–75, http://dx.doi.org/10.1016/j.knosys.2016.01.002.
[5] H. Chen, Q. Zhang, J. Luo, Y. Xu, X. Zhang, An enhanced Bacterial Foraging Optimization and its application for training kernel extreme learning machine, Appl. Soft Comput. 86 (2020) 105884, http://dx.doi.org/10.1016/j.asoc.2019.105884.
[6] Z. Feng, S. Liu, W. Niu, B. Li, W. Wang, B. Luo, S. Miao, A modified sine cosine algorithm for accurate global optimization of numerical functions and multiple hydropower reservoirs operation, Knowl.-Based Syst. 208 (2020) 106461, http://dx.doi.org/10.1016/j.knosys.2020.106461.
[7] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82, http://dx.doi.org/10.1109/4235.585893.
[8] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, Future Gener. Comput. Syst. 111 (2020) 300–323, http://dx.doi.org/10.1016/j.future.2020.03.055.
[9] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872, http://dx.doi.org/10.1016/j.future.2019.02.028.
[10] P. Civicioglu, Backtracking Search Optimization Algorithm for numerical optimization problems, Appl. Math. Comput. 219 (2013) 8121–8144, http://dx.doi.org/10.1016/j.amc.2013.02.017.
[11] A. Chatzipavlis, G.E. Tsekouras, V. Trygonis, A.F. Velegrakis, J. Tsimikas, A. Rigos, T. Hasiotis, C. Salmas, Modeling beach realignment using a neuro-fuzzy network optimized by a novel backtracking search algorithm, Neural Comput. Appl. 31 (2019) 1747–1763, http://dx.doi.org/10.1007/s00521-018-3809-2.
[12] K. Yu, J.J. Liang, B.Y. Qu, Z. Cheng, H. Wang, Multiple learning backtracking search algorithm for estimating parameters of photovoltaic models, Appl. Energy 226 (2018) 408–422, http://dx.doi.org/10.1016/j.apenergy.2018.06.010.
[13] A. Pourdaryaei, H. Mokhlis, H.A. Illias, S.H.A. Kaboli, S. Ahmad, Short-term electricity price forecasting via hybrid backtracking search algorithm and ANFIS approach, IEEE Access. 7 (2019) 77674–77691, http://dx.doi.org/10.1109/ACCESS.2019.2922420.
[14] J. Zhou, H. Ye, X. Ji, W. Deng, An improved backtracking search algorithm for casting heat treatment charge plan problem, J. Intell. Manuf. 30 (2019) 1335–1350, http://dx.doi.org/10.1007/s10845-017-1328-0.
[15] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, Opposition-based differential evolution, IEEE Trans. Evol. Comput. 12 (2008) 64–79, http://dx.doi.org/10.1109/TEVC.2007.894200.
[16] A.A. Ewees, M. Abd Elaziz, E.H. Houssein, Improved grasshopper optimization algorithm using opposition-based learning, Expert Syst. Appl. 112 (2018) 156–172, http://dx.doi.org/10.1016/j.eswa.2018.06.023.
[17] S. Shekhawat, A. Saxena, Development and applications of an intelligent crow search algorithm based on opposition based learning, ISA Trans. 99 (2020) 210–230, http://dx.doi.org/10.1016/j.isatra.2019.09.004.
[18] H. Chen, S. Jiao, A.A. Heidari, M. Wang, X. Chen, X. Zhao, An opposition-based sine cosine approach with local search for parameter estimation of photovoltaic models, Energy Convers. Manage. 195 (2019) 927–942, http://dx.doi.org/10.1016/j.enconman.2019.05.057.
[19] R.A. Ibrahim, M.A. Elaziz, S. Lu, Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization, Expert Syst. Appl. 108 (2018) 1–27, http://dx.doi.org/10.1016/j.eswa.2018.04.028.
[20] Y. Xu, Z. Yang, X. Li, H. Kang, X. Yang, Dynamic opposite learning enhanced teaching–learning-based optimization, Knowl.-Based Syst. 188 (2020) 104966, http://dx.doi.org/10.1016/j.knosys.2019.104966.
[21] Z. Xing, An improved emperor penguin optimization based multilevel thresholding for color image segmentation, Knowl.-Based Syst. 194 (2020) 105570, http://dx.doi.org/10.1016/j.knosys.2020.105570.
[22] S. Sapre, S. Mini, Opposition-based moth flame optimization with Cauchy mutation and evolutionary boundary constraint handling for global optimization, Soft Comput. 23 (2019) 6023–6041, http://dx.doi.org/10.1007/s00500-018-3586-y.
[23] W. Dong, L. Kang, W. Zhang, Opposition-based particle swarm optimization with adaptive mutation strategy, Soft Comput. 21 (2017) 5081–5090, http://dx.doi.org/10.1007/s00500-016-2102-5.
[24] J. Liang, B. Qu, P. Suganthan, A.G. Hernández-Díaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization, Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou China Nanyang Technol. Univ. Singap. Tech. Rep. 201212, 2013, pp. 281–295.
[25] J. Liang, B. Qu, P. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou China Tech. Rep. Nanyang Technol. Univ. Singap. 635, 2013.
[26] H. Zhang, R. Li, Z. Cai, Z. Gu, A.A. Heidari, M. Wang, H. Chen, M. Chen, Advanced orthogonal moth flame optimization with Broyden–Fletcher–Goldfarb–Shanno algorithm: Framework and real-world problems, Expert Syst. Appl. 159 (2020) 113617, http://dx.doi.org/10.1016/j.eswa.2020.113617.
[27] H. Zhang, Z. Cai, X. Ye, M. Wang, F. Kuang, H. Chen, C. Li, Y. Li, A multi-strategy enhanced salp swarm algorithm for global optimization, Eng. Comput. (2020) http://dx.doi.org/10.1007/s00366-020-01099-4.
[28] Y. Xu, H. Chen, J. Luo, Q. Zhang, S. Jiao, X. Zhang, Enhanced Moth-flame optimizer with mutation strategy for global optimization, Inform. Sci. 492 (2019) 181–203, http://dx.doi.org/10.1016/j.ins.2019.04.022.
[29] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18, http://dx.doi.org/10.1016/j.swevo.2011.02.002.
[30] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Mixed variable structural optimization using Firefly algorithm, Comput. Struct. 89 (2011) 2325–2336, http://dx.doi.org/10.1016/j.compstruc.2011.08.002.
[31] S. Gupta, R. Tiwari, S.B. Nair, Multi-objective design optimisation of rolling bearings using genetic algorithms, Mech. Mach. Theory 42 (2007) 1418–1443, http://dx.doi.org/10.1016/j.mechmachtheory.2006.10.002.
[32] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (2011) 303–315, http://dx.doi.org/10.1016/j.cad.2010.12.015.
[33] G. Dhiman, V. Kumar, Emperor penguin optimizer: A bio-inspired algorithm for engineering problems, Knowl.-Based Syst. 159 (2018) 20–50, http://dx.doi.org/10.1016/j.knosys.2018.06.001.
[34] G. Dhiman, V. Kumar, Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, Knowl.-Based Syst. 165 (2019) 169–196, http://dx.doi.org/10.1016/j.knosys.2018.11.024.
[35] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110–111 (2012) 151–166, http://dx.doi.org/10.1016/j.compstruc.2012.07.010.
[36] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (2013) 2592–2612, http://dx.doi.org/10.1016/j.asoc.2012.11.026.
[37] G. Dhiman, V. Kumar, Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications, Adv. Eng. Softw. 114 (2017) 48–70, http://dx.doi.org/10.1016/j.advengsoft.2017.05.014.
[38] B. Kannan, S.N. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, J. Mech. Des. 116 (1994) 405–411.

[39] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249, http://dx.doi.org/10.1016/j.knosys.2015.07.006.
[40] S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133, http://dx.doi.org/10.1016/j.knosys.2015.12.022.
[41] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61, http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.
[42] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-Verse Optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2016) 495–513, http://dx.doi.org/10.1007/s00521-015-1870-7.
[43] H. Garg, A hybrid GSA-GA algorithm for constrained optimization problems, Inform. Sci. 478 (2019) 499–523, http://dx.doi.org/10.1016/j.ins.2018.11.041.
[44] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, Comput. Struct. 169 (2016) 1–12, http://dx.doi.org/10.1016/j.compstruc.2016.03.001.
[45] S.A. Hoseini Shekarabi, A. Gharaei, M. Karimi, Modelling and optimal lot-sizing of integrated multi-level multi-wholesaler supply chains under the shortage and limited warehouse space: generalised outer approximation, Int. J. Syst. Sci. Oper. Logist. 6 (2019) 237–257, http://dx.doi.org/10.1080/23302674.2018.1435835.
[46] A. Gharaei, M. Karimi, S.A. Hoseini Shekarabi, An integrated multi-product, multi-buyer supply chain under penalty, green, and quality control polices and a vendor managed inventory with consignment stock agreement: The outer approximation with equality relaxation and augmented penalty algorithm, Appl. Math. Model. 69 (2019) 223–254, http://dx.doi.org/10.1016/j.apm.2018.11.035.
[47] C. Duan, C. Deng, A. Gharaei, J. Wu, B. Wang, Selective maintenance scheduling under stochastic maintenance quality with multiple maintenance actions, Int. J. Prod. Res. 56 (2018) 7160–7178, http://dx.doi.org/10.1080/00207543.2018.1436789.
[48] M. Rabbani, S.A.A. Hosseini-Mokhallesun, A.H. Ordibazar, H. Farrokhi-Asl, A hybrid robust possibilistic approach for a sustainable supply chain location–allocation network design, Int. J. Syst. Sci. Oper. Logist. 7 (2020) 60–75, http://dx.doi.org/10.1080/23302674.2018.1506061.
[49] N. Kazemi, S.H. Abdul-Rashid, R.A.R. Ghazilla, E. Shekarian, S. Zanoni, Economic order quantity models for items with imperfect quality and emission considerations, Int. J. Syst. Sci. Oper. Logist. 5 (2018) 99–115, http://dx.doi.org/10.1080/23302674.2016.1240254.
[50] S. Sarkar, B.C. Giri, Stochastic supply chain model with imperfect production and controllable defective rate, Int. J. Syst. Sci. Oper. Logist. 7 (2020) 133–146, http://dx.doi.org/10.1080/23302674.2018.1536231.
[51] B.C. Giri, S. Bardhan, Coordinating a supply chain with backup supplier through buyback contract under supply disruption and uncertain demand, Int. J. Syst. Sci. Oper. Logist. 1 (2014) 193–204, http://dx.doi.org/10.1080/23302674.2014.951714.
[52] A. Gharaei, M. Karimi, S.A. Hoseini Shekarabi, Joint economic lot-sizing in multi-product multi-level integrated supply chains: Generalized benders decomposition, Int. J. Syst. Sci. Oper. Logist. (2019) 1–17, http://dx.doi.org/10.1080/23302674.2019.1585595.
[53] S. Yin, T. Nishi, G. Zhang, A game theoretic model for coordination of single manufacturer and multiple suppliers with quality variations under uncertain demands, Int. J. Syst. Sci. Oper. Logist. 3 (2016) 79–91, http://dx.doi.org/10.1080/23302674.2015.1050079.
[54] N.H. Shah, U. Chaudhari, L.E. Cárdenas-Barrón, Integrating credit and replenishment policies for deteriorating items under quadratic demand in a three echelon supply chain, Int. J. Syst. Sci. Oper. Logist. 7 (2020) 34–45, http://dx.doi.org/10.1080/23302674.2018.1487606.
[55] A. Gharaei, S.A. Hoseini Shekarabi, M. Karimi, E. Pourjavad, A. Amjadian, An integrated stochastic EPQ model under quality and green policies: generalised cross decomposition under the separability approach, Int. J. Syst. Sci. Oper. Logist. (2019) 1–13, http://dx.doi.org/10.1080/23302674.2019.1656296.
[56] B.C. Giri, M. Masanta, Developing a closed-loop supply chain model with price and quality dependent demand and learning in production in a stochastic environment, Int. J. Syst. Sci. Oper. Logist. 7 (2020) 147–163, http://dx.doi.org/10.1080/23302674.2018.1542042.
[57] A. Gharaei, S.A. Hoseini Shekarabi, M. Karimi, Modelling and optimal lot-sizing of the replenishments in constrained, multi-product and bi-objective EPQ models with defective products: Generalised Cross Decomposition, Int. J. Syst. Sci. Oper. Logist. 7 (2020) 262–274, http://dx.doi.org/10.1080/23302674.2019.1574364.