

RESEARCH ARTICLE

Self-adaptive dual-strategy differential evolution algorithm

Meijun Duan^{1*}, Hongyu Yang^{1,2}, Shangping Wang³, Yu Liu²

1 National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University, Chengdu, China, **2** College of Computer Science, Sichuan University, Chengdu, China, **3** Science and Technology on Electronic Information Control Laboratory, Chengdu, China

* dmj23@163.com

Abstract

Exploration and exploitation are contradictory in differential evolution (DE) algorithm. In order to balance the search behavior between exploitation and exploration better, a novel self-adaptive dual-strategy differential evolution algorithm (SaDSDE) is proposed. Firstly, a dual-strategy mutation operator is presented based on the “DE/best/2” mutation operator with better global exploration ability and “DE/rand/2” mutation operator with stronger local exploitation ability. Secondly, the scaling factor self-adaption strategy is proposed in an individual-dependent and fitness-dependent way without extra parameters. Thirdly, the exploration ability control factor is introduced to adjust the global exploration ability dynamically in the evolution process. In order to verify and analyze the performance of SaDSDE, we compare SaDSDE with 7 state-of-art DE variants and 3 non-DE based algorithms by using 30 Benchmark test functions of 30-dimensions and 100-dimensions, respectively. The experiments results demonstrate that SaDSDE could improve global optimization performance remarkably. Moreover, the performance superiority of SaDSDE becomes more significant with the increase of the problems' dimension.



OPEN ACCESS

Citation: Duan M, Yang H, Wang S, Liu Y (2019) Self-adaptive dual-strategy differential evolution algorithm. PLoS ONE 14(10): e0222706. <https://doi.org/10.1371/journal.pone.0222706>

Editor: Mohd Nadhir Ab Wahab, Universiti Sains Malaysia, MALAYSIA

Received: June 3, 2019

Accepted: September 4, 2019

Published: October 3, 2019

Copyright: © 2019 Duan et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the manuscript.

Funding: This work is supported by the NSFC and CAAC Joint Fund Project (NO. U1833115). The funder had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Introduction

DE is a stochastic optimization method based on population for real-parameter optimization problems, which is proposed by Storn and Price [1]. As a kind of heuristic global search evolution algorithm, differential evolution (DE) has an evolution mechanism similar to the other evolution algorithms, all of which are mutated, crossed and selected. Unlike that the other evolution algorithms (EAs) use the defined probability distribution function, the DE selects individuals from the current population to do differential operations and multiplies by scaling factor. It currently is a very attractive evolutionary algorithm for optimization in continuous search spaces, mainly for its simplicity, a small number of parameters to tune and notable performance. At present, DE is widely used in the research and engineering fields, such as classification [2], neural network training [3–4], data clustering [5], solar energy [6].

Exploration and exploitation are contradictory for most EAs [7]. To further improve the exploitation and exploration capability of DE algorithm, a novel self-adaptive dual-strategy

differential evolution algorithm (SaDSDE) is proposed. In summary, the main contributions of this paper are as follows. (1) A dual-strategy mutation operator is proposed. In the proposed mutation operator, the exploitation and exploration capability of mutation operator are simultaneously considered. (2) The scaling factor self-adaption strategy is proposed in an individual-dependent and fitness-dependent way. (3) A dynamic adjustment strategy of exploration ability control factor is introduced to adjust the exploration ability of DE in different stages of evolution process. (4) The performance of SaDSDE is verified by comparing it with 7 existing state-of-art DE variants and 3 non-DE algorithms on 30 benchmark functions. The experimental results demonstrate the effectiveness of the proposed SaDSDE. The performance sensitivities to population size, crossover probability and the impacts of the proposed algorithmic components are also investigated.

The rest of this paper is organized as follows. Section 2 briefly introduces the original DE algorithm and reviews the related works on the DE variants. In Section 3, the proposed SaDSDE algorithm is presented in detail. The experiment comparisons and analysis with 7 state-of-art algorithms and 3 non-DE algorithms are given in Section 4. Finally, Section 5 draws the conclusions of this work, and future work is also described in this section.

Related work

2.1 Original differential evolution algorithm

DE is based on population evolution. Generally, DE is composed of four components: initialization, mutation, crossover and selection. After initialization, DE enters a loop of mutation, crossover and selection. The details of main components are introduced as follows. Without loss of generality, this paper considers minimization problems. For D -dimensional minimum optimization problems:

$$\begin{cases} \min f(x_1, x_2, \dots, x_D) \\ s.t. x_j^{\min} \leq x_{ij} \leq x_j^{\max}, \quad j = 1, 2, \dots, D \end{cases} \quad (1)$$

where $i = \{1, 2, \dots, N_p\}$, N_p is the population size, x_j^{\min} and x_j^{\max} are respectively the lower bound and upper bound of x_j .

2.1.1 Mutation. It is mainly to generate mutation vector by scaling the difference of different individuals. There are many mutation operators proposed by Storn and Price [8, 9] and a widely used mutation strategy is listed as follows:

$$V_i^t = X_{r_1}^t + F \cdot (X_{r_2}^t - X_{r_3}^t) \quad (2)$$

where t indicates the generation, the indexes $r_1 \sim r_3$ are distinct integers randomly chosen from the set $\{1, 2, \dots, N_p\} \setminus \{i\}$, V_i^t is the mutation vector of the i th target vector X_i^t . The scaling factor F controls the amplification of the difference vector and is closely related to convergence speed. The small scaling factor determines the exploitation ability. On the contrary, the large scaling factor determines the exploration ability. It is note that the optimization performance of DE mainly depends on the choice of mutation strategies and the setting of control parameters.

2.1.2 Crossover. By crossover operation on the mutation vector V_i^t and the i th target vector X_i^t , the trial vector is generated. The diversity of the population can be increased by crossover operation. There are mainly three classic crossover operators: binomial crossover, exponential crossover and rotationally invariant arithmetic crossover operators. The following

binomial crossover operator is the most commonly used.

$$U_{ij}^t = \begin{cases} V_{ij}^t & \text{if } rand_j \leq CR \text{ or } j = j_{rand} \\ X_{ij}^t & \text{otherwise} \end{cases} \quad (3)$$

where $rand_j$ is a uniformly distributed random variable within $(0, 1)$. j_{rand} is randomly chosen from the set $\{1, 2, \dots, D\}$, which can prevent a direct copy from X_i^t to U_i^t . The crossover probability CR is a user-defined crossover factor restricted to $(0, 1]$, which controls the diversity of the population and is closely connected with exploration power.

2.1.3 Selection. After crossover, the objective function value of the trial vector can be obtained according to the optimization problem. The newly generated trial vectors are evaluated and compared with the target vectors. A greedy strategy is adopted to perform the selection operation in DE. The superior of the target vector X_i^t and trial vector U_i^t will survive in the next generation. In mathematics, one has

$$X_i^{t+1} = \begin{cases} U_i^t & \text{if } f(U_i^t) \leq f(X_i^t) \\ X_i^t & \text{others} \end{cases} \quad (4)$$

2.2 Literature review. The performance of DE is mainly influenced by mutation mode, control parameters (i.e. population size (N_p), scaling factor (F) and crossover probability (CR)) and crossover mode. Inappropriate configurations of mutation strategies and control parameters can cause stagnation or premature convergence. Therefore, many scholars have proposed a series of improved DE algorithms [10–46].

To avoid manually tuning parameters, researchers have developed some techniques to automatically set the parameter values. Ryoji and Alex [10] used a historical memory of successful control parameters to guide the selection of future control parameters and proposed a parameter adaptation technique for DE (SHADE). Rcr-JADE [11] was an improved version of JADE [12] which employs successful parameters to repair crossover rate. To enhance the performance of L-SHADE algorithm, Awad and Ali et al. [13] proposed LSHADE-EpSin by using an adaptive approach based on sinusoidal formulas to adapt the scaling factor. In addition, some scholars have proposed adaptive strategies for the population size. Zhu et al. [14] proposed an adaptive population tuning scheme. In SapsDE [15], a self-adaptive population resizing mechanism was employed to adjust the population size. Chen and Zhao [16] et al. proposed population adaptive differential evolution (PADE). Award and Ali [17] et al. presented ensemble sinusoidal differential evolution with niching-based population reduction (called EsDEr-NR).

A large number of studies have carried out on improving mutation strategies. A part of studies focused on single-mutation mode. In JADE [12], an optional external archive is combined with a mutation strategy DE/current-to-pbest/1 that utilizes historical information to direct population searching. Wang et al. [18] proposed a self-adaptive differential evolution algorithm with improved mutation mode (IMMSADE) by improving “DE/rand/1”. Cai et al. [19] designed a neighborhood-dependent directional mutation operator and presented a neighborhood-adaptive DE (NaDE). A novel “DE/current-to-SP-best-ring/1” mutation operation is introduced in decentralizing and coevolving differential evolution (DCDE), proposed by Tang [20]. A novel and effective adaptation scheme is used to update the crossover rate in adaptive guided differential evolution algorithm (AGDE) [21]. He and Zhou [22] presented a novel DE variant with covariance matrix self-adaptation (DECM-SA). In EFADE [23], a new triangular mutation operator is introduced. Cai et al. [24] presented an adaptive social learning (ASL) strategy to extract the neighborhood relationship information. The best search strategies

and parameters of an EA are generally different in solving different optimization problems. Therefore, in order to improve the performance of an EA like DE, researchers make efforts to realize an ensemble of multiple strategies and parameters [25–33]. Qin et al. [25] proposed a self-adaptive DE algorithm (SaDE), both trial vector generation strategies and their associated control parameter values were gradually self-adapted. Wang et al. [26] introduced a Composite Differential Evolution algorithm (CoDE), which used three trial vector generation strategies and three control parameter settings. Mallipeddi et al. [27] employed an ensemble of mutation strategies and control parameters with the DE (EPSDE). Elsayed et al. [28] used multiple search operators in conjunction with multiple constraint handling techniques. Wu and Mallipeddi et al. [29] proposed a multi-population ensemble DE (MPEDE). YEH et al. [30] mixed Gauss mutation and the “DE/best/1” operator. Cui et al. [31] proposed an adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism (AME-CoDEs). Wu et al. [32] focused on the high-level ensemble of different DE variants and proposed a new algorithm named EDEV. Lin and Ma et al. [33] proposed an adaptive immune-inspired multi-objective algorithm (AIMA).

In addition to the modification of mutation and control parameters optimization, enhancements in crossover operators have also been investigated, such as covariance matrix learning operator [34], hybrid linkage crossover [35], a crossover operator utilizing eigenvectors of covariance matrix of individual solutions [36], superior-inferior crossover scheme [37], an adaptive hybrid crossover operator (AHX) [38], optional blending crossover scheme [39] and a multiple exponential recombination that inherits all the main advantages of existing crossover operators [40].

Through the synergistic mechanism, a hybrid algorithm could take advantage of various merits within different algorithms, and then yields more favorable performance than a single algorithm. Some preliminary research manifest that hybrid optimizers are effective and competent for global optimization. Li et al. [41] proposed a new hybrid algorithm, called as differential evolution algorithm (DE) / artificial bee colony (ABC) algorithm. Vaisakh et al. [42] came up with a hybrid approach involving differential evolution (DE) and bacterial foraging optimization algorithm (BFOA). Ponsich and Coello [43] hybridized DE with Tabu Search (TS). Gu et al. [44] mixed binary differential evolution (BDE) and Tabu search (TS) to propose the memetic algorithm. Le et al. [45] merged differential evolution and harmony search. Nenuvath and Jatoth [46] hybridized sine cosine algorithm with differential evolution.

SaDSDE algorithm

3.1 Dual-strategy mutation operator

In the original mutation strategies [8, 9], “DE/best/1” and “DE/best/2” conduct mutation on the best individual with better local exploitation ability and fast convergence speed, but they are easy to suffer premature convergence. “DE/rand/1” and “DE/rand/2” do mutation based on random individuals with stronger global exploration ability, but they are lack of the guidance of the optimal individual, so the convergence speed is slow. “DE/current-to-best/1” performs mutation based on the parent individual and the optimal individual with high convergence precision, but it is easy to fall into local optimum. Above mutation strategies are either too greedy or too stochastic. Therefore, in order to balance the exploitation and exploration better, a novel dual-strategy mutation operator is proposed based on “DE/best/2” and “DE/rand/2”, which is shown in Eqs (5)–(7).

$$V_{i-1}^t = X_b^t + F_i^t \cdot (X_{r_1}^t - X_{r_2}^t) + F_i^t \cdot (X_{r_3}^t - X_{r_4}^t) \quad (5)$$

$$V_{i-2}^t = X_{r_1}^t + F_i^t \cdot (X_{r_2}^t - X_{r_3}^t) + F_i^t \cdot (X_{r_4}^t - X_{r_5}^t) \tag{6}$$

$$V_i^t = \omega_{i-1} \cdot V_{i-1}^t + \lambda \cdot \omega_{i-2} \cdot V_{i-2}^t \tag{7}$$

where X_b^t is the best solution at the current generation t . $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5$ are randomly chosen from the set $\{1, \dots, N_p\} \setminus \{i\}$, F_i^t is the scaling factor of the i th individual. ω_{i-1} and ω_{i-2} are the weights of two mutation operators, which take the value of either 0 or 1. If $rand_i() < 0.5$, $\omega_{i-1} = 1, \omega_{i-2} = 0$; otherwise, $\omega_{i-1} = 0, \omega_{i-2} = 1$. Two weights satisfy $\omega_{i-1} + \omega_{i-2} = 1$. In the above mutation scheme, only one of the two strategies is used for each vector depending on a uniformly distributed random value within the range (0, 1). λ is the exploration ability control factor and adjusts the exploration ability of mutation operator.

3.2 The scaling factor self-adaption strategy

The solution quality and the fitness are closely related. For the solution quality, the fitness is the larger the better. Therefore, the scaling factor is tuned according to the fitness in a self-adaptive and individual-dependent way, as is described in Eq (8).

$$F_i^t = (f_{max}^t - f(X_i^t)) / (f_{max}^t - f_{min}^t) \tag{8}$$

Where $f(X_i^t)$ is the fitness of the individual X_i^t , f_{max}^t is the maximum fitness at the current generation, and f_{min}^t is the minimum fitness. In particular, when f_{max}^t is equal to f_{min}^t , F_i^t is set to a uniformly distributed random variable within (0, 1) to increase disturbance. Superior individuals tend to be assigned with smaller parameter values so as to exploit their neighborhoods in which better solutions, while inferior individuals tend to be assigned with larger parameter values so as to explore further areas in the solution space.

3.3 The dynamic adjustment strategy of the exploration ability control factor λ

A key problem in many evolutionary algorithms is premature convergence, especially in the later stage of evolution. However, promoting strong global exploration ability at all stages of an evolutionary process might even be counterproductive in a phase where high exploitation is needed. In general, most algorithms perform more effectively with a high convergence rate in the earlier stage of the searching process relative to the later stage, especially for multimodal functions. At the later stage, the algorithms are easily trapped in local optimal solutions due to poor population diversity. Therefore, the dynamic adjustment strategy of the exploration ability control factor is introduced according to the evolution property, which is shown in Eq (9).

$$\lambda = 1 - \cos((t/T)^2) \tag{9}$$

Taking 1000 generations as an example, the dynamic adjustment curve of the exploration ability control factor is shown in the Fig 1. During evolution, the population diversity is better and λ is assigned a smaller value at the earlier stage. In response to the decreased diversity of the population at the later stage of evolution, λ is assigned a larger value to increase the proportions of the explorative mutation operator.

Based on all the components described above, the pseudo-code of the complete procedure of SaDSDE is outlined in Algorithm 1. Comparing to the basic DEs, the time complexity of proposed SaDSDE is still $O(T * N_p * D)$ without extra time complexity. Herein, T, N_p and D are respectively the maximum generation number, the population size and the dimension of problem.

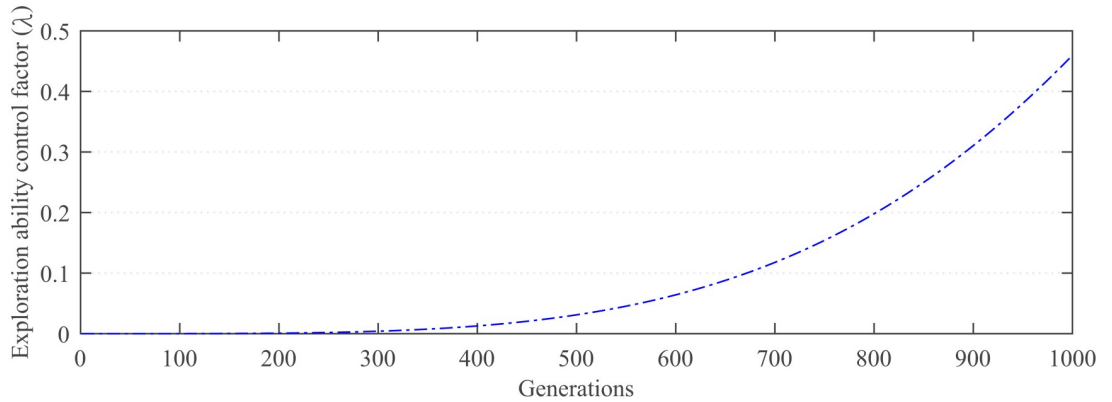


Fig 1. The dynamic adjustment curve of λ .

<https://doi.org/10.1371/journal.pone.0222706.g001>

Algorithm 1. Pseudo-Code for the SaDSDE

```

Begin
#01  $t = 0$ ;
#02 Randomly initialize an initial population
 $X^{(0)} = \{X_{ij}^{(0)} | \forall i, i = 1, \dots, N_p, \forall j, j = 1, \dots, D\}$ ,  $X_{ij}^{(0)} \in [x_j^{\min}, x_j^{\max}]$ ,  $T = 1000$ ,  $CR = 0.9$ ;
#03 For  $t = 1$  to  $T$  Do
#04 Calculate the exploration ability control factor  $\lambda = 1 - \cos((t/T)^2)$ ;
#05 Evaluate  $f(X_i^t | \forall i, i = 1, \dots, N_p)$  and find  $X_b^t$ ;
#06 For each individual  $X_i^t$  in current population Do
#07 If  $f_{\max}^t \neq f_{\min}^t$  Then
#08 Update  $F_i^t = (f_{\max}^t - f(X_i^t)) / (f_{\max}^t - f_{\min}^t)$ ;
#09 Else
#10 Set  $F_i^t = rand(0, 1)$  to increase disturbance;
#11 End
#12 Generate a random real number  $r_i^t = rand(0, 1)$ ;
#13 If  $r_i^t < 0.5$  Then
#14  $\omega_{i-1} = 1$  and  $\omega_{i-2} = 0$ ;
#15 Else
#16  $\omega_{i-1} = 0$  and  $\omega_{i-2} = 1$ ;
#17 End
#18 Generate  $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5$  from the set  $\{1, \dots, N_p\} \setminus \{i\}$ 
randomly;
#19  $V_{i-1}^t = X_b^t + F_i^t \cdot (X_{r_1}^t - X_{r_2}^t) + F_i^t \cdot (X_{r_3}^t - X_{r_4}^t)$ ;
#20  $V_{i-2}^t = X_{r_1}^t + F_i^t \cdot (X_{r_2}^t - X_{r_3}^t) + F_i^t \cdot (X_{r_4}^t - X_{r_5}^t)$ ;
#21 For  $j = 1$  to  $D$  Do
#22 Generate  $j_{rand} = randint(1, D)$ ;
#23 If  $j = j_{rand}$  or  $rand(0, 1) < CR$  Then
#24  $U_{ij}^t = \omega_{i-1} \cdot V_{i-1j}^t + \lambda \cdot \omega_{i-2} \cdot V_{i-2j}^t$ ;
#25 Else
#26  $U_{ij}^t = X_{ij}^t$ ;
#27 End If
#28 End For
#29 If  $f(U_i^t) \leq f(X_i^t)$  Then
#30  $X_i^{t+1} = U_i^t$ ;
#31 Else
#32  $X_i^{t+1} = X_i^t$ ;
#33 End If
#34 End For

```

```
#35 t = t + 1
#36 End While
End
```

Experiments analysis

4.1 Experiments setup

In our experiments, 30 benchmark test functions from references [47–49] are used to test the performance of SaDSDE, which are listed in Table 1. f_1 – f_{11} are unimodal functions, f_{12} – f_{30} are multimodal functions.

4.2 Time complexity

The time complexity is calculated as described in [47]. The codes are implemented in Matlab 2015a and run on a PC with an Intel (R) Core (TM) i5-6500 CPU (3.20GHz) and 8GB RAM. The algorithm complexity is listed in Table 2. In Table 2, T_0 denotes the running time of the following program:

```
for i = 1 : 1000000
x = (double)5.55;
x = x + x; x = x./2; x = x * x; x = sqrt(x); x = ln(x); x = exp(x); y = x/x;
end
```

T_1 is the computing time just for Function 3 (Elliptic function) for 200,000 evaluations at a certain dimension D . T_2 is the complete computing time for the algorithm with 200,000 evaluations of D dimensional Elliptic Function. T_2 is evaluated five times, and \hat{T}_2 is used to denote the mean T_2 . At last, the complexity of the algorithm is reflected by: T_0 , T_1 , \hat{T}_2 , $(\hat{T}_2 - T_1)/T_0$.

4.3 Sensitivity analysis to control parameters

4.3.1 Sensitivity analysis to population size. The effect of population size in SaDSDE is investigated by testing 30-dimensional problems. N_p is set to 50, 75, 100, 125, 150, 175, 200, 225 and 250. For each test function with a different population size, the maximum generation number is still set to 1000. From the statistical Friedman test [50] results in Fig 2, the performance of SaDSDE decreases with reducing population size, because a larger population size has high exploration ability.

Although SaDSDE with $N_p = 250$ is best in Fig 2, there is no obvious performance difference in Table 3. Therefore, we can say that SaDSDE is not sensitive to the population size.

4.3.2 Sensitivity analysis to crossover probability. The impact of the crossover probability CR on the performance of proposed algorithm is also analyzed. The candidate values of CR are 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9, respectively. We perform Friedman test and Wilcoxon's rank-sum test [50] among the optimization result on different CR values, respectively. The test results are shown in Fig 3 and Table 4, respectively. From Fig 3, we can observe that the performance of SaDSDE is best at $CR = 0.9$.

From Table 4, it can be observed that SaDSDE is not sensitive to CR except for $CR = \{0.1, 0.2, 0.3\}$ at the 0.1 significance level. Based on the trade-off of the convergence precision and convergence rate, we think that $CR = 0.9$ is a more appropriate value. Therefore, we set $CR = 0.9$ in the following series of experiments unless noted otherwise.

Table 1. Benchmark test functions.

Name	Function	Domain
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	$[-100,100]^D$
Schwefel 1.2	$f_2(x) = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	$[-100,100]^D$
Elliptic	$f_3(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2$	$[-100,100]^D$
Schwefel 2.22	$f_4(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10,10]^D$
Schwefel 2.21	$f_5(x) = \max\{ x_i , 1 \leq i \leq D\}$	$[-100,100]^D$
SumSquares	$f_6(x) = \sum_{i=1}^D ix_i^2$	$[-1,1]^D$
Tablet	$f_7(x) = 10^6 x_1^2 + \sum_{i=2}^D x_i^2$	$[-100,100]^D$
Zakharov	$f_8(x) = \sum_{i=1}^D x_i^2 + (\sum_{i=1}^D 0.5ix_i)^2 + (\sum_{i=1}^D 0.5ix_i)^4$	$[-5,10]^D$
Bent Cigar	$f_9(x) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2$	$[-100,100]^D$
Step	$f_{10}(x) = \sum_{i=1}^D (x_i + 0.5)^2$	$[-100,100]^D$
Noise Quartic	$f_{11}(x) = \sum_{i=1}^D ix_i^4 + \text{rand}[0, 1]$	$[-1.28,1.28]^D$
Rastrigin	$f_{12}(x) = \sum_{i=1}^D (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12,5.12]^D$
Griewank	$f_{13}(x) = \sum_{i=1}^D x_i^2/4000 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$	$[-600,600]^D$
Scaffer's F6	$f_{14}(x) = \sum_{i=1}^D (0.5 + \frac{\sin^2(\sqrt{x_i^2 + x_{i+1}^2}) - 0.5}{(1 + 0.001(x_i^2 + x_{i+1}^2))^2})x_{D+1} = x_1$	$[-0.5,0.5]^D$
Salomon	$f_{15}(x) = 1 - \cos(2\pi\sqrt{\sum_{i=1}^D x_i^2}) + 0.1\sqrt{\sum_{i=1}^D x_i^2}$	$[-100,100]^D$
Ackley	$f_{16}(x) = 20 + e - 20\exp(-0.2\sqrt{\sum_{i=1}^D x_i^2/D}) - \exp(\sum_{i=1}^D \cos(2\pi x_i)/D)$	$[-32,32]^D$
Rosenbrock	$f_{17} = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	$[-100,100]^D$
Scaffer2	$f_{18}(x) = \sum_{i=1}^D (x_i^2 + x_{i+1}^2)^{0.25} (\sin(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1)$	$[-100,100]^D$
Modified Schwefel	$f_{19}(x) = 418.9829 * D - \sum_{i=1}^D g(z_i), z_i = x_i + 4.20968746227503e + 002$ $g(z_i) = \begin{cases} z_i \sin(z_i ^{1/2}) & \text{if } z_i < 500 \\ (500 - \text{mod}(z_i, 500))\sin(\sqrt{ 500 - \text{mod}(z_i , 500) }) - \frac{(z_i - 500)^2}{1000D} & \text{if } z_i > 500 \\ (\text{mod}(z_i, 500) - 500)\sin(\sqrt{ \text{mod}(z_i , 500) - 500 }) - \frac{(z_i - 500)^2}{1000D} & \text{if } z_i < -500 \end{cases}$	$[-100,100]^D$
HappyCat	$f_{20}(x) = \sum_{i=1}^D x_i^2 - D ^{1/4} + (0.5\sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i)/D + 0.5$	$[-100,100]^D$
HGBat	$f_{21}(x) = (\sum_{i=1}^D x_i^2)^2 - (\sum_{i=1}^D x_i)^2 ^{1/2} + (0.5\sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i)/D + 0.5$	$[-100,100]^D$
Weierstrass	$f_{22}(x) = \sum_{i=1}^D (\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k(x_i + 0.5))]) - D\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k \cdot 0.5)], a = 0.5, b = 3, k_{\max} = 20$	$[-100,100]^D$
Katsuura	$f_{23}(x) = \frac{10}{D^2} \prod_{i=1}^D (1 + i \sum_{j=1}^{32} \frac{ 2^j x_i - \text{round}(2^j x_i) }{2^j})^{\frac{10}{D^2}} - \frac{10}{D^2}$	$[-5,5]^D$
E_ScafferF6	$f_{24}(x) = f_{14}(x_1, x_2) + \dots + f_{14}(x_{D-1}, x_D) + f_{14}(x_D, x_1)$	$[-3,1]^D$
Griewank+Rosenbrock	$f_{25}(x) = f_{13}(f_{17}(x_1, x_2)) + \dots + f_{13}(f_{17}(x_{D-1}, x_D)) + f_{13}(f_{17}(x_D, x_1))$	$[-5.12,5.12]^D$
NCRastrigin	$f_{26} = \sum_{i=1}^D [y_i^2 - 10\cos(2\pi y_i + 10)], y_i = \begin{cases} x_i, x_i < 0.5 \\ \text{round}(2x_i)/2, x_i \geq 0.5 \end{cases}$	$[-10,10]^D$
Apline	$f_{27} = \sum_{i=1}^D x_i \sin x_i + 0.1x_i $	$[-100,100]^D$
Bohachevsky_2	$f_{28} = \sum_{i=1}^{D-1} [x_i^2 + 2x_{i+1}^2 - 0.3\cos(3\pi x_i)\cos(3\pi x_{i+1}) + 0.3]$	$[-100,100]^D$
Levy and Montalvo 1	$f_{29}(x) = \frac{\pi}{D} (10\sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 (1 + 10\sin^2(\pi y_{i+1})) + (y_D - 1)^2), y = 1 + \frac{1}{4}(x_i + 1)$	$[-10, 10]^D$
Levy and Montalvo 2	$f_{30}(x) = 0.1(10\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) + (x_D - 1)^2 (1 + \sin^2(2\pi x_D)))$	$[-5, 5]^D$

<https://doi.org/10.1371/journal.pone.0222706.t001>

Table 2. Time complexity (time in seconds).

Dimensions	T_0	T_1	\hat{T}_2	$(\hat{T}_2 - T_1)/T_0$
$D = 30$	0.1160	0.1030	3.6340	30.4394
$D = 100$		0.1170	3.6175	30.1767

<https://doi.org/10.1371/journal.pone.0222706.t002>

4.4 Parameter settings and involved algorithms

For rigorous performance verification, SaDSDE is compared with the following 7 well-established DE variants (i.e., Rcr-JADE [11], EsDer-NR [17], IMMSADE [18], AGDE [21], EFADE [23], MPEDE [29] and EDEV [32]) and 3 non-DE algorithms (i.e., social learning particle swarm optimization (SL-PSO) [51], grey wolf optimization (GWO) [52] and whale optimization algorithm (WOA) [53]). Most parameters of compared algorithms are kept the same as used in their original literatures, which are shown in Table 5.

The common parameters are set as follows. The max number of iterations is set to $T = 1000$, the population size is set to $N_p = 100$, and the times of all experiment runs is set to 30.

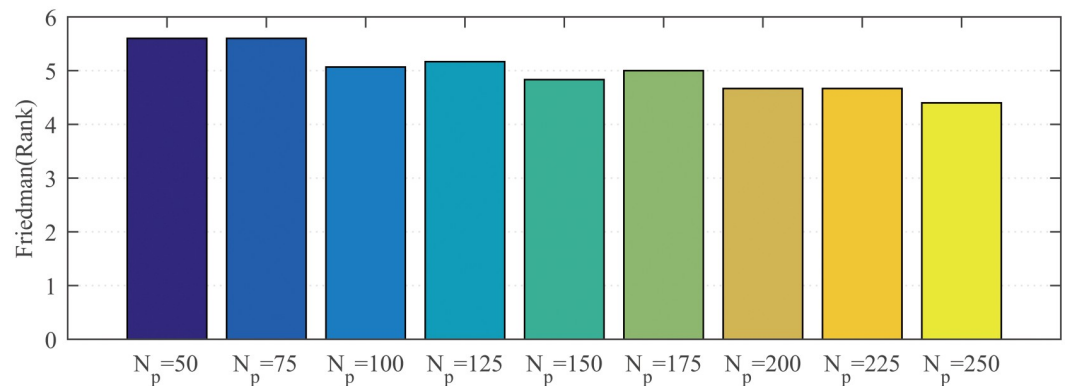


Fig 2. Friedman test results at $D = 30$ over 30 independent runs.

<https://doi.org/10.1371/journal.pone.0222706.g002>

Table 3. The results of Wilcoxon’s rank-sum test over 30 independent runs.

Comparisons	R+	R-	p-value	$\alpha = 0.05$	$\alpha = 0.1$
SaDSDE with $N_p = 250$ versus SaDSDE with $N_p = 50$	21	0	9.17E-01	No	No
SaDSDE with $N_p = 250$ versus SaDSDE with $N_p = 75$	16	5	9.32E-01	No	No
SaDSDE with $N_p = 250$ versus SaDSDE with $N_p = 100$	21	0	9.62E-01	No	No
SaDSDE with $N_p = 250$ versus SaDSDE with $N_p = 125$	15	6	9.62E-01	No	No
SaDSDE with $N_p = 250$ versus SaDSDE with $N_p = 150$	18	3	9.77E-01	No	No
SaDSDE with $N_p = 250$ versus SaDSDE with $N_p = 175$	15	6	9.77E-01	No	No
SaDSDE with $N_p = 250$ versus SaDSDE with $N_p = 200$	21	0	9.62E-01	No	No
SaDSDE with $N_p = 250$ versus SaDSDE with $N_p = 225$	10	11	9.92E-01	No	No

Sign “No” indicates that the performance of SaDSDE with $N_p = 250$ is similar to its competitor.

<https://doi.org/10.1371/journal.pone.0222706.t003>

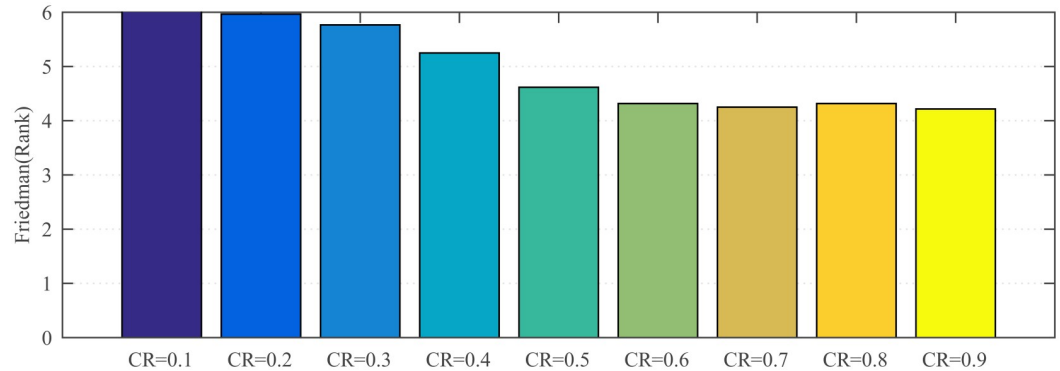


Fig 3. The result of Friedman test with 30 variables over 30 independent runs.

<https://doi.org/10.1371/journal.pone.0222706.g003>

Table 4. The results of Wilcoxon’s rank-sum test over 30 independent runs.

Comparisons	R+	R-	p-value	$\alpha = 0.05$	$\alpha = 0.1$
SaDSDE with CR = 0.9 versus SaDSDE with CR = 0.1	110	80	6.68E-02	No	Yes
SaDSDE with CR = 0.9 versus SaDSDE with CR = 0.2	109	81	6.68E-02	No	Yes
SaDSDE with CR = 0.9 versus SaDSDE with CR = 0.3	123	67	7.94E-02	No	Yes
SaDSDE with CR = 0.9 versus SaDSDE with CR = 0.4	108	63	1.11E-01	No	No
SaDSDE with CR = 0.9 versus SaDSDE with CR = 0.5	30	36	5.83E-01	No	No
SaDSDE with CR = 0.9 versus SaDSDE with CR = 0.6	21	45	6.06E-01	No	No
SaDSDE with CR = 0.9 versus SaDSDE with CR = 0.7	22	33	7.39E-01	No	No
SaDSDE with CR = 0.9 versus SaDSDE with CR = 0.8	29	7	7.22E-01	No	No

R+ is the sum of ranks in which the first algorithm outperformed the second, and R- is the sum of ranks for the opposite. The R+ is bigger, the first algorithm is better. “Yes” indicates that the performance of the SaDSDE with CR = 0.9 is better than its competitor significantly, “No” indicates that there is no significant performance discrepancy.

<https://doi.org/10.1371/journal.pone.0222706.t004>

4.5 Experimental comparisons

4.5.1 Numerical analysis. The optimization performance of DE is evaluated in terms of the Mean and Standard Deviation (STD) of the function solutions, and the best and the second

Table 5. Parameter settings.

Algorithm	Parameters
Rcr-JADE	$\mu_F = \mu_{CR} = 0.5, c = 0.1, p = 0.5, F = \text{randn}(\mu_F, 0.1), CR = \text{randn}(\mu_{CR}, 0.1)$
EsDer-NR	$\mu_F = \mu_{CR} = \text{freq} = 0.5, G_{LS} = 250, Np_{max} = 18 * D, Np_{min} = 4$
IMMSADE	$T = 0.7, \lambda \in [0.7, 1.0], Fe \in [0.1, 0.8], CR \in [0.3, 1.0]$
AGDE	$CR_1 \in [0.05, 0.15], CR_2 \in [0.9, 1.0], F \in [0.1, 1.0], p = 0.1$
EFADE	$\epsilon = 0.01, CR_1 \in [0.05, 0.15], CR_2 \in [0.9, 1.0]$
MPEDE	$c = 0.1, p = 0.4, \lambda_1 = \lambda_2 = \lambda_3$
EDEV	$\lambda_1 = \lambda_2 = \lambda_3 = 0.1, \lambda_4 = 0.7, ng = 20$
SL-PSO	$\alpha = 0.5, \beta = 0.01$
GWO	$a = 2 - (\frac{2}{T}), A = 2 * a * \text{rand}() - a, C = 2 * \text{rand}()$
WOA	$p = \text{rand}(), a = 2 - t * (\frac{2}{T}), A = 2 * a * \text{rand}() - a, C = 2 * \text{rand}()$

<https://doi.org/10.1371/journal.pone.0222706.t005>

best results are respectively marked in bold and italic. In order to show the performance clearly, the number of the best and second best results are summarize graphically. In addition, 4 non-parametric statistical tests [50] are used to compare the performances among all the algorithms, including Wilcoxon signed-rank test, Friedman test, Kruskal_Wallis test and Wilcoxon rank-sum test. Friedman test and Kruskal_Wallis test are conducted based on the optimization results to evaluate the overall performance and we can obtain the rankings of algorithms. Wilcoxon signed-rank test at the 0.05 significance level and Wilcoxon rank-sum test respectively are used to verify the differences among algorithms for single and multiple problems. Signs “+”, “-” and “≈” indicate that the performance of SaDSDE is significantly better than, worse than and similar to its competitor, respectively. R^+ denotes the sum of ranks for the test problems in which the first algorithm performs better than the second algorithm, and R^- represents the sum of ranks for the test problems in which the first algorithm performs worse than the second algorithm. Larger ranks indicate larger performance discrepancy. “p-value” indicates the probability of rejecting the hypothesis. “Yes” indicates that the performance of the first algorithm is better than the second algorithm significantly, and “No” indicates that there is no significant performance discrepancy.

Table 6. The optimization results obtained by SaDSDE and 7 DE variants at $D = 30$.

F	Rcr-JADE	EsDEr-NR	IMMSADE	AGDE	EFADE	MPEDE	EDEV	SaDSDE
	Mean ± STD	Mean ± STD	Mean ± STD	Mean ± STD	Mean ± STD	Mean ± STD	Mean ± STD	Mean ± STD
f_1	3.52E-51±1.06E-50	<i>1.80E-54±1.22E-54</i>	4.41E-28±1.67E-27	2.25E-09±6.18E-10	3.64E-12±5.56E-12	7.76E-36±2.31E-35	2.54E-31±1.09E-30	0.00E+00±0.00E+00
f_2	1.74E-13±4.67E-13	1.32E-11±3.14E-11	6.51E+00±1.63E+01	1.72E+00±9.76E-01	2.52E+00±2.63E+00	<i>1.70E-21±8.93E-21</i>	1.19E-07±3.35E-07	0.00E+00±0.00E+00
f_3	3.75E-46±9.37E-46	<i>3.04E-50±4.19E-50</i>	8.56E-26±4.67E-25	5.24E-06±2.57E-06	4.69E-10±4.27E-10	1.55E-31±4.74E-31	1.40E-26±7.39E-26	0.00E+00±0.00E+00
f_4	1.12E-24±2.60E-24	<i>4.60E-26±2.88E-26</i>	7.96E-17±3.30E-16	2.42E-06±3.51E-07	2.80E-05±6.21E-06	2.19E-17±3.78E-17	1.18E-14±4.36E-14	0.00E+00±0.00E+00
f_5	2.09E-11±2.50E-11	5.33E-13±1.44E-12	2.72E-08±4.78E-08	1.18E+00±1.74E-01	2.72E+00±4.92E-01	<i>5.39E-14±4.81E-14</i>	1.94E-02±1.82E-02	0.00E+00±0.00E+00
f_6	9.28E-52±1.68E-51	<i>3.07E-55±2.46E-55</i>	4.15E-31±1.46E-30	2.53E-10±9.49E-11	2.08E-13±2.06E-13	1.28E-37±5.09E-37	5.82E-31±2.72E-30	0.00E+00±0.00E+00
f_7	8.73E-45±2.61E-44	<i>2.69E-48±2.55E-48</i>	9.11E-25±3.50E-24	2.39E-03±6.11E-04	3.65E-06±8.10E-06	1.93E-31±3.95E-31	3.58E-26±1.63E-25	0.00E+00±0.00E+00
f_8	3.44E-10±7.06E-10	4.36E-09±1.12E-08	4.02E+00±3.42E+00	3.41E-09±7.38E-10	<i>9.11E-13±1.11E-12</i>	1.42E-06±6.54E-06	2.42E-05±8.19E-05	0.00E+00±0.00E+00
f_9	3.81E-44±8.30E-44	<i>5.54E-47±1.10E-46</i>	1.02E-20±5.50E-20	1.52E-03±4.89E-04	1.78E-06±3.36E-06	3.43E-30±9.72E-30	1.26E-26±2.58E-26	0.00E+00±0.00E+00
f_{10}	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00
f_{11}	1.25E+03±8.23E-06	2.13E-03±7.04E-04	1.25E+03±5.75E-06	2.01E-02±4.87E-03	4.50E-03±1.77E-03	1.25E+03±5.04E-06	<i>2.56E-03±7.91E-04</i>	1.25E+03±3.83E-06
f_{12}	3.15E-02±2.32E-02	6.16E-02±2.25E-01	4.13E+01±3.25E+01	7.32E+00±2.83E+00	2.60E+01±4.49E+00	<i>3.13E-11±3.41E-11</i>	6.07E-05±2.45E-04	0.00E+00±0.00E+00
f_{13}	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	1.07E-06±3.25E-06	<i>1.37E-09±1.18E-09</i>	2.47E-04±1.35E-03	0.00E+00±0.00E+00	0.00E+00±0.00E+00
f_{14}	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	9.19E-14±2.57E-14	<i>1.07E-16±5.88E-16</i>	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00
f_{15}	3.30E-52±5.98E-52	<i>2.50E-55±2.54E-55</i>	1.05E-26±5.75E-26	6.44E-10±2.30E-10	2.53E-12±2.00E-12	3.55E-38±6.14E-38	6.53E-30±3.55E-29	0.00E+00±0.00E+00
f_{16}	5.68E-15±1.77E-15	<i>3.67E-15±6.49E-16</i>	3.79E-15±1.30E-15	1.13E-05±1.48E-06	8.46E-05±2.09E-05	<i>3.67E-15±6.49E-16</i>	3.79E-15±9.01E-16	0.00E+00±0.00E+00
f_{17}	3.15E-01±1.02E+00	1.11E+01±1.95E+00	2.59E+01±1.53E-01	4.13E+01±2.94E+01	3.17E+01±2.10E+01	<i>5.63E-01±1.38E+00</i>	5.71E+00±1.58E+00	2.35E+01±6.98E-01
f_{18}	8.28E-02±5.98E-02	9.30E-02±1.66E-01	<i>2.42E-05±5.63E-05</i>	9.75E-01±1.16E-01	4.52E+00±3.29E-01	3.64E-03±6.38E-03	1.71E-01±8.76E-02	0.00E+00±0.00E+00
f_{19}	3.82E-04±0.00E+00	3.82E-04±0.00E+00	3.82E-04±0.00E+00	3.82E-04±5.40E-11	3.82E-04±2.88E-09	3.82E-04±0.00E+00	3.82E-04±0.00E+00	3.82E-04±0.00E+00
f_{20}	<i>2.22E-01±3.61E-02</i>	2.36E-01±3.32E-02	4.34E-01±6.80E-02	4.24E-01±3.81E-02	4.74E-01±6.65E-02	2.18E-01±2.58E-02	2.67E-01±3.78E-02	5.29E-01±8.87E-02
f_{21}	3.83E-01±1.18E-01	3.73E-01±1.36E-01	3.97E-01±3.27E-02	2.79E-01±2.39E-02	3.32E-01±2.92E-02	<i>3.20E-01±8.86E-02</i>	3.59E-01±1.08E-01	5.00E-01±0.00E+00
f_{22}	5.59E-04±1.70E-03	1.39E-03±5.45E-03	0.00E+00±0.00E+00	8.61E-04±1.29E-04	7.15E-03±7.52E-04	0.00E+00±0.00E+00	<i>1.01E-07±4.34E-07</i>	0.00E+00±0.00E+00
f_{23}	5.15E-03±7.43E-04	5.01E-03±1.21E-03	6.01E-01±8.73E-02	1.14E-01±1.17E-02	1.14E-01±1.50E-02	<i>1.34E-03±3.70E-04</i>	1.00E-02±1.74E-03	0.00E+00±0.00E+00
f_{24}	4.09E+00±3.25E-01	<i>4.85E-01±6.53E-02</i>	8.70E+00±4.92E-01	1.23E+00±9.76E-02	4.80E+00±3.90E-01	2.03E+00±3.35E-01	1.22E+00±1.98E-01	0.00E+00±0.00E+00
f_{25}	4.14E+00±3.30E-01	2.08E+00±1.82E-01	1.19E+01±1.08E+00	5.71E+00±3.97E-01	6.71E+00±7.35E-01	<i>2.32E+00±2.58E-01</i>	3.54E+00±5.65E-01	1.25E+01±7.66E-01
f_{26}	8.46E+00±1.30E+00	6.31E-01±5.86E-01	5.20E+01±7.55E+00	1.57E+01±8.42E-01	2.11E+01±1.13E+00	<i>4.81E-04±7.07E-04</i>	5.95E-01±5.12E-01	0.00E+00±0.00E+00
f_{27}	1.02E-03±7.39E-04	6.29E-07±3.44E-06	<i>4.20E-15±1.23E-14</i>	3.77E-03±4.32E-04	9.22E-03±8.62E-04	1.58E-05±1.52E-05	1.82E-03±6.17E-04	0.00E+00±0.00E+00
f_{28}	3.50E-02±1.92E-01	1.38E-02±7.54E-02	0.00E+00±0.00E+00	2.17E-07±7.59E-08	<i>1.41E-08±1.73E-08</i>	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00
f_{29}	1.57E-32±5.57E-48	1.57E-32±5.57E-48	1.68E-03±6.77E-04	2.95E-11±1.40E-11	1.51E-12±2.76E-12	1.57E-32±5.57E-48	<i>1.72E-31±4.26E-31</i>	1.34E-10±2.75E-10
f_{30}	1.68E-32±1.40E-33	<i>1.37E-32±4.26E-34</i>	4.72E-03±1.89E-03	6.35E-13±2.51E-13	4.89E-16±6.39E-16	1.35E-32±5.57E-48	1.34E-31±4.59E-31	1.24E-01±3.61E-01

<https://doi.org/10.1371/journal.pone.0222706.t006>

A. Comparison with 7 improved DE variants. In this experiment, the proposed algorithm is compared with 7 improved DE variants on 30-dimensional and 100-dimensional problems. The experimental results are listed in Tables 6 and 7. To show the performance clearly, we summarize the number of the best, the second best results and global optimum graphically in Figs 4 and 5, respectively. Wilcoxon signed-rank test results and the average rankings of all the DEs are shown in Figs 6 and 7, respectively. In addition, Wilcoxon rank-sum test results are represented in Table 8.

From Tables 6 and 7, we can draw the following conclusions: (1) SaDSDE can obtain the global optimal solution on 10 unimodal functions (f_1 - f_{10}) and 12 multimodal functions (f_{12} - f_{16} , f_{18} , f_{20} , f_{22} - f_{24} and f_{25} - f_{28}). (2) Except for f_{11} , f_{17} , f_{20} - f_{21} , f_{27} and f_{29} - f_{30} , the convergence precision of SaDSDE is best at $D = 30$ and $D = 100$, and SaDSDE is comparable to the other compared algorithms on f_{19} , f_{22} and f_{28} . In addition, at $D = 30$, Rcr-JADE is best on f_{17} and f_{29} , EsDER-NR is best on f_{11} , f_{25} and f_{29} , IMMSADE is best on f_{22} , AGDE is best on f_{21} , MPEDE is best on f_{17} , f_{20} and f_{29} ; at $D = 100$, EsDER-NR is best on f_{11} , IMMSADE is best on f_{17} , MPEDE is best on f_{20} - f_{21} , f_{25} and f_{29} , EDEV is best on f_{30} . (3) The optimization performance of other algorithms

Table 7. The optimization results obtained by SaDSDE and 7 DE variants at $D = 100$.

F	Rcr-JADE	EsDER-NR	IMMSADE	AGDE	EFADE	MPEDE	EDEV	SaDSDE
	Mean ± STD	Mean ± STD	Mean ± STD	Mean ± STD	Mean ± STD	Mean ± STD	Mean ± STD	Mean ± STD
f_1	1.36E-15±1.51E-15	5.47E-04±1.44E-03	4.76E-17±1.81E-16	3.97E+00±7.32E-01	2.56E+01±1.33E+01	1.74E-16±3.07E-16	8.17E-10±6.88E-10	0.00E+00±0.00E+00
f_2	1.99E+02±6.49E+01	6.55E+02±1.97E+02	1.22E+04±1.44E+04	5.84E+04±3.33E+04	9.93E+03±2.34E+03	7.22E+00±3.38E+00	3.92E+02±1.47E+02	0.00E+00±0.00E+00
f_3	8.36E-10±1.13E-09	1.12E+03±2.52E+03	5.96E-12±2.85E-11	5.77E+03±9.39E+02	4.38E+03±2.61E+03	7.56E-11±9.07E-11	2.23E-06±1.63E-06	0.00E+00±0.00E+00
f_4	2.58E-06±4.13E-06	7.09E-02±7.71E-02	1.73E-09±4.51E-09	7.55E-01±5.60E-02	2.59E+00±4.60E-01	4.42E-08±3.95E-08	2.05E-06±3.04E-06	0.00E+00±0.00E+00
f_5	8.01E+00±1.32E+00	1.01E+01±1.92E+00	5.71E-04±1.39E-03	4.04E+01±1.73E+00	5.03E+01±3.01E+00	3.97E-01±1.20E-01	1.20E+01±1.60E+00	0.00E+00±0.00E+00
f_6	1.16E-15±2.05E-15	4.77E-04±1.15E-03	3.44E-13±1.88E-12	1.52E+00±2.11E-01	9.31E+00±5.25E+00	5.73E-17±8.31E-17	2.82E-10±1.74E-10	0.00E+00±0.00E+00
f_7	3.98E-09±4.50E-09	7.89E+02±2.54E+03	2.08E-11±1.04E-10	3.88E+06±7.56E+05	2.57E+07±1.11E+07	1.35E-10±1.39E-10	1.79E-04±1.71E-04	0.00E+00±0.00E+00
f_8	4.88E+00±2.34E+00	5.02E+00±1.14E+01	8.63E+01±4.07E+01	4.65E-01±1.94E-01	2.23E-01±1.35E-01	1.19E+01±1.71E+01	1.22E+00±1.55E+00	0.00E+00±0.00E+00
f_9	2.49E-08±3.31E-08	9.87E+02±1.28E+03	5.01E-10±2.54E-09	3.77E+06±7.22E+05	2.29E+07±1.33E+07	5.77E-09±1.59E-08	2.48E-04±3.05E-04	0.00E+00±0.00E+00
f_{10}	6.47E+00±4.86E+00	7.61E+01±2.96E+01	0.00E+00±0.00E+00	2.70E+00±1.82E+00	3.20E+01±1.21E+01	1.97E+00±1.71E+00	3.20E+00±5.62E+00	0.00E+00±0.00E+00
f_{11}	1.36E+04±6.12E-06	5.25E-02±1.30E-02	1.36E+04±5.84E-06	2.43E-01±4.77E-02	6.48E-02±1.57E-02	1.36E+04±4.05E-06	8.02E-02±1.49E-02	1.36E+04±5.13E-06
f_{12}	2.02E+02±1.12E+01	1.02E+02±9.02E+00	1.36E+02±2.20E+02	4.52E+02±1.36E+01	5.61E+02±1.45E+01	4.48E+01±5.65E+00	1.97E+02±1.60E+01	0.00E+00±0.00E+00
f_{13}	3.28E-03±5.50E-03	1.73E-02±3.31E-02	3.38E-15±1.60E-14	9.53E-01±5.43E-02	1.38E+00±1.36E-01	4.02E-03±7.30E-03	7.11E-03±1.52E-02	0.00E+00±0.00E+00
f_{14}	2.35E-16±1.91E-16	3.37E-08±1.36E-07	0.00E+00±0.00E+00	2.17E-04±3.38E-05	1.25E-03±8.02E-04	0.00E+00±0.00E+00	5.87E-14±2.84E-14	0.00E+00±0.00E+00
f_{15}	1.19E-15±1.61E-15	7.59E-05±1.19E-04	1.46E-15±7.94E-15	1.08E+00±1.11E-01	4.52E+00±1.19E+00	2.27E-17±4.40E-17	1.90E-10±7.61E-11	0.00E+00±0.00E+00
f_{16}	1.29E+00±3.98E-01	2.75E+00±3.43E-01	9.97E-10±2.30E-09	4.26E-01±4.69E-02	3.91E+00±2.44E-01	1.61E-01±3.68E-01	1.37E+00±4.04E-01	0.00E+00±0.00E+00
f_{17}	1.42E+02±4.35E+01	3.42E+02±1.50E+02	9.69E+01±2.86E-01	2.46E+04±5.24E+03	1.08E+03±4.62E+02	1.13E+02±3.47E+01	3.33E+02±1.50E+02	9.70E+01±6.77E-01
f_{18}	2.75E+01±1.38E+01	1.01E+02±1.52E+01	8.48E-03±1.44E-02	1.65E+02±6.67E+00	3.72E+02±1.57E+01	2.97E+01±9.40E+00	1.25E+01±4.45E+00	0.00E+00±0.00E+00
f_{19}	1.27E-03±5.03E-12	1.30E-03±6.94E-05	1.27E-03±0.00E+00	4.27E-01±6.59E-02	3.19E+00±1.55E+00	1.27E-03±1.33E-12	1.27E-03±3.26E-11	1.27E-03±0.00E+00
f_{20}	5.31E-01±6.62E-02	6.57E-01±6.67E-02	8.54E-01±6.00E-02	7.96E-01±6.00E-02	2.31E+00±9.65E-01	4.94E-01±5.71E-02	5.22E-01±6.80E-02	9.31E-01±6.87E-02
f_{21}	6.04E-01±2.01E-01	5.44E-01±2.46E-01	5.01E-01±6.34E-02	7.11E-01±2.89E-01	2.12E+01±1.42E+01	4.97E-01±1.63E-01	5.57E-01±1.93E-01	5.00E-01±0.00E+00
f_{22}	4.54E+00±1.79E+00	1.29E+01±2.24E+00	1.54E-06±6.70E-06	4.19E+00±2.68E-01	1.99E+01±1.48E+00	3.43E+00±1.10E+00	1.71E+00±8.96E-01	0.00E+00±0.00E+00
f_{23}	1.17E-01±1.16E-02	1.08E-01±1.53E-02	2.00E+00±1.58E-01	5.71E-01±4.70E-02	5.87E-01±4.06E-02	1.35E-02±2.97E-03	1.87E-01±5.13E-02	0.00E+00±0.00E+00
f_{24}	3.35E+01±5.89E-01	5.78E+00±4.78E-01	4.20E+01±7.16E-01	1.83E+01±6.23E-01	3.81E+01±4.67E-01	2.46E+01±1.70E+00	1.50E+01±4.21E+00	0.00E+00±0.00E+00
f_{25}	3.67E+01±2.09E+00	2.15E+01±1.62E+00	6.67E+01±7.36E+00	5.84E+01±1.76E+00	9.89E+01±2.59E+01	2.13E+01±2.16E+00	3.91E+01±2.64E+00	4.39E+01±7.11E-01
f_{26}	1.47E+02±6.66E+00	7.72E+01±4.76E+00	4.75E+02±9.81E+01	3.49E+02±2.30E+01	4.22E+02±2.11E+01	6.95E+01±6.31E+00	1.40E+02±8.54E+00	0.00E+00±0.00E+00
f_{27}	1.99E-06±9.67E-06	7.80E-03±9.03E-03	1.43E-08±7.61E-08	2.50E+00±6.58E-01	4.82E+01±2.03E+00	1.08E-08±1.56E-08	7.27E-04±1.74E-03	0.00E+00±0.00E+00
f_{28}	8.89E+00±2.09E+00	2.21E+01±3.01E+00	4.68E-15±2.50E-14	4.16E+01±2.13E+00	1.21E+02±4.60E+01	5.22E+00±1.68E+00	7.73E+00±2.00E+00	0.00E+00±0.00E+00
f_{29}	1.56E-03±8.57E-03	7.21E-05±2.34E-04	3.03E-02±9.81E-03	1.22E-01±2.64E-02	4.31E-01±1.38E-01	8.74E-20±1.10E-19	3.30E-11±1.23E-10	3.96E-02±2.27E-02
f_{30}	2.56E-03±4.73E-03	3.66E-03±5.27E-03	3.45E-01±9.96E-02	1.92E-03±2.83E-04	8.57E-03±5.20E-03	2.56E-03±4.73E-03	1.39E-03±5.91E-03	9.33E+00±1.73E+00

<https://doi.org/10.1371/journal.pone.0222706.t007>

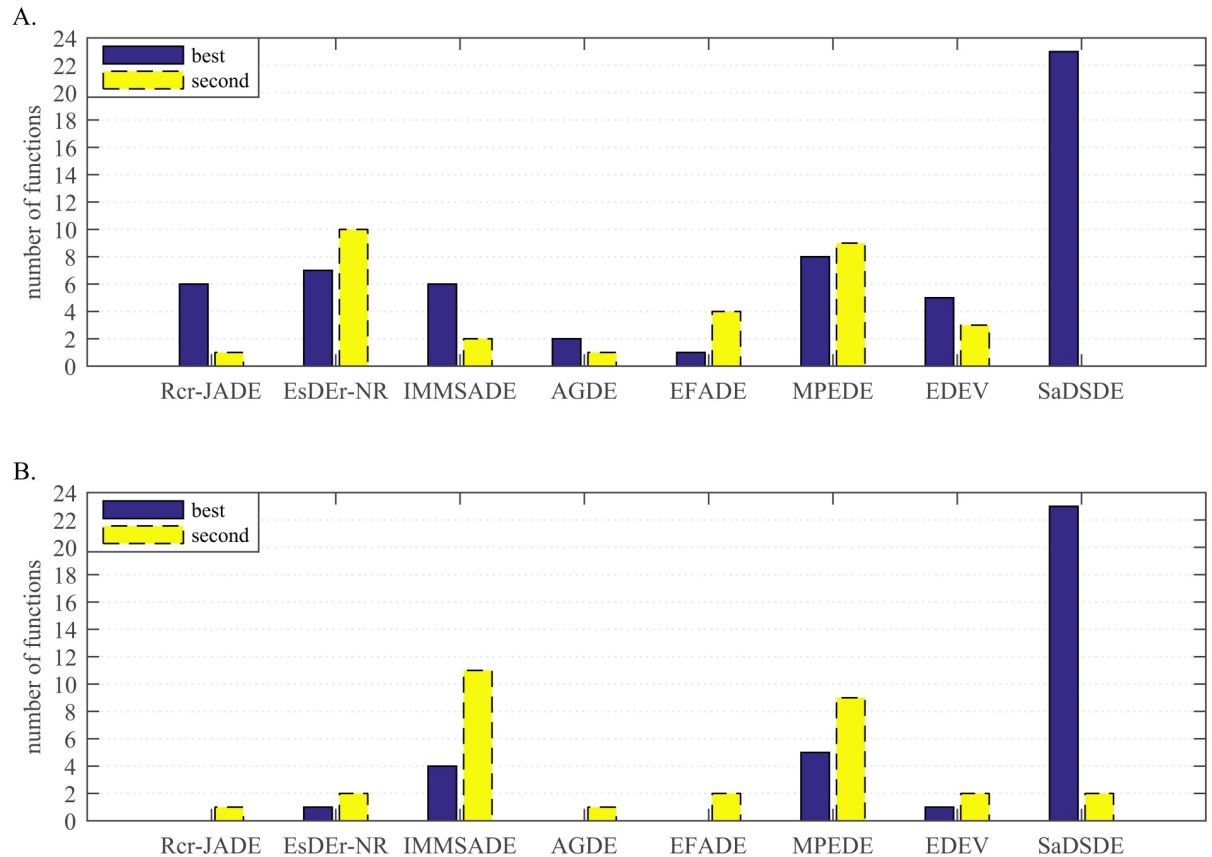


Fig 4. Number of cases on which each algorithm performs the best and second best in the comparison. (A) $D = 30$. (B) $D = 100$.

<https://doi.org/10.1371/journal.pone.0222706.g004>

decreases rapidly for the high dimensions, but SaDSDE still keep the excellent optimization performance.

Form the statistical histogram in Fig 4, SaDSDE and Rcr-JADE, EsDEr-NR, IMMSADE, AGDE, EFADE, MPEDE and EDEV can obtain the best results on 23, 6, 7, 6, 2, 1, 8 and 5 functions and obtain the second best results on 0, 1, 10, 2, 1, 4, 9 and 3 functions at $D = 30$, respectively. For the high-dimensional problem tests, the advantage of the proposed SaDSDE is more prominent. SaDSDE and 7 improved DE variants can obtain the best results on 23, 0, 1, 4, 0, 0, 5 and 1 functions and obtain the second best results on 2, 1, 2, 11, 1, 2, 9 and 2 functions at $D = 100$, respectively.

According to the statistical results in Fig 5, SaDSDE can obtain global optimums on 22 (= 10 unimodal cases+12 multimodal cases) functions in all dimensions, while 7 DE variants in turn obtain global optimums on 3 (= 1+2), 3 (= 1+2), 5 (= 1+4), 1 (= 1+0), 1 (= 1+0), 4 (= 1 +3) and 4 (= 1+3) functions at $D = 30$. As the dimension increases, the number of global optimum decreases greatly for compared algorithms.

As shown in Fig 6, SaDSDE is significantly better than Rcr-JADE, EsDEr-NR, IMMSADE, AGDE, EFADE, MPEDE and EDEV on 19, 18, 19, 22, 21, 17 and 18 functions out of 30 functions at $D = 30$, respectively. On the contrary, SaDSDE is worse than 7 DE variants only on 6, 7, 2, 5, 6, 6 and 7 functions, respectively. At $D = 100$, SaDSDE is significantly better than all the competitors at least in 20 out of 30 functions and the advantage of proposed SaDSDE is more prominent.

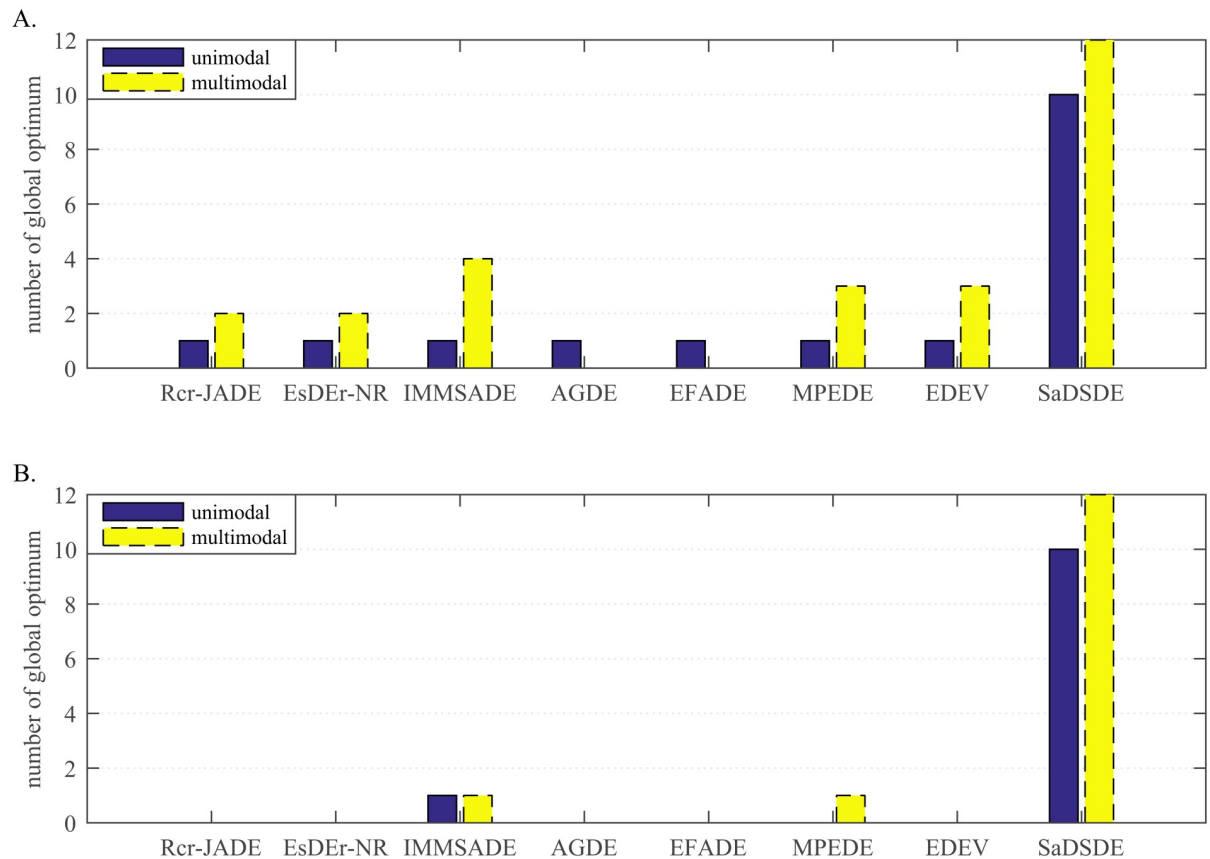


Fig 5. Number of cases on which each algorithm obtains global optimum. (A) $D = 30$. (B) $D = 100$.

<https://doi.org/10.1371/journal.pone.0222706.g005>

According to Fig 7, SaDSDE is the best, and EsDEr-NR and MPEDE are respectively the second best at $D = 30$ and $D = 100$.

As shown in Table 8, we can observe that SaDSDE gets higher R+ values than R- values for all the compared DEs, and all the p -values are less than 0.05. It proves that SaDSDE outperforms other compared DE algorithms significantly.

B. Comparison with 3 non-DE algorithms. DE simulates the natural evolution (such as mutation, crossover and selection) for global search. Unlike DE algorithm, SL-PSO [51], GWO [52] and WOA [53] are swarm intelligence optimization algorithms which simulate the social behavior of natural biological groups (such as foraging, nesting, migration, hunting, predation, etc.). SL-PSO, GWO and WOA are highly competitive swarm intelligence optimization algorithms. Therefore, in order to further verify the effectiveness of SaDSDE algorithm, SaDSDE is compared with SL-PSO, GWO and WOA algorithms. The obtained results (i.e., Mean and STD) and Wilcoxon signed-rank test results at the 0.05 significance level are shown in Tables 9 and 10. The number of the best and the second best results are shown in Fig 8. Additionally, the statistical analysis results for all test functions are listed in Fig 9 and the Wilcoxon rank-sum test results are represented in Table 11.

Analyzing the test results in Tables 9 and 10, SaDSDE is significantly better than SL-PSO, GWO and WOA on 17, 20 and 15 functions out of 30 functions at $D = 30$, respectively. At $D = 100$, SaDSDE performs better than 3 non-DE algorithms on 23, 19 and 12 functions. From the diagram in Fig 8, SaDSDE obtains best results on 24 and 23 functions at $D = 30$ and

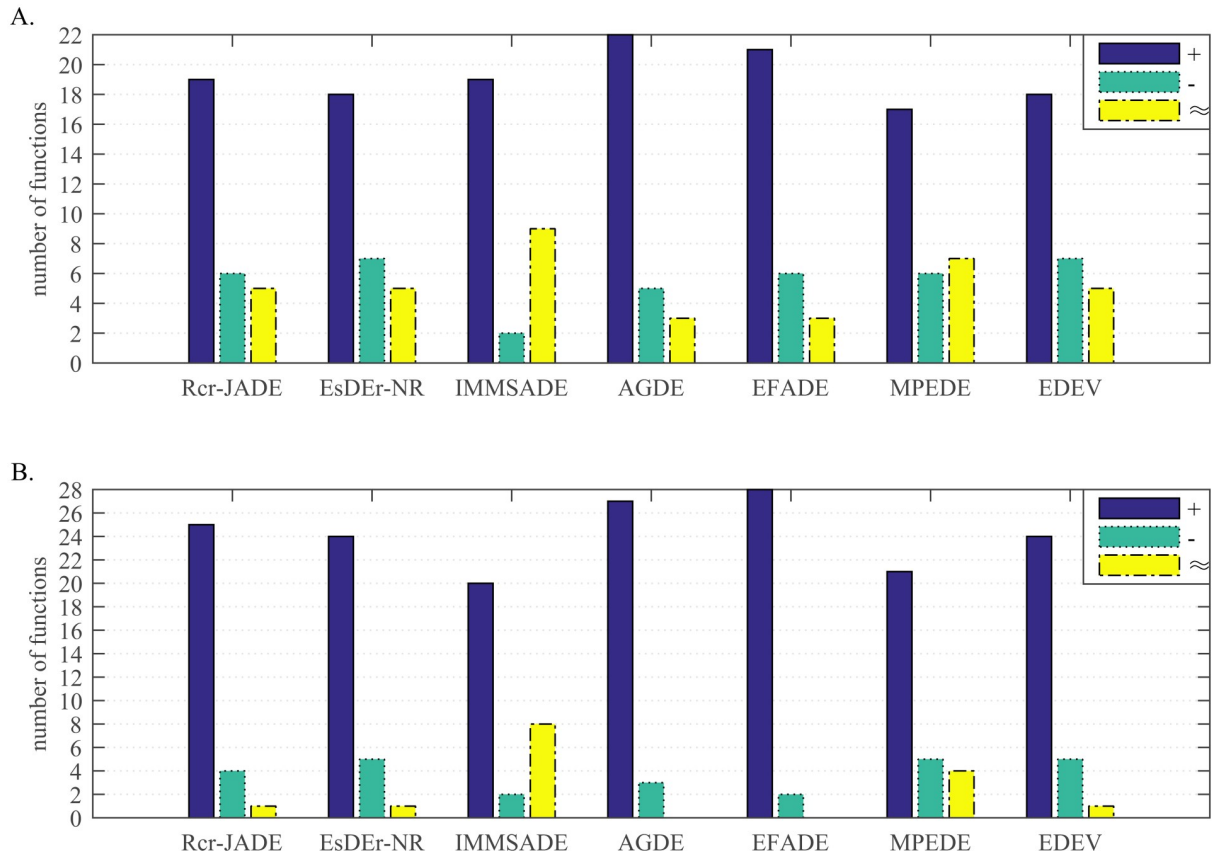


Fig 6. Wilcoxon signed-rank test results at the 0.05 significance level. (A) $D = 30$. (B) $D = 100$.

<https://doi.org/10.1371/journal.pone.0222706.g006>

$D = 100$, respectively. Fig 9 shows that SaDSDE gets the best ranking, followed by WOA. SaDSDE is better than 3 non-DE variants significantly, as shown in Table 11.

4.5.2 Convergence analysis. In order to study the evolution properties, convergence curves are utilized to prove the difference, as illustrated in Fig 10. Furthermore, to illustrate the distribution of the results of each algorithm, the box plots of function solutions of all the algorithms on 6 functions (1 unimodal function and 5 multimodal functions) at $D = 30$ are depicted in Fig 11.

Fig 10 shows that SaDSDE can obtain the global optimum and has the best convergence rate on 6 functions in the minimum generations, while other compared algorithms have straggled into "evolution stagnation" on different functions. For example, Non-Continuous Rastrigin's Function (f_{26}) is multi-modal, non-separable and asymmetrical function and the local optima's number is huge, DE is easy to suffer premature convergence to be trapped in one of its many local minima. SaDSDE can find the global optimal solution. However, IMMSADE, EFADE, AGDE and GWO have trapped into "evolution stagnation". In addition, we can see that SaDSDE performs more consistently than other algorithms on these problems, as shown in Fig 11.

As evident from above analysis, the proposed SaDSDE demonstrates the better performance of convergence precision and robustness than most of the compared algorithms. To summarize, the proposed SaDSDE has the best convergence precision and stability. The reason is that the proposed mutation strategy, scaling factor self-adaption strategy and dynamic

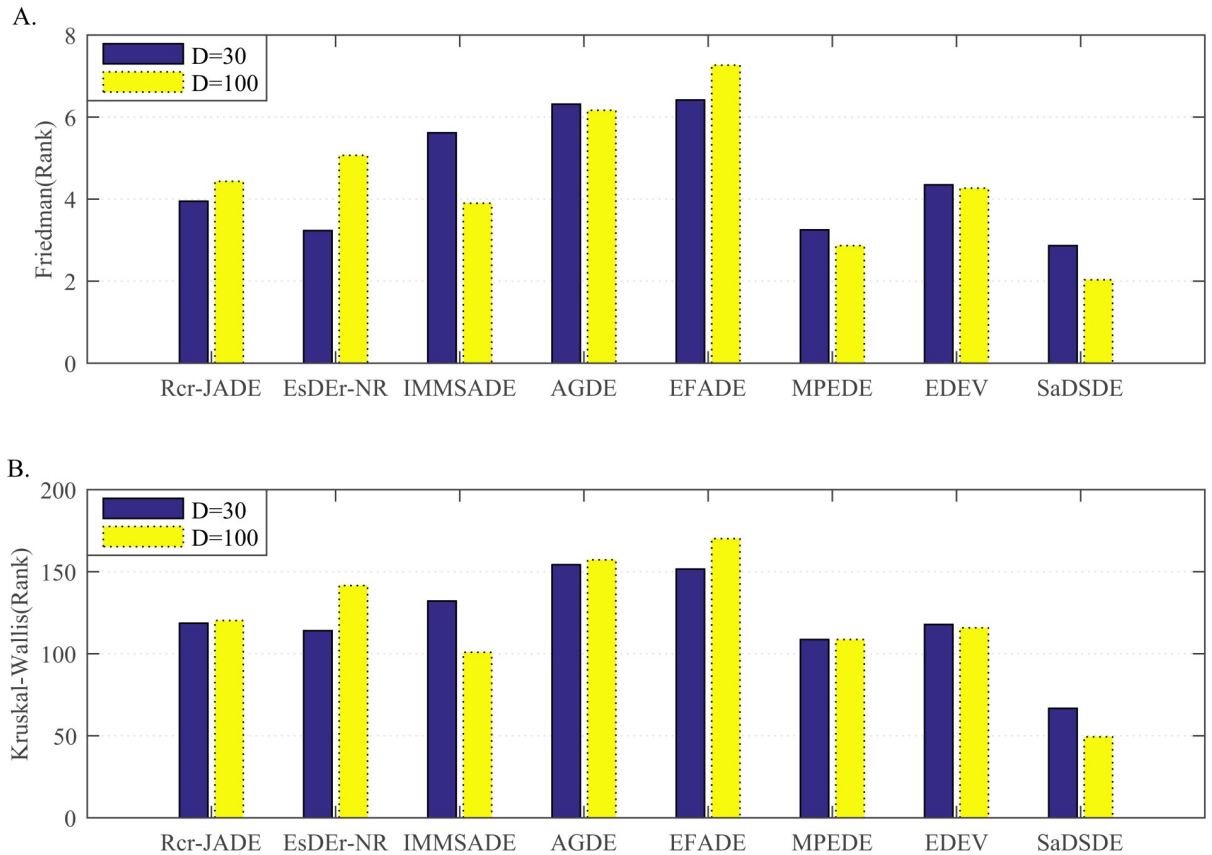


Fig 7. Non-parametric statistical test results over 30 independent runs. (A) Friedman test results. (B) Kruskal-Wallis test results.

<https://doi.org/10.1371/journal.pone.0222706.g007>

adjustment strategy of the exploration ability control factor achieve a better balance of exploration and exploitation.

4.6 Efficiency analysis of proposed algorithmic components

The proposed algorithm represents a combined effect. Therefore, we do the efficiency analysis of proposed algorithmic components, including dual-strategy mutation operator, scaling factor self-adaptation strategy and dynamic adjustment strategy of exploration ability control factor. Some variants of SaDSDE are listed as follows.

Table 8. The results of Wilcoxon rank-sum test over 30 independent runs.

Comparisons	D = 30					D = 100				
	R+	R-	p-value	$\alpha = 0.05$	$\alpha = 0.1$	R+	R-	p-value	$\alpha = 0.05$	$\alpha = 0.1$
SaDSDE versus Rcr-JADE	227	124	6.19E-04	Yes	Yes	391	74	6.94E-06	Yes	Yes
SaDSDE versus EsDEr-NR	205	146	9.15E-04	Yes	Yes	379	86	1.37E-06	Yes	Yes
SaDSDE versus IMMSADE	238	62	5.76E-04	Yes	Yes	303	75	1.04E-04	Yes	Yes
SaDSDE versus AGDE	323	112	3.84E-05	Yes	Yes	419	46	2.24E-07	Yes	Yes
SaDSDE versus EFADE	319	116	5.00E-05	Yes	Yes	425	40	6.47E-08	Yes	Yes
SaDSDE versus MPEDE	188	137	1.65E-03	Yes	Yes	336	99	3.37E-05	Yes	Yes
SaDSDE versus EDEV	189	136	1.34E-03	Yes	Yes	366	99	1.06E-05	Yes	Yes

<https://doi.org/10.1371/journal.pone.0222706.t008>

Table 9. The optimization results obtained by SaDSDE and 3 non-DE algorithms at $D = 30$.

F	SL-PSO	GWO	WOA	SaDSDE
	Mean \pm STD	Mean \pm STD	Mean \pm STD	Mean \pm STD
f_1	2.83E-44 \pm 3.37E-44	2.34E-85 \pm 4.17E-85	1.60E-191 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_2	6.78E-02 \pm 4.83E-02	9.46E-27 \pm 4.30E-26	4.41E+03 \pm 3.15E+03	0.00E+00 \pm 0.00E+00
f_3	1.76E-38 \pm 6.23E-38	1.10E-81 \pm 2.91E-81	9.44E-186 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_4	1.98E-23 \pm 1.12E-23	2.59E-49 \pm 2.33E-49	2.22E-114 \pm 7.19E-114	0.00E+00 \pm 0.00E+00
f_5	4.36E-12 \pm 2.46E-12	1.01E-21 \pm 1.41E-21	6.46E-08 \pm 1.90E-07	0.00E+00 \pm 0.00E+00
f_6	4.65E-45 \pm 7.14E-45	3.08E-86 \pm 9.99E-86	2.07E-188 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_7	3.63E-38 \pm 4.09E-38	2.29E-79 \pm 6.05E-79	1.82E-184 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_8	1.15E+02 \pm 3.95E+01	8.78E-83 \pm 2.04E-82	6.44E+01 \pm 6.70E+01	0.00E+00 \pm 0.00E+00
f_9	5.79E-38 \pm 1.16E-37	8.18E-80 \pm 1.17E-79	1.42E-184 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{10}	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{11}	1.25E+03 \pm 9.78E-06	2.98E-04 \pm 1.58E-04	1.87E-05 \pm 2.12E-05	1.25E+03 \pm 3.83E-06
f_{12}	1.34E+01 \pm 3.19E+00	1.47E-01 \pm 8.04E-01	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{13}	9.85E-04 \pm 3.21E-03	1.45E-03 \pm 4.47E-03	3.20E-03 \pm 1.04E-02	0.00E+00 \pm 0.00E+00
f_{14}	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{15}	4.02E-45 \pm 4.18E-45	1.86E-86 \pm 3.46E-86	6.82E-194 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{16}	6.16E-15 \pm 1.60E-15	1.05E-14 \pm 2.87E-15	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{17}	4.84E+01 \pm 4.58E+01	2.62E+01 \pm 5.51E-01	2.59E+01 \pm 2.21E-01	2.35E+01 \pm 6.98E-01
f_{18}	2.45E-02 \pm 2.39E-02	1.30E-22 \pm 1.07E-22	2.18E-65 \pm 6.09E-65	0.00E+00 \pm 0.00E+00
f_{19}	3.82E-04 \pm 0.00E+00	3.82E-04 \pm 7.40E-13	3.82E-04 \pm 0.00E+00	3.82E-04 \pm 0.00E+00
f_{20}	2.19E-01 \pm 4.18E-02	5.35E-01 \pm 7.87E-02	3.68E-01 \pm 7.32E-02	5.29E-01 \pm 8.87E-02
f_{21}	4.18E-01 \pm 7.51E-02	4.31E-01 \pm 5.58E-02	3.30E-01 \pm 7.09E-02	5.00E-01 \pm 0.00E+00
f_{22}	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{23}	0.00E+00 \pm 0.00E+00	1.69E-01 \pm 4.27E-02	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{24}	3.77E-01 \pm 1.85E-01	6.10E-01 \pm 3.57E-01	1.06E+00 \pm 1.39E+00	0.00E+00 \pm 0.00E+00
f_{25}	4.30E+00 \pm 2.05E+00	8.17E+00 \pm 1.38E+00	0.00E+00 \pm 0.00E+00	1.25E+01 \pm 7.66E-01
f_{26}	2.55E+01 \pm 8.28E+00	1.90E+00 \pm 5.18E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{27}	4.07E-17 \pm 1.55E-16	5.24E-05 \pm 1.55E-04	3.63E-01 \pm 1.99E+00	0.00E+00 \pm 0.00E+00
f_{28}	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{29}	1.57E-32 \pm 5.57E-48	8.40E-02 \pm 5.10E-02	1.91E-01 \pm 8.57E-01	1.34E-10 \pm 2.75E-10
f_{30}	1.10E-03 \pm 3.35E-03	1.39E-01 \pm 1.39E-01	5.26E-03 \pm 5.28E-03	1.24E-01 \pm 3.61E-01
+	17	20	15	-
-	5	3	4	-
\approx	8	7	11	-

<https://doi.org/10.1371/journal.pone.0222706.t009>

1. To study the effectiveness of scaling factor self-adaptation strategy, SaDSDE variants adopt dynamic λ and fixed scaling factor of $F = 0.3$, $F = 0.7$ and random real number in $[0, 1]$, which are respectively named SaDSDE-1, SaDSDE-2 and SaDSDE-3 one by one.
2. To verify the contribution of dynamic adjustment strategy of the exploration ability control factor (λ), SaDSDE variants with self-adaptive F , $\lambda = 0.3$, $\lambda = 0.7$ and random real number in $[0, 1]$ are respectively named SaDSDE-4, SaDSDE-5 and SaDSDE-6 for short.

In order to evaluate and compare the performance of different SaDSDE variants, Friedman test and Wilcoxon's rank-sum test are employed, and the test results are plotted in Fig 12 and Table 12, respectively. Observing the test results in Fig 12 and Table 12, we can obtain the following conclusions easily. (1) From Fig 12, the proposed SaDSDE and SaDSDE-1 are

Table 10. The optimization results obtained by SaDSDE and 3 non-DE algorithms at $D = 100$.

F	SL-PSO	GWO	WOA	SaDSDE
	Mean \pm STD	Mean \pm STD	Mean \pm STD	Mean \pm STD
f_1	4.27E-15 \pm 2.74E-15	3.51E-41 \pm 3.48E-41	3.69E-188 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_2	1.06E+05 \pm 1.42E+04	3.02E-03 \pm 9.79E-03	4.62E+05 \pm 7.13E+04	0.00E+00 \pm 0.00E+00
f_3	6.07E-07 \pm 2.65E-06	8.48E-38 \pm 8.31E-38	2.09E-183 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_4	3.15E-08 \pm 1.88E-08	1.18E-24 \pm 9.13E-25	3.39E-112 \pm 1.58E-111	0.00E+00 \pm 0.00E+00
f_5	2.65E+00 \pm 7.10E-01	2.04E-06 \pm 4.86E-06	2.34E-02 \pm 8.78E-02	0.00E+00 \pm 0.00E+00
f_6	1.82E-15 \pm 1.59E-15	1.38E-41 \pm 1.43E-41	9.20E-191 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_7	4.27E-09 \pm 2.45E-09	6.27E-35 \pm 1.04E-34	2.52E-180 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_8	1.60E+05 \pm 4.07E+05	1.30E-40 \pm 1.53E-40	1.27E+03 \pm 4.18E+02	0.00E+00 \pm 0.00E+00
f_9	7.24E-08 \pm 3.75E-07	6.56E-35 \pm 6.19E-35	4.07E-182 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{10}	2.00E-01 \pm 4.84E-01	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{11}	1.36E+04 \pm 1.05E-05	8.48E-04 \pm 3.77E-04	2.50E-05 \pm 1.77E-05	1.36E+04 \pm 5.13E-06
f_{12}	6.84E+02 \pm 1.67E+02	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{13}	1.81E-03 \pm 4.34E-03	1.03E-03 \pm 3.17E-03	4.58E-03 \pm 1.45E-02	0.00E+00 \pm 0.00E+00
f_{14}	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{15}	1.26E-15 \pm 7.25E-16	4.37E-42 \pm 4.41E-42	9.55E-193 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{16}	1.04E-08 \pm 3.04E-09	4.42E-14 \pm 4.03E-15	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{17}	3.53E+02 \pm 3.71E+02	9.68E+01 \pm 9.76E-01	9.65E+01 \pm 3.29E-01	9.70E+01 \pm 6.77E-01
f_{18}	2.57E+00 \pm 3.98E+00	3.82E-10 \pm 2.18E-10	1.13E-64 \pm 5.78E-64	0.00E+00 \pm 0.00E+00
f_{19}	1.23E+01 \pm 6.74E+01	1.27E-03 \pm 3.69E-12	1.27E-03 \pm 0.00E+00	1.27E-03 \pm 0.00E+00
f_{20}	6.76E-01 \pm 5.70E-02	8.58E-01 \pm 9.08E-02	4.48E-01 \pm 1.23E-01	9.31E-01 \pm 6.87E-02
f_{21}	7.01E-01 \pm 3.18E-01	5.63E-01 \pm 9.11E-02	4.43E-01 \pm 8.63E-02	5.00E-01 \pm 0.00E+00
f_{22}	5.59E-01 \pm 1.05E+00	2.56E-14 \pm 1.56E-14	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{23}	0.00E+00 \pm 0.00E+00	5.05E-01 \pm 7.60E-02	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{24}	2.27E+00 \pm 6.99E-01	1.17E+00 \pm 1.26E+00	6.46E-01 \pm 2.03E+00	0.00E+00 \pm 0.00E+00
f_{25}	6.74E+01 \pm 3.63E+00	4.05E+01 \pm 1.93E+00	0.00E+00 \pm 0.00E+00	4.39E+01 \pm 7.11E-01
f_{26}	8.34E+02 \pm 3.50E+01	2.53E+00 \pm 4.40E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{27}	3.70E-08 \pm 2.22E-08	6.67E-05 \pm 1.80E-04	1.35E-110 \pm 6.85E-110	0.00E+00 \pm 0.00E+00
f_{28}	5.60E-01 \pm 6.60E-01	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00
f_{29}	1.56E-03 \pm 8.57E-03	5.16E-01 \pm 7.38E-02	2.81E-03 \pm 7.52E-04	3.96E-02 \pm 2.27E-02
f_{30}	4.03E-03 \pm 8.89E-03	5.09E+00 \pm 4.57E-01	3.60E-02 \pm 1.69E-02	9.33E+00 \pm 1.73E+00
+	23	19	12	-
-	3	4	7	-
\approx	4	7	11	-

<https://doi.org/10.1371/journal.pone.0222706.t010>

respectively the best and the second best, followed by SaDSDE-4, SaDSDE-3, SaDSDE-2, SaDSDE-6 and SaDSDE-5. It is clear that the combined effect of proposed algorithm components is best. (2) From Table 12, there is no significant performance difference between proposed SaDSDE and SaDSDE variants with exploration ability control factor, while proposed SaDSDE is better than SaDSDE-5 with a larger exploration ability control factor significantly. The validity of two proposed strategies is verified by means of above experimental comparisons. It is note that the contribution of dynamic adjustment strategy of the exploration ability control factor is larger than scaling factor self-adaptation strategy. In other words, although the scaling factor self-adaption is effective, SaDSDE is less susceptible to scaling factor.

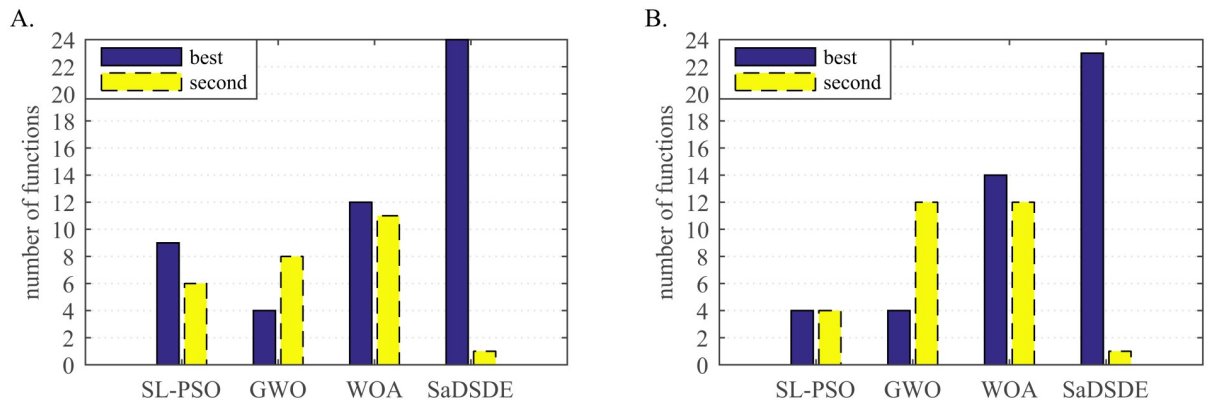


Fig 8. Number of cases on which each algorithm performs the best and second best in the comparison. (A) $D = 30$. (B) $D = 100$.

<https://doi.org/10.1371/journal.pone.0222706.g008>

4.7 Adaptability analysis to high-dimensional problems

In order to evaluate the adaptability to high-dimensional problems of each algorithm, we conduct Wilcoxon’s signed-rank test and Wilcoxon rank-sum test at the 0.05 significance level for each algorithm between its own 30 variables and 100 variables. The nonparametric statistical test results are shown in Fig 13 and Table 13. From Fig 13, we can obtain the following observations: (1) For EsDER-NR, AGDE and EFADE, the low-dimensional performance is significantly better than high-dimensional performance on all the test functions. (2) For SaDSDE, the low dimensional performance is significantly better than, worse than and similar to high dimensional performance on 7 and 23 out of 30 functions, respectively. (3) For IMMSADE, GWO, WOA and SaDSDE, the difference is not statistically significant in Table 13, while the gap between R^+ and R^- of SaDSDE is significantly smaller than IMMSADE, GWO and WOA. Therefore, SaDSDE has the best adaptability to high-dimensional problems.

4.8 Discussions on the comparison results

A series of experiment comparisons have proved the effectiveness of SaDSDE. The reasons can be concluded as follows. (1) The novel dual-strategy mutation operator gives attention to exploitation and exploration ability simultaneously. Exploration can make the algorithm

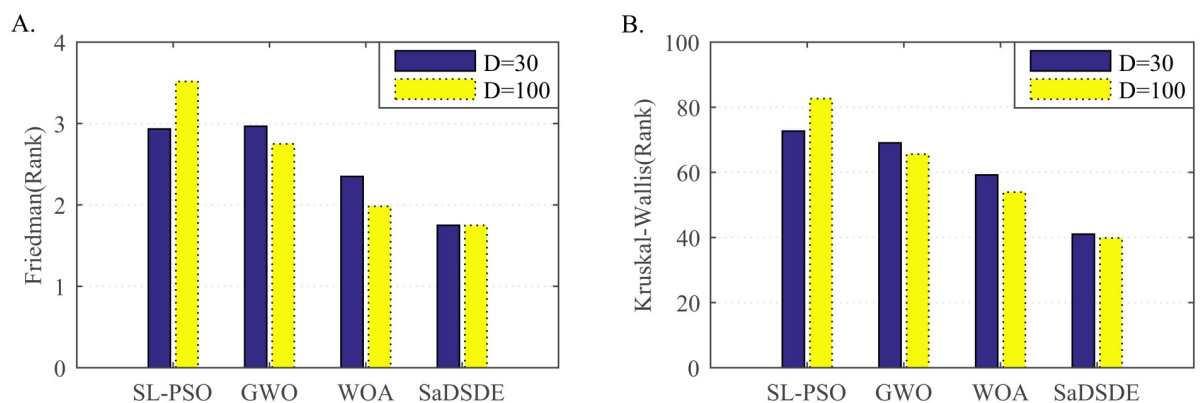


Fig 9. Non-parametric statistical test results over 30 independent runs. (A) Friedman test results. (B) Kruskal-Wallis test results.

<https://doi.org/10.1371/journal.pone.0222706.g009>

Table 11. The results of Wilcoxon rank-sum test over 30 independent runs.

Comparisons	D = 30					D = 100				
	R+	R-	p-value	$\alpha = 0.05$	$\alpha = 0.1$	R+	R-	p-value	$\alpha = 0.05$	$\alpha = 0.1$
SaDSDE versus SL-PSO	218	82	9.56E-04	Yes	Yes	358	48	3.13E-05	Yes	Yes
SaDSDE versus GWO	282	69	1.05E-03	Yes	Yes	239	112	1.79E-03	Yes	Yes
SaDSDE versus WOA	157	74	1.24E-02	Yes	Yes	122	109	2.28E-02	Yes	Yes

<https://doi.org/10.1371/journal.pone.0222706.t011>

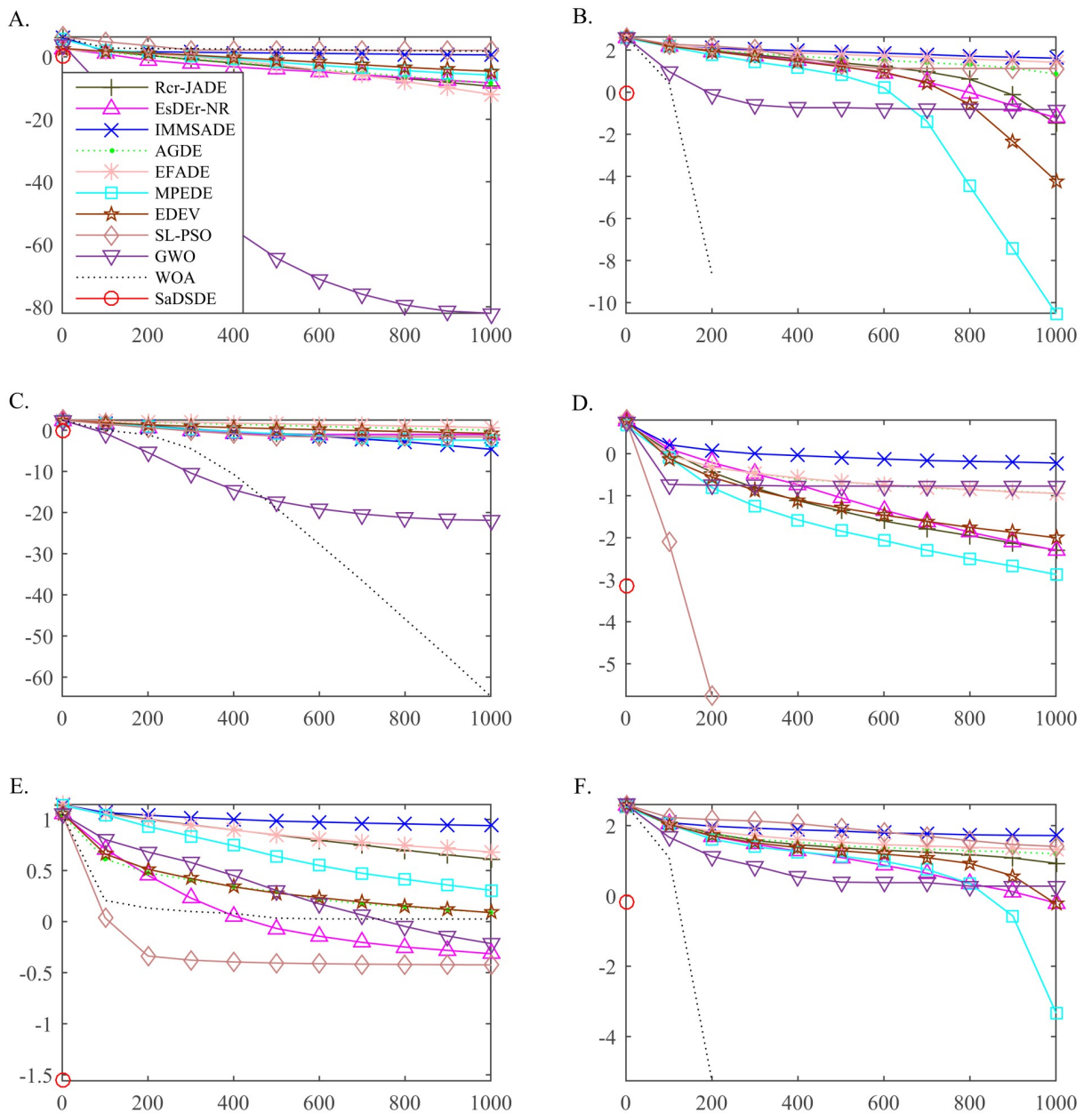


Fig 10. Convergence curves of the mean function error values for six test functions at $D = 30$. The horizontal axis and the vertical axis are generations and the mean function error values over 30 independent runs. The legends of Fig 10 (B-F) are the same as Fig 10(A). (A) f_8 . (B) f_{12} . (C) f_{18} . (D) f_{23} . (E) f_{24} . (F) f_{26} .

<https://doi.org/10.1371/journal.pone.0222706.g010>

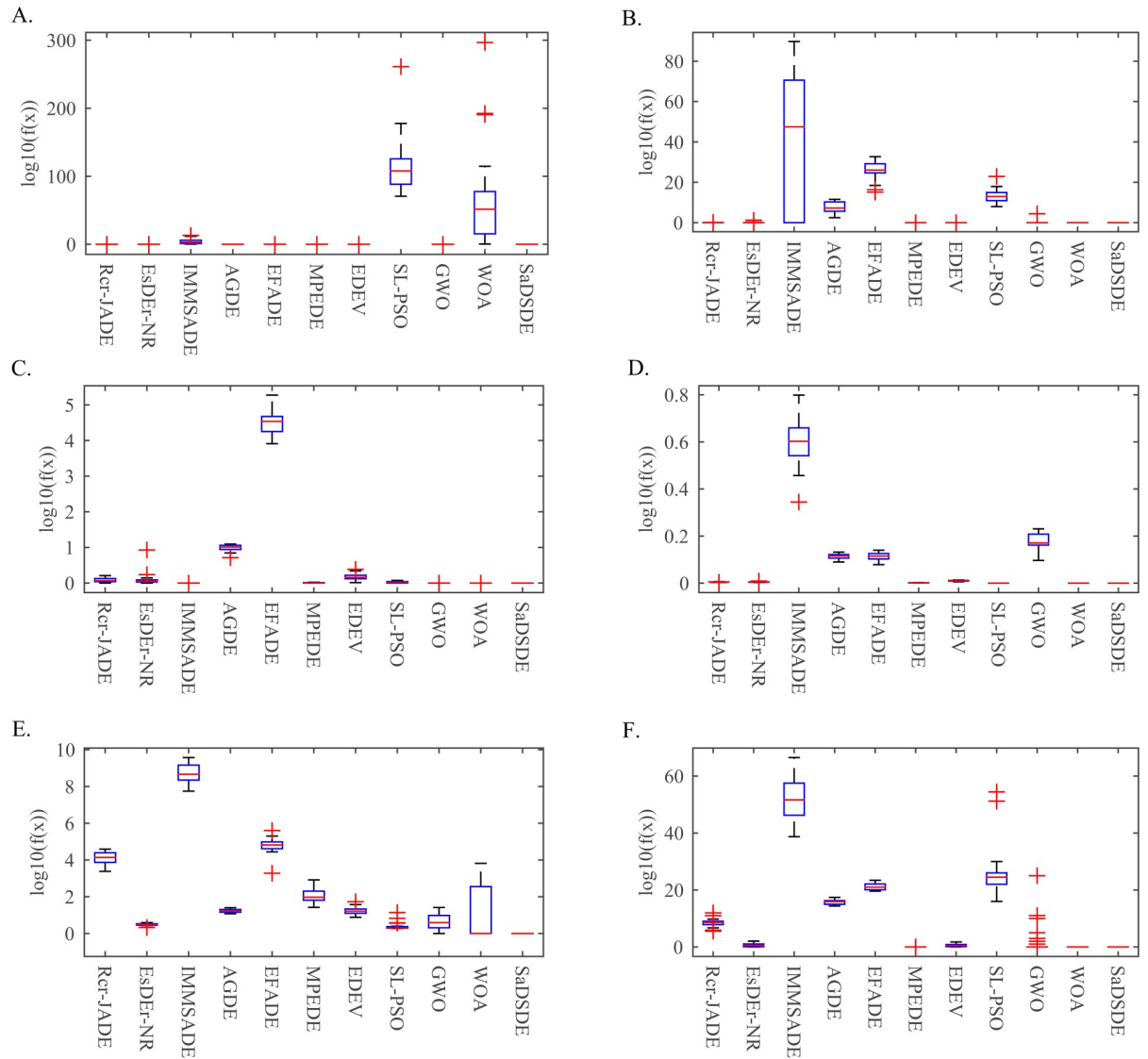


Fig 11. Box plots of the result of solution error at $D = 30$ over independent 30 runs. (A) f_8 . (B) f_{12} . (C) f_{18} . (D) f_{23} . (E) f_{24} . (F) f_{26} .

<https://doi.org/10.1371/journal.pone.0222706.g011>

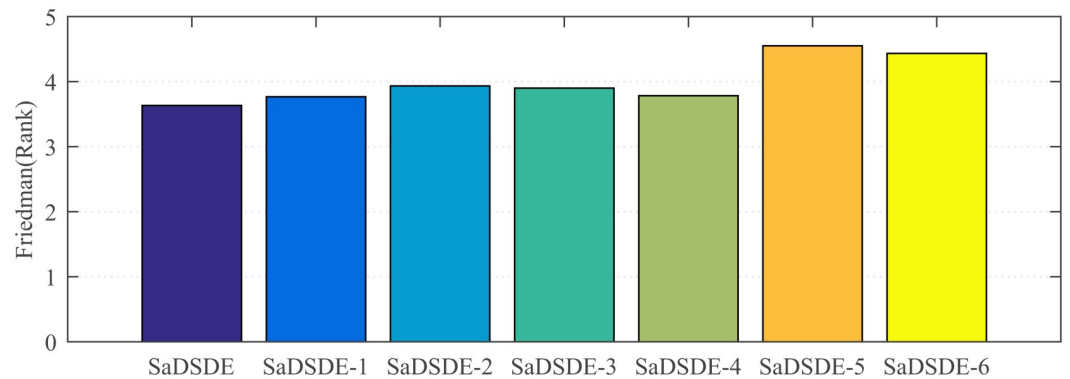


Fig 12. The Friedman test results over 30 independent runs.

<https://doi.org/10.1371/journal.pone.0222706.g012>

Table 12. The results of Wilcoxon’s rank-sum test of proposed SaDSDE and 6 SaDSDE variants over 30 independent runs.

Comparisons	R+	R-	<i>p</i> -value	$\alpha = 0.05$	$\alpha = 0.1$
SaDSDE versus SaDSDE-1	14	7	9.92E-01	No	No
SaDSDE versus SaDSDE-2	16	5	9.02E-01	No	No
SaDSDE versus SaDSDE-3	15	6	9.32E-01	No	No
SaDSDE versus SaDSDE-4	24	12	6.03E-01	No	No
SaDSDE versus SaDSDE-5	114	39	2.85E-02	Yes	Yes
SaDSDE versus SaDSDE-6	55	11	2.75E-01	No	No

<https://doi.org/10.1371/journal.pone.0222706.t012>

search every promising solution area with good diversity, while exploitation can make the algorithm execute a local search in some promising solution areas to find the optimal point with a high convergence rate. (2) In scaling factor self-adaption strategy, the scaling factors of each individual are set for individuals to make full use of the differences in their fitness. Therefore, the superior individuals are assigned relatively smaller values to exploit their neighborhoods in which better solutions. On the contrary, the inferior individuals are assigned larger values to explore further areas in the solution space. (3) The exploration ability control factor

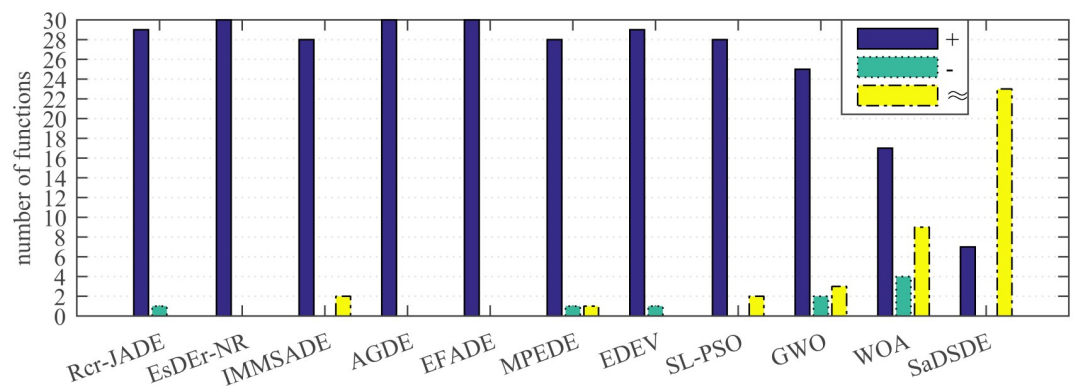


Fig 13. The Wilcoxon’s signed-rank test results with a significance level of 0.05 over 30 independent runs. Signs “+”, “-” and “~” indicate the number of functions that the performance of algorithm for low dimensional problems is better than, worse than and similar to the performance for high dimensional problems, respectively.

<https://doi.org/10.1371/journal.pone.0222706.g013>

Table 13. The results of Wilcoxon rank-sum test over 30 independent runs.

Comparisons	R+	R-	<i>p</i> -value	$\alpha = 0.05$	$\alpha = 0.1$
Rcr-JADE at <i>D</i> = 30 versus Rcr-JADE at <i>D</i> = 100	455	10	7.29E-04	Yes	Yes
EsDEr-NR at <i>D</i> = 30 versus EsDEr-NR at <i>D</i> = 100	465	0	2.88E-06	Yes	Yes
IMMSADE at <i>D</i> = 30 versus IMMSADE at <i>D</i> = 100	406	0	1.19E-01	No	No
AGDE at <i>D</i> = 30 versus AGDE at <i>D</i> = 100	465	0	6.74E-06	Yes	Yes
EFADE at <i>D</i> = 30 versus EFADE at <i>D</i> = 100	465	0	7.60E-07	Yes	Yes
MPEDE at <i>D</i> = 30 versus MPEDE at <i>D</i> = 100	426	9	5.54E-04	Yes	Yes
EDEV at <i>D</i> = 30 versus EDEV at <i>D</i> = 100	454	11	2.83E-04	Yes	Yes
SL-PSO at <i>D</i> = 30 versus SL-PSO at <i>D</i> = 100	406	0	1.27E-02	Yes	Yes
GWO at <i>D</i> = 30 versus GWO at <i>D</i> = 100	345	33	3.87E-01	No	No
WOA at <i>D</i> = 30 versus WOA at <i>D</i> = 100	178	53	9.16E-01	No	No
SaDSDE at <i>D</i> = 30 versus SaDSDE at <i>D</i> = 100	28	0	8.94E-01	No	No

<https://doi.org/10.1371/journal.pone.0222706.t013>

is adjusted dynamically according to the generations, which takes into account the evolutionary property in different evolution stages. It can get a good balance between exploration and exploitation ability to make the algorithm search promising solution area and help algorithm jump over the trap of local optimal solution.

Conclusions

The novel mutation operator with “DE/best/2” operator and “DE/rand/2” operator is proposed in this paper. The “DE/rand/2” mutation operator can expand the search area in the later stage of evolution and avoid suffering premature convergence. The “DE/best/2” mutation operator can accelerate the convergence speed greatly. The performance of the algorithm is excellent even if *CR* is set to a fixed value. Experiment results show that: (1) SaDSDE is not sensitive to the population size and it is easier to implement. (2) SaDSDE has the best performance among all the compared variants. (3) For the high-dimensional functions, with the same-scale population and iterations, SaDSDE can still get the excellent global optimization performance, while there is serious performance degradation for other compared algorithms. In other words, SaDSDE has the best adaptability to high-dimensional problems.

As a continuation of this research, we will focus on multi-objective optimization (MOO) problems and some actual engineering applications, such as flight conflict resolution and image registration, etc.

Author Contributions

Funding acquisition: Hongyu Yang.

Methodology: Meijun Duan.

Project administration: Hongyu Yang.

Software: Meijun Duan.

Validation: Meijun Duan, Shangping Wang, Yu Liu.

Visualization: Meijun Duan.

Writing – original draft: Meijun Duan.

Writing – review & editing: Shangping Wang, Yu Liu.

References

1. Storn R, Price K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. California: University of California, Berkeley. 1995.
2. Zhai S, Jiang T. A new sense-through-foliage target recognition method based on hybrid differential evolution and self-adaptive particle swarm optimization-based support vector machine. *Neuro computing*. 2015; 149:573–584.
3. Bui NT, Hasegawa H. Training Artificial Neural Network Using Modification of Differential Evolution Algorithm. *International Journal of Machine Learning and Computing*. 2015; 5(1):1–6.
4. Arce F, Zamora E, Sossa H, Barrón R. Differential evolution training algorithm for dendrite morphological neural networks. *Applied Soft Computing*. 2018; 68:303–313.
5. Das S, Abraham A, Konar A. Automatic clustering using an improved differential evolution algorithm. *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*. 2008; 38(1): 218–236.
6. El-Quliti SA, Mohamed AW. A large-scale nonlinear mixed binary goal programming model to assess candidate locations for solar energy stations: an improved real-binary differential evolution algorithm with a case study. *J Comput Theor Nanosci*. 2016; 13(11):7909–7921.
7. ĆrepinŠek M, Liu SH, Mernik M. Exploration and exploitation in evolutionary algorithms: a survey. *ACM Computing Surveys*. 2013; 45:1–33.

8. Price K, Storn R, Lampinen J. *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer-VerlagR. 2005.
9. Storn R, Price K. Home Page of Differential Evolution. *Int. Comput. Sci. Inst.*, Berkeley, CA, USA. 2010.
10. Tanabe R, Fukunaga A. Success-History Based Parameter Adaptation for Differential Evolution. *IEEE Congress on Evolutionary Computation (CEC)*. 2013; 71–78.
11. Gong WY, Cai ZH, Wang Y. Repairing the crossover rate in adaptive differential evolution. *Applied Soft Computing*. 2014; 15:149–168.
12. Zhang J, Sanderson AC. JADE: Adaptive differential evolution with optional external archive. *IEEE Trans on Evolutionary Computation*. 2009; 13(5):945–958.
13. Awad NH, Ali MZ, Suganthan PN, Reynolds RG. An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. *Proceedings of the IEEE Congress on Evolutionary Computation*. 2016; 2958–2965.
14. Wu Z, Yang T, Fang JA, Zhang WB. Adaptive population tuning scheme for differential evolution. *Information Sciences*. 2013; 223:164–191
15. Wang X, Zhao SG. Differential Evolution Algorithm with Self-Adaptive Population Resizing Mechanism. *Mathematical Problems in engineering*. 2013; Article ID:419372.
16. Chen L, Zhao S, Zhu W, L YY, Z WB. A self-Adaptive differential evolution algorithm for parameters identification of stochastic genetic regulatory networks with random delays. *Arabian Journal for Science and Engineering*. 2014; 39(2):821–835.
17. Awad NH, Ali MZ, Suganthan PN. Ensemble of parameters in a sinusoidal differential evolution with niching-based population reduction. *Swarm and Evolutionary Computation*. 2018; 39:141–156.
18. Wang SH, Li YZ, Yang HY. Self-adaptive differential evolution algorithm with improved mutation mode. *Applied Intelligence*. 2017; 47:644–658.
19. Cai YQ, Sun G, Wang T, Tian H, Chen YH, Wang JH. Neighborhood-adaptive differential evolution for global numerical optimization. *Applied Soft Computing*. 2017; 59:659–706.
20. Tang RL. Decentralizing and coevolving differential evolution for large-scale global optimization problems. *Applied Intelligence*. 2017; 47:1208–1223.
21. Mohamed AW, Mohamed AK. Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. *International Journal of Machine Learning and Cybernetics*. 2017; 1–23.
22. He XY, Zhou YR. Enhancing the performance of differential evolution with covariance matrix self-adaptation. *Applied Soft Computing*. 2018; 64:227–243.
23. Mohamed AW, Suganthan PN. Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation. *Soft Computing*. 2018; 22(10):3215–3235.
24. Cai YQ, Liao JL, Wang T, Chen YH, Tian H. Social learning differential evolution. *Information Sciences*. 2018; 433–444:464–509.
25. Qin AK, Huang VL, Suganthan PN. Differential evolution algorithm with strategy adaption for global numerical optimization. *IEEE Trans on Evolutionary Computation*. 2009; 13:398–417.
26. Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans on Evolutionary Computation*. 2011; 15(1):55–66.
27. Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*. 2011; 11(2): 1679–1696.
28. Elsayed SM, Sarker RA, Essam DL. A self-adaptive combined strategies algorithm for constrained optimization using differential evolution. *Applied Mathematics and Computation*. 2014; 241:267–282.
29. Wu GH, Mallipeddi R, Suganthan PN, Wang R, Chen HK. Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences*. 2016; 329:329–345.
30. YEH MF, LU HC, CHEN TH, LEU MS. Modified Gaussian barebones differential evolution with hybrid crossover strategy. *Proceedings of the 2016 International Conference on Machine Learning and Cybernetics*. 2017; 7–12.
31. Cui LZ, Li GH, Zhu ZX, Lin QZ, Wong KC, Chen JY, et al. Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism. *Information Sciences*. 2018; 422:122–142.
32. Wu GH, Shen X, Li HF, Chen HK, Lin AP, Suganthan PN. Ensemble of differential evolution variants. *Information Sciences*. 2018; 423:172–186.
33. Lin QZ, Ma YP, Chen JY, Zhu QL, Coello CAC, W KC, et al. An adaptive immune-inspired multi-objective algorithm with multiple differential evolution strategies. *Information Sciences*. 2018; 430–431:46–64.

34. Wang Y, Li HX, Huang T, Li L. Differential evolution based on covarianc matrix learning and bimodal distribution parameter setting. *Appl. Soft Comput.* 2014; 18:232–247.
35. Cai YQ, Wang JH. Differential evolution with hybrid linkage crossover. *Inf. Sci* 2015; 320:244–287.
36. Guo SM, Yang CC. Enhancing Differential Evolution Utilizing Eigenvector-Based Crossover Operator. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.* 2015; 19(1):31–49.
37. Xu YL, Fang JA, Zhu W, Wang XP, Zhao LD. Differential evolution using a superior-inferior crossover scheme. *Comput Optim Appl.* 2015; 61:243–274.
38. Zhu QL, Lin QZ, Du ZH, Liang ZP, Wang WJ, Zhu ZX, et al. A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm. *Information Sciences.* 2016; 345:177–198.
39. Ghosh A, Das S, Mullick SS, Mallipeddi R, Das AK. A switched parameter differential evolution with optional blending crossover for scalable numerical optimization. *Applied Soft Computing.* 2017; 57:329–352.
40. Qiu X, Tan KC, Xu JX. Multiple Exponential Recombination for Differential Evolution. *IEEE TRANSACTIONS ON CYBERNETICS.* 2017; 47(4):995–1005. <https://doi.org/10.1109/TCYB.2016.2536167> PMID: 28113880
41. Li X, Yin M. Hybrid differential evolution with artificial bee colony and its application for design of a reconfigurable antenna array with discrete phase shifters. *Iet Microwaves Antennas & Propagation.* 2012; 6(6):1573–1582.
42. Vaisakh K, Praveena P, Sujatah KN. Differential evolution and bacterial foraging optimization based dynamic economic dispatch with non-smooth fuel cost functions. *Swarm, Evolutionary, and Memetic Computing.* 2013; 583–594.
43. Ponsich A, Coello CAC. A hybrid differential evolution-Tabu search algorithm for the solution of job-shop scheduling problems. *Applied Soft Computing.* 2013; 13(1):462–474.
44. Gu XP, Li Y, Jia JH. Feature selection for transient stability assessment based on kernelized fuzzy rough sets and memetic algorithm. *Electrical Power and Energy Systems.* 2015; 64:664–670.
45. Le LD, Vo D, Nguyen TH, Le AD. A hybrid differential evolution and harmony search for non-convex economic dispatch problems. *IEEE Conference on Power Engineering and Optimization.* 2013; 238–243.
46. Nenavath H, Jatoth RK. Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. *Applied Soft Computing.* 2018; 62:1049–1043.
47. Suganthan PN, Hansen N, Liang J, Deb K. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. 2005.
48. Liang JJ, Qu BY, Suganthan PN, Chen Q. Problem definition and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization. 2015.
49. Awad NH, Ali MZ, Suganthan PN, Liang JJ, Qu BY. Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. 2016.
50. Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evolutionary Computation.* 2011; 1:3–18.
51. Cheng R, Jin YC. A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences.* 2015; 291: 43–60.
52. Mirjalili S, Mirjalili SM, Lewis A. Grey Wolf Optimizer. *Advances in Engineering Software.* 2014; 69:46–61.
53. Mirjalili S, Lewis A. The Whale Optimization Algorithm. *Advances in Engineering Software.* 2016; 91:51–67.