scientific reports

OPEN

A novel hybrid adaptive differential evolution for global optimization

Zhiyong Zhang[1,2], Jianyong Zhu[1]✉ & Feiping Nie[3]

Differential Evolution (DE) stands as a potent global optimization algorithm, renowned for its application in addressing a myriad of practical engineering issues. The efficacy of DE is profoundly influenced by its control parameters and mutation strategies. In light of this, we introduce a refined DE algorithm characterized by adaptive parameters and dual mutation strategies (APDSDE). APDSDE inaugurates an adaptive switching mechanism that alternates between two innovative mutation strategies: DE/current-to-*p*Best-w/1 and DE/current-to-Amean-w/1. Furthermore, a novel parameter adaptation technique rooted in cosine similarity is established, with the derivation of explicit calculation formulas for both the scaling factor weight and crossover rate weight. In pursuit of optimizing convergence speed whilst preserving population diversity, a sophisticated nonlinear population size reduction method is proposed. The robustness of each algorithm is rigorously evaluated against the CEC2017 benchmark functions, with empirical evidence underscoring the superior performance of APDSDE in comparison to a host of advanced DE variants.

Keywords Differential evolution, Cosine similarity, Adaptive parameters, Dual mutation strategies, Population size

Many scientific and engineering problems are often transformed into optimization challenges by formulating appropriate objective functions. Population-based optimization algorithms have been proved to be effective in solving these problems. Over the past decade, researchers have developed numerous new population-based optimization algorithms, categorized into five types: physical-based algorithms[1,2], swarm intelligence algorithms[3,4], human-based algorithms[5], math-based algorithms[6], and evolutionary algorithms[7,8]. These methods have broad applications across various fields. For instance, physics-based optimization algorithms are combined with chaotic mapping for model identification[9,10]. The grey wolf optimizer, as a swarm intelligence algorithm, is improved for parameter estimation of the autoregressive exogenous model[11]. The particle swarm optimization, a type of swarm intelligence algorithm, is widely utilized for addressing complex optimization problems like target searching[12,13] and developing recommender systems[14]. A math-based optimization algorithm, Runge-Kutta optimizer, is used to identify the model parameters of fractional input nonlinear autoregressive exogenous system[15].

Differential Evolution (DE) is a prominent strategy in evolutionary algorithms, acclaimed for its effectiveness in navigating complex optimization landscapes. Originally proposed in 1997[16], DE has garnered widespread application across diverse domains, including ship-unloading scheduling optimization[17], large-scale feature selection[18], fluid catalytic cracking[19], microwave circuit designs[20].

The potency of DE is contingent upon the optimal tuning of parameter settings and the selection of mutation operators. Tailoring these elements to the unique requisites of specific optimization problems is essential, given that traditional methods, often grounded in manual expertise or iterative trial and error, prove insufficiently adaptive. Inadequately tuned parameters and ill-suited mutation strategies often culminate in premature convergence, precluding the attainment of global optima. Consequently, there is a burgeoning interest in the research community towards the development of adaptive parameters and mutation strategies, as evidenced by studies[21,22].

The population size is intrinsic to DE's capability, where a suitable size amplifies diversity and hastens convergence. A plethora of adaptive methods tailored for population size optimization have been delineated in the literature[23–28]. In this paper, a new nonlinear population size adaptive method is designed. In addition, ideal results have been obtained by adaptively adjusting the control parameters according to the difference between the target individual and testing vector in DISH[29] and SLDE[23]. Inspired by these two methods, we propose a modified adaptive control parameter method.

[1]School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang 330013, China. [2]Jiangxi Technical College of Manufacturing, Nanchang 330013, China. [3]School of Artifcial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China. ✉email: zhujyemail@163.com

The dual faculties of exploration and exploitation in DE are intimately tethered to the selection of the mutation strategy. Certain mutation strategies exhibit prowess in exploration but are found wanting in exploitation, and vice versa. The challenge of identifying an apt mutation strategy that caters to diverse optimization conundrums and varying evolutionary phases is non-trivial. A solitary mutation operation often grapples with balancing between exploration and exploitation. In response, many refined mutation strategies have been introduced by specialists and scholars over the years[18,24,25,30]. These strategies range from enhancements of individual mutation strategies to integrations of multiple mutations to amplify DE's operational capacity. This paper integrates two mutation operations to enhance the capacity of DE.

In this context, our work advances an enriched DE algorithm, anchored in adaptive parameters and a dual mutation strategy ensemble. The contributions delineated herein are multifaceted.

- A dual mutation strategy is introduced, characterized by two novel variants, each enhancing existing mutation operators. The first variant refines the computation methodology of the factor F_w, while the second incorporates the weight factor F_w. The integration of these strategies is facilitated by an adaptive switching mechanism, designed to maximize their combined effectiveness.
- A new calculation formula of the weights for F and CR is adopted. The calculation formula adopts the cosine similarity between the parent and trial vectors instead of the Euclidean distance in DISH and the squared Euclidean distance in SLDE.
- A new adaptive population size adjustment method is proposed. In this method, the population size decreases nonlinearly as the number of iterations increases.

Differential evolution and related work
Classical DE
DE includes four basic evolutionary processes. In the beginning, DE produces an initial population $X_i^G = (x_{i,1}^G, \cdots, x_{i,D}^G), i = 1, \cdots, NP$ randomly as follows:

$$x_{i,j}^G = x_{min,j} + rand(0,1) \cdot (x_{max,j} - x_{min,j}), j = 1, 2, \ldots, D \tag{1}$$

where G is the number of generations.

After that, a mutant vector $V_i^G = (v_{i,1}^G, \cdots, v_{i,D}^G)$ is generated for each individual X_i^G by differential mutation strategy. Here are six mutation operators that are frequently used.

$$DE/rand/1: V_i^G = X_{r1}^G + F \cdot (X_{r2}^G - X_{r3}^G), \tag{2}$$

$$DE/rand/2: V_i^G = X_{r1}^G + F \cdot (X_{r2}^G - X_{r3}^G) + F \cdot (X_{r4}^G - X_{r5}^G), \tag{3}$$

$$DE/best/1: V_i^G = X_{best}^G + F \cdot (X_{r1}^G - X_{r2}^G), \tag{4}$$

$$DE/best/2: V_i^G = X_{best}^G + F \cdot (X_{r1}^G - X_{r2}^G) + F \cdot (X_{r3}^G - X_{r4}^G), \tag{5}$$

$$DE/current-to-best/1: V_i^G = X_i^G + F \cdot (X_{best}^G - X_i^G) + F \cdot (X_{r1}^G - X_{r2}^G), \tag{6}$$

$$DE/rand-to-best/1: V_i^G = X_{r1}^G + F \cdot (X_{best}^G - X_{r1}^G) + F \cdot (X_{r2}^G - X_{r3}^G), \tag{7}$$

The indices $r1$ to $r5$ are randomly generated mutually exclusive integers between 1 and NP.

Then, the DE algorithm generates a trial vector $\mathrm{U}_\mathrm{i}^\mathrm{G} = (\mathrm{u}_{\mathrm{i},1}^\mathrm{G}, \cdots, \mathrm{u}_{\mathrm{i,D}}^\mathrm{G})$ from X_i^G and V_i^G. The binomial crossover is a commonly used strategy, and its calculation formula is shown in Eq. (8).

$$u_{i,j}^G = \begin{cases} v_{i,j}^G & if\ rand_j(0,1) \le CR\ or\ j = j_{rand}, \\ x_{i,j}^G & otherwise, \end{cases} \tag{8}$$

where $j_{rand} \in [1, D]$ is a random integer.

The feasible region of $u_{i,j}^G$ must satisfy Eq. (9).

$$u_{i,j}^G = \begin{cases} min\{U_j, 2L_j - u_{i,j}^G\} & if\ u_{i,j}^G < L_j, \\ max\{L_j, 2U_j - u_{i,j}^G\} & if\ u_{i,j}^G > U_j, \end{cases} \tag{9}$$

After that, the better one between X_i^G and U_i^G is saved for the next generation. Hence, the selection operator is shown in Eq. (10).

$$X_i^{G+1} = \begin{cases} X_i^G & if\ f(X_i^G) < f(\mathrm{U}_\mathrm{i}^\mathrm{G}), \\ \mathrm{U}_\mathrm{i}^\mathrm{G} & \text{otherwise}, \end{cases} \tag{10}$$

where $f(x)$ represents the fitness value of the individual x.

Existing related work

Over recent years, there has been a proliferation of enhanced differential evolution algorithms aimed at amplifying the global search proficiency of DE. It is well known that mutation strategy and parameters are two important aspects affecting the differential evolution algorithm. Population size directly affects population diversity and convergence speed. Therefore, our review mainly discusses the above aspects.

Improvement of mutation strategy

In addition to the six most classical mutation operators described in above, various mutation operators have been proposed to further improve DE. Zhang et al.[31] proposed JADE that adopted a new mutation operator with optional archives and adaptive update mechanism of control parameters. Zheng et al.[32] adopted a new mutation strategy that linearly combines the best m individuals with randomly chosen vectors to obtain a new individual. Wang et al.[33] designed a novel mutation operator "DE/current-to-*lbest* /1". Brest et al.[24] introduced a novel weighted mutation strategy that is the variant of iL-SHADE algorithm. Li et al.[34] proposed a novel DE algorithm based on leader-adjoint populations, in which the different populations adopt different mutation operators.

It is obvious that a single mutation operator is difficult to adapt to different optimization problems and different optimization stages. Thus, combining different strategies is a promising approach. In order to improve the adaptive ability, Qin et al.[35] constructed a candidate pool consisting of two common mutation strategies. Wang et al.[36] utilized three mutation strategies to build a strategy pool in which each individual selects the best strategy to generate mutation vector. Li et al.[37] proposed a dual mutation strategies collaboration method to ensure the global exploration capabilities without reducing the local exploitation capabilities. Xia et al.[25] adopted three popular breeding strategies which selected by fitness values of different individuals adaptively. It has been observed that multiple mutation strategies collaboration is a feasible and promising algorithm.

Improvement of control parameters

In fact, different control parameter settings of DE are required for different optimization problems. To find the appropriate control parameter values in time, many scholars have proposed a variety of parameter adaptive adjustment strategies in recent years. Zhang et al.[31] presented a differential evolution algorithm named JADE, which utilized normal distribution and Cauchy distribution to generate CR and F, respectively. In SaDE[35], two normal distributions were used to generate CR and F adaptively. In SHADE[38], a new algorithm for generating control parameters was proposed based on the historical values of successful control parameters. Mohamed et al.[39] proposed an adaptive DE (LSHADE-SPACMA), in which the control parameter values were obtained by semi-parameter adaption method. Brest et al.[40] proposed an iL-SHADE algorithm, which uses a memory update mechanism to improve algorithm performance. In TPDE[41], three different functions were used to generate control parameters for the three subpopulations, and the values of CR and F were adjusted adaptively during the searching process. Viktorin et al.[29] improved the control parameters adaptation method in SHADE by using the Euclidean distance to calculate weights. Based on the control parameter calculation formula of DISH, Zeng et al.[23] used the squared Euclidean distance to update the weight calculation formula for CR and F.

Population size

In the early evolutionary process, a large population is conducive to population diversity, while a small population in the late stage is conducive to accelerated convergence. Thus, many adaptive population size adjustment methods have been proposed. Tanabe et al.[28] designed a linear population size reduction strategy, which has been widely used. Mohamed et al.[27] proposed a population size adjustment method that uses nonlinear functions to gradually reduce population size during evolution. In reference[26], an adaptive population size adjustment method was proposed. This method can not only reduce population size but also increase population size during evolution. Xia et al.[25] proposed a population size adaptive method based on the performance of the optimal individual. Zeng et al.[23] introduced a population size adaptive method based on sawtooth function.

The proposed algorithm

This chapter introduces a new adaptive DE algorithm called APDSDE. APDSDE is improved in three parts. Firstly, we propose a novel weighted mutation operator combining DE/current-to-*p*Best-w/1 and DE/current-to-Amean-w/1, which adopts a novel adaptive scaling factor F_w and selection probability parameter SP_G for two mutation strategies. Secondly, the calculation method of weight parameter w_m for control parameters is improved. Finally, a nonlinear population size reduction method is proposed.

Adaptive mutation strategy

DE/current-to-pBest-w/1

The DE/current-to-*p*Best-w/1 is first proposed in jSO[24], which is shown in Eq. (11). It is an improved mutation strategy that uses the magnitude of the p-value to adjust the greediness of "DE/current-to-best/1" and sets different weight parameters F_w at different stages. Based on this strategy, a new adaptive scaling factor F_w was adopted to improve the mutation strategy. F_w can be calculated according to Eq. (12).

$$V_i^G = X_i^G + F_w \cdot (X_{pbest}^G - X_i^G) + F \cdot (X_{r1}^G - X_{r2}^G), \tag{11}$$

$$F_w = (0.7 + FEs \cdot \frac{a - 0.7}{MaxFEs}) \cdot F, \tag{12}$$

where X_{pbest}^{G} is randomly chosen from the $NP \times p(p \in [0,1])$ vector, which has better fitness values in the G generation population. X_{r1}^{G} is chosen from the population of generation G, and X_{r2}^{G} is chosen from the population that combined the G generation population with the external archive.

DE/current-to-Amean-w/1
DE/current-to-Amean/1[42] is an improved operator, which estimates the global optimal solution X_{Amean}^{G} from the dominant individual in external archive A. Therefore, it can avoid the problem that X_{pbest} is trapped in local optimal value as a result of the decrease of population number in the late evolutionary process. In this study, a novel weighted DE/current-to-Amean-w/1 is proposed, and its expression is shown in Eq. (15). The mutation strategy multiplies the difference of vectors X_{Amean}^{G} and X_{i}^{G} by a scaling factor F_w, which is smaller at the beginning of the evolutionary process and becomes larger as evolution progresses. As the value of F_w increases, the influence of X_{Amean}^{G} gradually increases.

$$V_i^G = X_i^G + F_w \cdot (X_{Amean}^G - X_i^G) + F \cdot (X_{r1}^G - X_{r2}^G), \tag{13}$$

$$X_{Amean}^G = \sum_{i=1}^{m} w_i \cdot X_i^G, \tag{14}$$

$$w_i = \frac{\ln(m+1/2) - \ln(i)}{\sum_{i=1}^{m}(\ln(m+1/2) - \ln(i))}, \tag{15}$$

$$m = round(e \cdot |A|), \tag{16}$$

where $e \in (0,1)$ is the scale coefficient in the range 0 to 1. $|A|$ is the population size of external archive A.

Adaptive switching strategy
At different stages of evolution, there are different requirements for mutation strategies. To take full advantage of the proposed mutation operations, a novel strategy selection mechanism is proposed. In this study, an adaptive selection probability parameter SP_G is used to choose Eqs. (11) or Eq. (13). For each individual in each generation, if a random number between 0 and 1 is less than SP_G, DE/current-to-pBest-w/1 is selected, otherwise DE/current-to-Amean-w/1 is selected. SP is obtained according to the following formula.

$$SP_G = \frac{1}{1 + e^{1-(FEs/MaxFEs)^2}}, \tag{17}$$

where SP_G is an adaptive selection probability parameter. FEs and $MaxFEs$ are the current and maximum number of the evolutions of the objective function, respectively.

Control parameter adaptation strategy
The parameters CR and F are known to have large influence on DE. Therefore, we adopt the parameter calculation method in SLDE to update parameters[23], and proposes a new weight update formula for the control parameters. In SLDE, the crossover probability CR_i of each individual X_i^G is derived from a normal distribution. Conversely, the mutation factor F_i for each individual X_i^G originates from a Cauchy distribution. j is the memory index.

$$CR_i = randn(M_{CR,j}, 0.1), \tag{18}$$

$$F_i = randc(M_{F,j}, 0.1), \tag{19}$$

where M_{CR} and M_F are calculated by the following formulas.

$$M_{CR,n} = \begin{cases} \dfrac{\sum_{m=1}^{|S_{CR}|} w_m \times S_{CR,m}^2}{\sum_{m=1}^{|S_{CR}|} w_m \times S_{CR,m}} & S_{CR} \neq \varnothing, \\ M_{CR,n} & otherwise, \end{cases} \tag{20}$$

$$M_{F,n} = \begin{cases} \dfrac{\sum_{m=1}^{|S_F|} w_m \times S_{F,m}^2}{\sum_{m=1}^{|S_F|} w_m \times S_{F,m}} & S_F \neq \varnothing, \\ M_{F,n} & otherwise, \end{cases} \tag{21}$$

where S_{CR} and S_F are the set of CR_i and F_i which succeed in generating a trial vector, respectively.

To improve the algorithm's optimization capacity, the calculation formula of parameter w_m in F and CR is improved in this paper. For higher dimensional problems, cosine similarity is often used to solve Euclidean distance problems. Therefore, the weight w_m is calculated based on the cosine similarity between the parent and trial vectors instead of the squared Euclidean distance. The weight coefficient w_m can be updated as follows:

$$w_m = \frac{\sum_{j=1}^{D}(x_{m,j}^{G} \times u_{m,j}^{G})}{\sum_{k=1}^{|S_{CR}|}(\sqrt{\sum_{j=1}^{D}(x_{k,j}^{G})^2} \times \sqrt{\sum_{j=1}^{D}(u_{k,j}^{G})^2})}, \tag{22}$$

Adaptive population size adjustment

In the initial iteration phase, a large population size can enhance the population variety, while in the late iteration phase, a small population size can accelerate the convergence rate. Therefore, a new nonlinear population size adjustment method is proposed. The G+1 generation population size is determined according to Eq. (23).

$$NP^{G+1} = round\left(NP^{init} - (NP^{init} - NP^{end}) \times \left(\frac{FEs}{MaxFEs}\right)^{1-(\frac{FEs}{MaxFEs})^2}\right), \tag{23}$$

To sum up, the pseudo code for APDSDE is introduced in Algorithm 1.

Require: $NP^1 = NP^{init}$, the external archive $A = \varnothing$, G=1; All values of M_F and M_{CR} are set to 0.5; Generate initialize population X_i^G using Eq. (1)

while $FEs < MaxFEs$ **do**
    $S_F = \varnothing$, $S_{CR} = \varnothing$
    for i=1 to NP_G **do**
        Randomly select j from [1,H]
        if $M_{CR,j} < 0$ **then**
            $CR_j^G = 0$
        else
            Calculate control parameter CR_i^G according to Eq. (18)
        end if
        CR_i^G=max(CR_i^G,0)
        CR_i^G=min(CR_i^G,1)
        Calculate control parameter F_i^G according to Eq. (19)
        Calculate control parameter F_w according to Eq. (12)
        Update selection probability parameter SP_G using Eq. (17)
        if $SP_G > rand$ **then**
            Calculate the mutation vector V_i^G using Eq. (11)
        else
            Calculate the mutation vector V_i^G using Eq. (13)
        end if
        Calculate the trial vector U_i^G using Eq. (8)
    end for
    if $f(\mathrm{U}_i^G) < f(X_i^G)$ **then**
        $CR_i^G \rightarrow S_{CR}$, $F_i^G \rightarrow S_F$
        if $|A| < 2.6 \cdot NP^G$ **then**
            $X_i^G \rightarrow A$
        else
            Randomly replace a vector in A with X_i^G
        end if
        $X_i^{G+1} = \mathrm{U}_i^G$
    else
        X_i^{G+1}=X_i^G
    end if
    Update M_{CR} and M_F according to Eq. (20) and Eq. (21)
    Calculate NP^{G+1} according to Eq. (23)
    Update A and $|A|$
    $G = G + 1$
end while

Algorithm 1. Pseudo code of APDSDE.

Experimental analysis

Experimental introduction

In the simulation experiment, the CEC2017 test suite was selected to verify the performance of the APDSDE presented in this paper. There are 30 test functions in CEC2017 test suite. For these 30 test functions, the value range of individuals is limited to [−100100].

In this experiment, the maximum function evaluations (MaxFEs) for each run of the D-dimensional test function is 10000 $\times D$. In order to acquire scientific results, each method independently solves 51 times for each test function. The mean (Mean) and standard deviation (Std) of the difference between $f(x_{gbest})$ and $f(x^{'})$ obtained from 51 separate runs are utilized to assess the capacity of each method. $f(x_{gbest})$ is the optimal value and $f(x^{'})$ is the standard value.

The setting of experimental parameters

To assess the capacity of APDSDE, we select six advanced methods for comparative experiments. The seven comparison algorithms are LSHADE-SPACMA[39], DISH[29], FADE[25], MadDE[43], SLDE[23], AL-SHADE[42] and MIDE[44]. LSHADE-SPACMA adopts a semi-parameter adaptive method considering randomness and adaptability to update F. In terms of mutation strategy, DISH and SLDE both use the weighted mutation operator DE/current-to-*p*Best-w/1. But they propose different calculation formulas for control parameters and population size. In DISH, a weight calculation formula for F and CR based on Euclidean distance is introduced. SLDE improves the weight calculation formula in DISH by using the squared Euclidean distance instead of Euclidean distance, in which sawtooth-linear population size adaptive method is adopted to calculate the population size. In FADE, three different mutation operators are used for elite individuals, inferior individuals and medium individuals. At the same time, population size is updated based on fitness landscape. MadDE is an improved DE algorithm, in which the values of F and CR are set by the multiple adaptive methods. In AL-SHADE, two mutation operators are selected according to an adaptive selection mechanism. In MIDE, migration mechanism and information reutilization are used to improve the algorithm. Table 1 lists the parameter settings for APDSDE and other algorithms.

Strategy effectiveness analysis

In this section, we evaluate the effectiveness of the improvement strategies for enhancing the performance of the APDSDE algorithm. To study the effect of each improvement strategy on algorithm performance, three DE variants are designed, named APDSDE-1, APDSDE-2 and APDSDE-3. APDSDE-1 uses adaptive dual mutation strategy to enhance the original mutation strategy of DE. APDSDE-2 uses the parameters generated by the proposed control parameter adaptive strategy to replace the original control parameters of DE algorithm. APDSDE-3 generates the population size using the proposed adaptive population size adjustment strategy. Therefore, the effectiveness of the proposed adaptive dual mutation strategy can be evaluated by comparing APDSDE-1 and DE. The influence of the proposed control parameter adaptive strategy on the algorithm can be analyzed by comparing APDSDE-2 and DE. The effectiveness of the proposed adaptive population size adjustment strategy can be evaluated by comparing APDSDE-3 and DE. The three DE variants use the same parameter settings as the APDSDE algorithm.

Wilcoxon rank sum test with a 0.05 significance level is utilized to check the differences between DE and its three variants for each function. Table 2 summarizes the statistical analysis results between DE and its three variants on the three dimensions of the CEC2017 test suite based on the Wilcoxon's test. In Table 2, the symbols "+", "−" and "≈" represent that the DE variant is superior to, inferior to, and similar to the DE, respectively. The numbers in the table indicate the number of functions where the DE variant is superior to, poorer to, and like to the DE.

Table 2 shows that the number of "+" obtained by the three DE variants is much larger than the number of "-" compared to DE on the 10-dimensional functions. For 30 and 50 dimensional functions, the three variants of DE outperform DE on all functions. The values of "Total" in Table 2 show the overall performance of the three algorithms on CEC2017. APDSDE-1, APDSDE-2, and APDSDE-3 have larger number of "+" than "-" compared to DE. Furthermore, we introduce Friedman test to evaluate the effect of the proposed strategies. Table 3 summarizes the calculation results of rankings of APDSDE-1, APDSDE-2, APDSDE-3 and DE in CEC 2017 test suite derived from the Friedman test with significance level $\alpha = 0.05$. For 10D, 30D, and 50D in CEC2017 test suite, APDSDE-1, APDSDE-2, and APDSDE-3 all obtain better ranking than DE. "Mean Ranking" represents

Algorithm	Year	Parameter values
APDSDE	–	NP^{init}=$18 \times D$, NP^{end}=4, a=1.4, e=0.5, $\|A\|$=$2.6 \cdot NP$, p=0.11
LSHADE-SPACMA[39]	2017	NP^{init}=$18 \times D$, NP^{end}=4, $Pbest$=0.11, H=5, Arc_rate=1.4, F_{CP}=0.5, c=0.8, SPA=$FEs_{max}/2$
DISH[29]	2019	NP^{init}=$25 \times log(D)\sqrt{D}$, NP^{end}=4, p_{max}=0.25, M_F=0.5, M_{CR}=0.8, p_{min}=$p_{max}/2$, H=6
FADE[25]	2021	NP^{init}=$ns \times ss$, NP^{end}=$NP^{init}/3$, ss=3, ns=25 *or* 40, $MaxG_{imp}$=$MaxG_{stag}$=2
MadDE[43]	2021	NP^{init}=2, NP^{end}=4, ρ_{qBX}=0.01, A_{rate}=2.30, H_m=10, F_0=0.2, CR_0=0.2
SLDE[23]	2022	NP^{init}=$14 \times D$+$14 \times log(D)$+4, NP^{end}=4, M_F=0.5, M_{CR}=0.8, p_{max}=0.25, p_{min}=$p_{max}/2$, H=6, α=0.95
AL-SHADE[42]	2022	NP^{init}=$18 \times D$, NP^{end}=4, $\|A\|$=$2.6 \cdot NP$, p=0.11, H=6, e=0.5
MIDE[44]	2024	NP=50, CR=0.9, $F \in [0, 1]$, $T = 0.3 \times D$

Table 1. Parameter values of all algorithms.

Algorithm	10D			30D			50D			Total		
	+	−	≈	+	−	≈	+	−	≈	+	−	≈
APDSDE-1 vs DE	23	6	1	30	0	0	30	0	0	83	6	1
APDSDE-2 vs DE	23	7	0	30	0	0	30	0	0	83	7	0
APDSDE-3 vs DE	29	0	1	30	0	0	30	0	0	89	0	1

Table 2. Comparison results of improved DE based on different strategies on CEC2017.

Algorithm	10D ranking	30D ranking	50D ranking	Mean ranking
APDSDE-1	2.85	2.43	2.24	2.51
APDSDE-2	1.83	1.57	1.76	1.72
APDSDE-3	1.82	2.00	2.00	1.94
DE	3.50	4.00	4.00	3.83

Table 3. The rankings of improved DE based on different strategies on CEC2017.

the mean of the Friedman ranking value of the algorithm for each dimension. According to the "Mean Ranking", APDSDE-1, APDSDE-2, and APDSDE-3 are superior to DE. In summary, the results of statistical analysis show that the adaptive dual mutation strategy, the control parameter adaptive strategy, and the adaptive population size adjustment strategy have positive effects on the performance improvement of DE.

Experimental results on CEC2017 test suite

The results of 30 functions in CEC2017 test suite solved by each algorithm are shown in Tables 4, 5 and 6 for 10D, 30D, and 50D, respectively. The Mean and Std obtained by each algorithm are listed in the three tables, where the boldface is the best solution for each test function. In this article, on the basis of Wilcoxon's test, the symbols "+", "−" and "≈" are utilized to represent that the current method is superior to, inferior to, and similar to the APDSDE respectively.

The results of the test functions with 10-dimensional variables are delineated in Table 4. From this we can see that APDSDE can obtain the global best solution on 9 of the 30 functions. LSHADE-SPACMA, DISH, SLDE and AL-SHADE can obtain the global optimal solutions of 7 functions. MIDE can find the optimal solutions of 6 functions, and FADE and MadDE can find 2 functions. All algorithms can obtain the global optimum on unimodal functions, except FADE and MadDE. For F1–F6, F9–F11, F13–F14, F16, F19–F20, F22 and F30, APDSDE is superior to LSHADE-SPACMA, DISH, FADE, MadDE, SLDE, AL-SHADE and MIDE according to Wilcoxon rank-sum test results. On F7–F8, F12, F15, F17–F18, F21 and F23–F29, although APDSDE can not achieve better performance than all the comparison algorithms, it can achieve better performance than most of them. Among them, only LSHADE-SPACMA, MadDE, AL-SHADE and MIDE are better than APDSDE in a few functions. To be specific, APDSDE is worse than LSHADE-SPACMA on F7, F17 and F29, MadDE on F17, F24, F25 and F27, AL-SHADE on F17 and F29, and MIDE on F18 and F29.

For 30D problems, the optimal solutions attained by various methods are delineated in Table 5. It can be seen that APDSDE and LSHADE-SPACMA can acquire the best value on F1, F2, F3, F6 and F9, and DISH, SLDE and AL-SHADE can obtain the global optimal solution on F1, F2, F3 and F9. It is a great pity that FADE fails to obtain the global optimal solution of 30 functions. All algorithms except FADE, MadDE and MIDE can obtain the global optimal solution on unimodal functions. For F5, F10, F13–17, F19–20, F23–F24 and F29, the optimal solution obtained by APDSDE outperforms the other six algorithms after 51 runs. In addition, the performance of APDSDE is worse than LSHADE-SPACMA on F4, F7, F12, F22, F25 and F28, worse than DISH on F12, F18, F25 and F27, worse than FADE on F4, worse than MadDE on F26, worse than SLDE on F4, F12, F18, F25, F27, F28 and F30, worse than AL-SHADE on F4, and worse than MIDE on F20 and F27.

For 50D problems, the results are presented in Table 6. From this we can see that APDSDE and AL-SHADE can obtain the global optimal solution on F1, F2, F3 and F9, LSHADE-SPACMA on F1, F2, F3, F6 and F9, and DISH and SLDE on F1, F2 and F9. For F4, F10, F16-17, F20-F24 and F29, APDSDE has better performance than LSHADE-SPACMA, DISH, FADE, MadDE, SLDE, AL-SHADE and MIDE. Among the comparison algorithms, only LSHADE-SPACMA, DISH, MadDE, SLDE, AL-SHADE, and MIDE can outperform the proposed APDSDE method in a few functions. Specifically, APDSDE shows worse performance than LSHADE-SPACMA on F6–8, F11–13, F15, F18–19, F25 and F28, DISH on F6, F11–12, F14–15, F18–19, F25, F27–28 and F30, SLDE on F6, F11, F12, F15, F18, F19, F25, F27, F28 and F30, MadDE on F26, AL-SHADE on F19, and MIDE on F11, F15, F19 and F27.

Figure 1 shows the number of functions for which APDSDE and comparison algorithms can obtain the best solutions. For 10D, the optimal value obtained by APDSDE outperforms that obtained by other algorithms on 16 out of 30 functions. Among the comparison algorithms, LSHADE-SPACMA DISH, FADE, MadDE, SLDE, AL-SHADE and MIDE only obtain 12, 7, 3, 6, 8, 10 and 7 best solutions, respectively. For 30D, the number of best solutions obtained by APDSDE is much larger than that of the comparison algorithm. The optimal solution obtained by APDSDE is superior to that obtained by other algorithms on 17 out of 30 functions. However, the comparison algorithm LSHADE-SPACMA, DISH, FADE, MadDE, SLDE, AL-SHADE and MIDE can only obtain 8, 5, 1, 2, 9, 4 and 3 best solutions. For 50D, APDSDE gets best results on 13 out of 30 functions, which is far more than the comparison algorithms. Among the comparison algorithms, DISH and SLDE can obtain the best solutions on 7 functions, LSHADE-SPACMA on 9, MadDE on 1, AL-SHADE on 4, and MIDE on 3. According to the above analysis, the proposed APDSDE outperforms other comparative methods.

The statistical results of the Wilcoxon rank sum test obtained by APDSDE and the competitors on the CEC 2017 test suite (10D, 30D, 50D) are summarized in Table 7. In Table 7, for 10D, APDSDE outperforms other seven comparison algorithms in 106 functions and underperforms other algorithms in 11 functions. On more than 50.5% of the functions, APDSDE performs better than the competitors. For 30D, APDSDE outperforms other seven comparison algorithms in 136 functions and underperforms other algorithms in 22 functions. On more than 64.8% of the functions, APDSDE performs better than the competitors. For 50D, APDSDE outperforms

	LSHADE-SPACMA	DISH	FADE	MadDE	SLDE	AL-SHADE	MIDE	APDSDE
F1	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	8.62e−03 (6.06e−02)−	2.84e−14 (6.14e−14)−	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**
F2	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	1.00e−14 (1.87e−14)−	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**
F3	**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)**
F4	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	2.61e−10 (6.46e−10)−	2.79e−14 (3.48e−14)−	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**
F5	1.70e+00 (8.97e−01)−	2.97e+00 (8.56e−01)−	9.57e+00 (3.92e+00)−	4.04e+00 (1.08e+00)−	2.63e+00 (1.01e+00)−	1.40e+00 (9.37e−01)≈	4.96e+00 (2.13e+00)−	**1.37e+00** **(1.05e+00)**
F6	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	9.27e−06 (3.16e−05)−	6.97e−09 (4.98e−08)−	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**
F7	**1.10e+01** **(3.30e−01)+**	1.28e+01 (9.02e−01)−	2.06e+01 (5.36e+00)−	1.49e+01 (1.24e+00)−	1.27e+01 (7.36e−01)−	1.17e+01 (5.67e−01)≈	1.48e+01 (3.68e+00)−	1.17e+01 (6.51e−01)
F8	**1.19e+00** **(7.72e−01)**≈	2.98e+00 (9.33e−01)−	9.56e+00 (4.86e+00)−	4.86e+00 (1.37e+00)−	2.85e+00 (9.34e−01)−	1.39e+00 (1.04e+00)≈	5.70e+00 (2.19e+00)−	1.31e+00 (1.04e−01)
F9	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	1.76e−03 (1.25e−02)≈	**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)**
F10	2.62e+01 (4.39e+01)≈	5.26e+02 (2.54e+02)−	4.23e+02 (2.58e+02)−	1.19e+02 (7.84e+01)−	4.97e+02 (2.29e+02)−	3.30e+01 (4.88e+01)≈	1.03e+02 (1.47e+02)−	**2.59e+01** **(4.13e+01)**
F11	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	2.24e+00 (1.93e+00)−	1.65e+00 (6.28e−01)−	**0.00e+00** **(0.00e+00)**≈	**0.00e+00** **(0.00e+00)**≈	3.51e−01 (4.75e−01)−	**0.00e+00** **(0.00e+00)**
F12	1.18e+02 (9.12e+01)−	5.29e−01 (1.53e+00)≈	1.08e+03 (3.13e+03)−	2.02e+01 (4.41e+01)−	**2.61e−01** **(1.85e−01)**≈	2.65e+01 (5.00e+01)≈	7.37e−01 (2.13e+00)≈	1.43e+01 (3.87e+01)
F13	4.09e+00 (2.34e+00)≈	3.67e+00 (1.91e+00)−	7.21e+00 (4.19e+00)−	2.93e+00 (1.90e+00)≈	3.32e+00 (2.17e+00)≈	3.94e+00 (2.01e+00)−	2.62e+00 (2.53e+00)≈	**2.53e+00** **(2.34e+00)**
F14	1.17e−01 (3.80e−01)−	1.52e−01 (3.99e−01)−	6.11e+00 (6.75e+00)−	8.24e−01 (5.56e−01)−	1.76e−01 (3.83e−01)−	2.15e−01 (5.00e−01)−	5.17e−01 (6.31e−01)−	**0.00e+00** **(0.00e+00)**
F15	3.70e−01 (2.55e−01)−	2.49e−01 (2.19e−01)−	1.95e+00 (1.71e+00)−	3.80e−01 (3.10e−01)−	2.40e−01 (2.32e−01)≈	**2.05e−01** **(2.15e−01)**≈	2.16e−01 (2.58e−01)≈	1.85e−01 (2.22e−01)
F16	7.40e−01 (4.12e−01)−	9.48e−01 (2.15e+00)−	8.82e+00 (2.92e+01)−	4.94e−01 (2.16e−01)−	3.62e−01 (2.18e−01)≈	3.22e−01 (2.70e−01)≈	4.82e−01 (2.91e−01)−	**3.04e−01** **(1.95e−01)**
F17	**1.47e−01** **(1.51e−01)+**	1.08e+00 (6.37e−01)−	6.85e+00 (8.31e+00)−	2.55e−01 (2.74e−01)+	5.58e−01 (4.64e−01)≈	2.12e−01 (2.65e−01)+	6.35e+00 (8.36e+00)−	4.08e−01 (3.43e−01)
F18	3.97e+00 (7.67e+00)−	2.54e−01 (2.19e−01)≈	6.32e+00 (8.28e+00)−	2.96e−01 (2.39e−01)≈	2.07e−01 (1.99e−01)≈	2.31e−01 (1.95e−01)≈	**1.05e−01** **(1.74e−01)+**	2.46e−01 (1.96e−01)
F19	2.00e−01 (3.35e−01)−	1.65e−02 (1.12e−02)−	8.03e−01 (7.99e−01)−	3.06e−02 (1.42e−02)−	1.48e−02 (1.85e−02)≈	1.14e−02 (1.62e−02)≈	1.68e−02 (7.73e−03)−	**9.95e−03** **(1.08e−02)**
F20	3.12e−01 (1.65e−01)−	1.23e−02 (6.12e−02)≈	2.69e+00 (5.82e+00)−	4.46e−15 (3.18e−14)≈	1.22e−02 (6.12e−02)≈	1.84e−02 (7.42e−02)≈	2.65e−01 (2.49e−01)−	**0.00e+00** **(0.00e+00)**
F21	**1.05e+02** **(2.01e+01)**≈	1.35e+02 (4.98e+01)−	1.12e+02 (4.04e+01)≈	1.00e+02 (3.06e−01)−	1.35e+02 (4.97e+01)−	1.55e+02 (5.20e+01)−	1.57e+02 (5.36e+01)−	1.24e+02 (4.43e+01)
F22	1.00e+02 (9.73e−02)−	1.00e+02 (3.18e−13)−	8.23e+01 (3.65e+01)−	9.06e+01 (2.11e+01)−	1.00e+02 (2.30e−13)−	**1.00e+02** **(0.00e+00)**≈	9.07e+01 (2.99e+01)−	**1.00e+02** **(0.00e+00)**
F23	2.96e+02 (4.12e+01)−	3.04e+02 (1.72e+00)−	3.05e+02 (4.39e+01)−	2.75e+02 (9.17e+01)−	3.04e+02 (1.66e+00)−	**3.01e+02** **(1.73e+00)**≈	3.05e+02 (2.61e+00)−	3.02e+02 (1.43e+00)
F24	2.82e+02 (9.28e+01)≈	3.13e+02 (6.28e+01)−	2.63e+02 (1.16e+02)−	**8.75e+01** **(3.27e+01)+**	3.04e+02 (7.52e+01)−	3.18e+02 (4.51e+01)≈	3.06e+02 (7.52e+01)−	2.84e+02 (9.19e+01)
F25	4.19e+02 (2.29e+01)−	4.08e+02 (1.88e+01)≈	4.10e+02 (2.02e+01)−	**3.99e+02** **(6.39e+00)+**	4.05e+02 (1.67e+01)≈	4.11e+02 (2.09e+01)−	3.99e+02 (6.29e+00)−	4.01E+02 (1.08e+01)
F26	3.00e+02 (0.00e+00)≈	3.00e+02 (0.00e+00)≈	3.00e+02 (0.00e+00)≈	**1.37e+02** **(1.48e+02)**≈	3.00e+02 (0.00e+00)≈	3.00e+02 (0.00e+00)≈	3.00e+02 (0.00E+00)≈	3.00e+02 (0.00e+00)
F27	3.90e+02 (8.25e−01)−	3.89e+02 (1.80e−01)≈	3.92e+02 (2.67e+00)−	**3.88e+02** **(8.40e−01)+**	3.89e+02 (1.58e−01)−	3.89e+02 (2.05e−01)≈	3.89e+02 (6.65e−01)≈	3.89e+02 (4.17e−01)
F28	3.14e+02 (6.22e+01)≈	3.63e+02 (1.15e+02)−	**2.94e+02** **(4.20e+01)**≈	2.71e+02 (9.01e+01)−	3.33e+02 (8.44e+01)−	3.47e+02 (1.11e+02)−	3.00e+02 (0.00e+00)≈	3.06e+02 (4.37e+01)
F29	**2.31e+02** **(2.56e+00)+**	2.36e+02 (3.04e+00)≈	2.54e+02 (1.74e+01)−	2.49e+02 (5.87e+00)−	2.35e+02 (3.07e+00)≈	2.32e+02 (2.61e+00)+	2.33e+02 (3.23e+00)+	2.35e+02 (3.31e+00)
F30	3.25e+04 (1.60e+05)−	1.64e+04 (1.14e+05)−	6.23e+02 (2.68e+02)−	8.02e+02 (9.92e+02)−	4.14e+02 (3.23e+01)−	1.64e+04 (1.14e+05)≈	4.02e+02 (1.20e+01)−	**3.95e+02** **(3.94e−03)**
+/−/≈	3/13/14	0/16/14	0/24/6	4/20/6	0/12/18	2/5/23	2/16/12	

Table 4. Experimental results ($D = 10$).

	LSHADE-SPACMA	DISH	FADE	MadDE	SLDE	AL-SHADE	MIDE	APDSDE
F1	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	6.90e−01 (2.62e+00)−	1.98e+03 (4.53e+02)−	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	0.00e+00 (9.11e−15)−	**0.00e+00 (0.00e+00)**
F2	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	8.80e+00 (1.76e+01)−	5.73e+14 (8.52e+14)−	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	8.09e−08 (3.45e−07)−	**0.00e+00 (0.00e+00)**
F3	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	8.76e+03 (1.19e+04)−	2.65e+04 (1.29e+04)−	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	1.36e−12 (7.39e−13)−	**0.00e+00 (0.00e+00)**
F4	5.86e+01 (1.14e−14)+	5.86e+01 (1.97e−14)≈	**2.29e+01 (2.85e+01)**+	9.25e+01 (1.59e+01)−	5.86e+01 (1.14e−14)+	5.86e+01 (8.04e−15)+	5.88e+01 (1.31e+00)−	5.86e+01 (2.27e−14)
F5	4.08e+00 (2.53e+00)−	1.38e+01 (2.36e+00)−	5.74e+01 (1.48e+01)−	7.76e+01 (9.00e+00)−	1.45e+01 (2.42e+00)−	4.13e+00 (2.29e+00)−	2.20e+01 (6.75e+00)−	**2.21e+00 (1.69e+00)**
F6	**0.00e+00 (0.00e+00)**≈	2.68e−09 (1.92e−08)≈	2.33e−01 (3.10e−01)−	1.15e−01 (3.44e−02)−	2.28e−08 (8.37e−08)−	6.71e−10 (4.79e−09)≈	1.33e−06 (3.50e−06)−	**0.00e+00 (0.00e+00)**
F7	**3.40e+01 (9.33e−01)**+	4.37e+01 (2.87e+00)−	9.13e+01 (1.92e+01)−	1.06e+02 (9.49e+00)−	4.50e+01 (2.83e+00)−	3.61e+01 (1.55e+00)≈	5.00e+01 (7.66e+00)−	3.62e+01 (1.81e+00)
F8	**3.20e+00 (1.74e+00)**≈	1.44e+01 (2.56e+00)−	6.13e+01 (1.75e+01)−	7.38e+01 (7.84e+00)−	1.51e+01 (3.13e+00)−	3.75e+00 (1.98e+00)−	2.05e+01 (5.72e+00)−	3.39e+00 (2.35e+00)
F9	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	3.59e+01 (4.86e+01)−	1.50e+01 (9.30e+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)**
F10	1.40e+03 (2.23e+02)−	4.16e+03 (3.48e+02)−	3.18e+03 (6.63e+02)−	2.84e+03 (2.95e+02)−	4.02e+03 (4.77e+02)−	1.44e+03 (1.92e+02)−	2.28e+03 (6.13e+02)−	**1.33e+03 (1.47e+02)**
F11	1.36e+01 (2.11e+01)≈	6.62e+00 (8.60e+00)≈	5.35e+01 (2.64e+01)−	7.58e+01 (1.61e+01)−	**5.09e+00 (2.46e+00)**≈	1.87e+01 (2.61e+01)≈	8.75e+00 (1.15e+01)−	9.36e+00 (1.74e+01)
F12	4.97e+02 (2.44e+02)+	9.07e+01 (7.45e+01)+	9.74e+03 (7.97e+03)−	3.61e+05 (1.35e+05)−	**8.96e+01 (9.51e+01)**+	1.15e+03 (3.96e+02)≈	4.17e+03 (4.10e+03)−	1.10e+03 (4.07e+02)
F13	1.45e+01 (4.99e+00)≈	1.96e+01 (4.32e+00)−	1.56e+03 (1.38e+03)−	1.35e+04 (4.12e+03)−	1.98e+01 (3.28e+00)−	1.68e+01 (1.16e+01)−	1.90e+01 (5.66e+00)−	**1.41e+01 (4.86e+00)**
F14	2.29e+01 (1.62e+00)−	2.26e+01 (3.82e+00)−	4.85e+01 (2.15e+01)−	8.54e+01 (1.89e+01)−	2.24e+01 (5.00e+00)−	2.10e+01 (9.36e−01)≈	2.55e+01 (5.50e+00)−	**1.93e+01 (4.88e+00)**
F15	4.65e+00 (2.32e+00)−	3.94e+00 (1.53e+00)−	4.50e+02 (6.87e+02)−	3.25e+02 (2.65e+02)−	4.09e+00 (1.64e+00)−	3.57e+00 (1.59e+00)−	3.63e+00 (1.70E+00)−	**2.90e+00 (1.75e+00)**
F16	4.61e+01 (6.36e+01)≈	1.63e+02 (1.03e+02)−	8.65e+02 (2.66e+02)−	4.27e+02 (9.80e+01)−	1.32e+02 (1.02e+02)−	3.90e+01 (5.52e+01)≈	7.25e+01 (1.12e+02)≈	**4.49e+01 (5.55e+01)**
F17	3.20e+01 (9.78e+00)≈	4.96e+01 (9.09e+00)−	2.55e+02 (1.63e+02)−	7.08e+01 (1.26e+01)−	4.82e+01 (8.28e+00)−	3.01e+01 (7.63e+00)≈	3.37E+01 (1.04E+01)−	**2.83e+01 (5.08e+00)**
F18	2.34e+01 (1.96e+00)−	**2.08e+01 (3.78e−01)**+	8.86e+03 (1.16e+04)−	5.56e+04 (3.07e+04)−	2.09e+01 (3.94e−01)+	2.24e+01 (1.07e+00)≈	2.27e+01 (5.40e+00)−	2.21e+01 (1.60e+00)
F19	9.51e+00 (2.13e+00)−	8.47e+00 (2.02e+00)−	1.17e+02 (1.80e+02)−	2.15e+02 (3.68e+02)−	8.61e+00 (2.12e+00)−	6.13e+00 (1.92e+00)≈	5.26e+00 (1.53e+00)−	**5.87e+00 (2.19e+00)**
F20	7.91e+01 (5.44e+01)−	2.48e+02 (2.62e+02)−	2.96e+02 (1.19e+02)−	9.30e+01 (4.62e+01)−	2.61e+02 (2.64e+02)−	2.75e+01 (7.88e+00)−	**1.98e+01 (8.13e+00)**+	2.43e+01 (7.84e+00)
F21	2.08e+02 (4.25e+00)−	2.15e+02 (2.50e+00)−	2.60e+02 (1.58e+01)−	**2.02e+02 (6.58e+01)**≈	2.16e+02 (2.85e+00)−	2.06e+02 (1.90e+00)−	2.20e+02 (6.64e+00)−	2.05e+02 (2.53e+00)
F22	**1.00e+02 (1.95e−13)**+	1.34e+03 (1.72e+03)−	1.01e+02 (1.62e+00)−	1.00e+02 (2.08e−04)−	1.72e+03 (1.77e+03)≈	1.00e+02 (1.44e−14)≈	1.00e+02 (6.33e−14)≈	1.00e+02 (1.44e−14)
F23	3.55e+02 (3.45e+00)−	3.61e+02 (4.08e+00)−	4.12e+02 (1.97e+01)−	4.16e+02 (7.98e+00)−	3.61e+02 (4.67e+00)−	3.50e+02 (3.20e+00)≈	3.66e+02 (8.03e+00)−	**3.50e+02 (2.88e+00)**
F24	4.28e+02 (2.58e+00)−	4.36e+02 (3.18e+00)−	4.86e+02 (1.97e+01)−	4.85e+02 (9.03e+00)−	4.36e+02 (3.50e+00)−	4.27e+02 (2.00e+00)≈	4.40e+02 (8.23e+00)−	**4.26e+02 (1.55e+00)**
F25	3.87e+02 (1.06e−02)+	3.87e+02 (7.16e−03)+	3.89e+02 (6.02e+00)−	3.87e+02 (9.63e−02)−	**3.87e+02 (4.75e−03)**+	3.87e+02 (2.33e−02)−	3.87e+02 (2.43e−02)	3.87e+02 (1.19e−02)
F26	9.51e+02 (4.12e+01)≈	1.07e+03 (4.12e+01)−	1.70e+03 (2.19e+02)−	**2.67e+02 (4.76e+01)**+	1.06e+03 (4.24e+01)−	9.47e+02 (4.86e+01)≈	1.04e+03 (7.81e+01)−	9.36e+02 (3.35e+01)
F27	5.06e+02 (4.42e+00)−	4.96e+02 (5.47e+00)+	5.18e+02 (9.06e+00)−	5.13e+02 (3.62e+00)−	4.96e+02 (6.15e+00)+	5.06e+02 (5.09e+00)−	**4.87e+02 (9.04e+00)**+	5.02e+02 (6.43e+00)
F28	3.23e+02 (4.83e+01)+	3.06e+02 (2.54e+01)≈	3.39e+02 (5.39e+01)−	3.98e+02 (3.65e+00)−	**3.08e+02 (2.88e+01)**+	3.40e+02 (6.09e+01)≈	3.26e+02 (4.76e+01)−	3.13e+02 (3.60e+01)
F29	4.46e+02 (1.24e+01)−	4.48e+02 (4.65e+01)−	6.78e+02 (1.62e+02)−	5.31e+02 (2.68e+01)−	4.33e+02 (3.38e+01)−	4.32e+02 (7.86e+00)≈	4.31e+02 (2.03e+01)≈	**4.28e+02 (1.17e+01)**
F30	2.00e+03 (6.31e+01)≈	1.96e+03 (1.38e+01)≈	2.15e+03 (2.05e+02)−	1.23e+04 (4.71e+03)−	**1.96e+03 (1.29e+01)**+	1.98e+03 (4.86e+01)≈	2.06e+03 (8.16e+01)−	1.98e+03 (4.64e+01)
+/−/≈	6/12/12	4/17/9	1/29/0	1/28/1	7/17/6	1/9/20	2/24/4	

Table 5. Experimental results ($D = 30$).

	LSHADE-SPACMA	DISH	FADE	MadDE	SLDE	AL-SHADE	MIDE	APDSDE
F1	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	2.32e+03 (2.87e+03)−	1.08e+04 (4.05e+03)−	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	2.84e−13 (1.98e−13)−	**0.00e+00 (0.00e+00)**
F2	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	4.48e+10 (2.79e+11)−	1.00e+30 (1.42e+14)−	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	3.39e+02 (1.81e+03)−	**0.00e+00 (0.00e+00)**
F3	**0.00e+00 (0.00e+00)**≈	5.13e−09 (2.18e−08)−	2.38e+04 (2.05e+04)−	1.20e+05 (1.10e+04)−	1.70e−06 (5.86e−06)−	**0.00e+00 (0.00e+00)**≈	7.74e+00 (1.41e+01)−	**0.00e+00 (0.00e+00)**
F4	6.02e+01 (4.29e+01)≈	5.83e+01 (4.79e+01)≈	6.27e+01 (4.07e+01)−	1.15e+02 (2.30e+01)−	5.25e+01 (4.39e+01)−	6.80e+01 (4.84e+01)≈	6.14e+01 (4.67e+01)−	**4.83e+01 (3.98e+01)**
F5	**6.83e+00 (1.57e+00)**≈	2.53e+01 (4.45e+00)−	1.25e+02 (3.05e+01)−	3.07e+02 (1.76e+01)−	2.78e+01 (5.12e+00)−	9.34e+00 (2.77e+00)−	3.49e+01 (8.74e+00)−	7.42e+00 (2.82e+00)
F6	**0.00e+00 (0.00e+00)**+	1.59e−08 (7.51e−08)+	3.61e+00 (2.77e+00)−	1.97e+00 (3.11e−01)−	1.18e−07 (3.29e−07)+	1.45e−03 (2.33e−03)−	3.75e−04 (2.42e−03)−	7.81e−08 (9.11e−08)
F7	**5.75e+01 (9.63e−01)**+	7.29e+01 (4.45e+00)−	2.08e+02 (3.91e+01)−	3.59e+02 (1.72e+01)−	7.97e+01 (6.50e+00)−	6.25e+01 (2.04e+00)−	8.48e+01 (7.13e+00)−	6.07e+01 (1.96e+00)
F8	**6.50e+00 (1.79e+00)**+	2.48e+01 (4.71e+00)−	1.23e+02 (2.49e+01)−	3.12e+02 (1.68e+01)−	2.73e+01 (5.50e+00)−	9.46e+00 (2.74e+00)−	3.51e+01 (1.01e+01)−	7.17e+00 (3.04e+00)
F9	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	5.14e+02 (3.33e+02)−	2.74e+03 (7.70e+02)−	**0.00e+00 (0.00e+00)**≈	**0.00e+00 (0.00e+00)**≈	4.95e−01 (1.13e+00)−	**0.00e+00 (0.00e+00)**
F10	3.93e+03 (6.03e+02)−	8.50e+03 (1.52e+03)−	5.70e+03 (7.76e+02)−	8.16e+03 (3.58e+02)−	7.38e+03 (6.85e+02)−	3.21e+03 (3.52e+02)−	4.67e+03 (9.01e+02)−	**2.95e+03 (3.25e+02)**
F11	3.53e+01 (3.17e+00)+	2.93e+01 (2.29e+00)+	1.16e+02 (3.64e+01)−	3.42e+02 (3.24e+01)−	**2.93e+01 (1.93e+00)**+	4.68e+01 (8.09e+00)−	3.20e+01 (4.87e+00)+	3.90e+01 (7.77e+00)
F12	1.77e+03 (2.78e+02)+	1.22e+03 (3.45e+02)+	5.21e+04 (3.22e+04)−	3.33e+06 (6.28e+05)−	**9.60e+02 (2.69e+02)**+	3.05e+03 (1.56e+03)≈	2.29e+04 (1.47e+04)−	2.84e+03 (7.65e+02)
F13	**4.18e+01 (6.88e+00)**+	5.09e+01 (3.33e+01)≈	4.74e+03 (5.54e+03)−	2.36e+04 (6.70e+03)−	4.92e+01 (2.92e+01)≈	5.75e+01 (3.10e+01)−	6.79e+01 (5.83e+01)≈	4.30e+01 (1.63e+01)
F14	3.02e+01 (2.33e+00)−	**2.70e+01 (2.53e+00)**+	8.48e+02 (9.43e+02)−	1.01e+05 (4.43e+04)−	2.87e+01 (2.08e+00)≈	2.83e+01 (3.07e+00)≈	3.40e+01 (4.97e+00)−	2.87e+01 (3.02e+00)
F15	3.27e+01 (4.54e+00)+	2.22e+01 (2.52e+00)+	2.03e+03 (2.04e+03)−	1.49e+04 (1.82e+03)−	**2.19e+01 (1.99e+00)**+	3.41e+01 (7.52e+00)≈	2.54e+01 (3.24e+00)+	3.67e+01 (9.54e+00)
F16	5.01e+02 (1.65e+02)−	5.69e+02 (1.65e+02)−	1.60e+03 (4.14e+02)−	1.10e+03 (1.66e+02)−	5.75e+02 (1.75e+02)−	3.87e+02 (1.30e+02)≈	5.12e+02 (3.15e+02)−	**3.84e+02 (1.31e+02)**
F17	3.18e+02 (9.61e+01)−	3.79e+02 (1.11e+02)−	1.03e+03 (2.58e+02)−	7.85e+02 (1.21e+02)−	4.00e+02 (1.15e+02)−	2.66e+02 (9.20e+01)≈	2.57e+02 (1.82e+02)≈	**2.65e+02 (8.82e+01)**
F18	3.50e+01 (5.34e+00)+	**2.33e+01 (1.14e+00)**+	2.32e+04 (4.09e+04)−	7.12e+05 (2.74e+05)−	2.39e+01 (1.56e+00)+	4.59e+01 (1.89e+01)≈	3.15e+02 (2.13e+02)−	4.47e+01 (1.36e+01)
F19	2.31e+01 (2.88e+00)+	1.59e+01 (3.27e+00)+	1.78e+03 (2.44e+03)−	1.69e+04 (1.37e+03)−	1.67e+01 (2.70e+00)+	2.27e+01 (4.97e+00)+	**1.37e+01 (2.76e+00)**+	2.85e+01 (7.48e+00)
F20	2.04e+02 (1.04e+02)−	3.10e+02 (1.42e+02)−	8.80e+02 (2.91e+02)−	5.96e+02 (1.35e+02)−	3.76e+02 (2.29e+02)−	1.27e+02 (6.12e+01)≈	2.96e+02 (1.74e+02)−	**1.17e+02 (6.29e+01)**
F21	2.20e+02 (9.40e+00)−	2.28e+02 (4.86e+00)−	3.24e+02 (3.09e+01)−	4.71e+02 (1.65e+01)−	2.30e+02 (5.89e+00)−	2.11e+02 (2.72e+00)−	2.36e+02 (7.14e+00)−	**2.09e+02 (3.36e+00)**
F22	2.57e+03 (1.89e+03)≈	9.08e+03 (2.45e+03)−	6.22e+03 (1.72e+03)−	1.44e+02 (1.44e+01)≈	7.96e+03 (1.34e+03)−	1.98e+03 (1.81e+03)≈	4.14e+03 (1.83e+03)−	**1.87e+03 (1.77e+03)**
F23	4.44e+02 (3.90e+00)−	4.43e+02 (8.37e+00)−	5.68e+02 (4.14e+01)−	7.09e+02 (1.81e+01)−	4.46e+02 (8.20e+00)−	4.34e+02 (4.90e+00)−	4.55e+02 (1.10e+01)−	**4.31e+02 (4.53e+00)**
F24	5.15e+02 (4.95e+00)−	5.27e+02 (7.85e+00)−	6.27e+02 (3.08e+01)−	7.71e+02 (1.74e+01)−	5.26e+02 (7.90e+00)−	5.11e+02 (2.77e+00)−	5.30e+02 (8.69e+00)−	**5.09e+02 (2.70e+00)**
F25	4.81e+02 (2.75e+00)+	**4.81e+02 (2.75e+00)**+	5.42e+02 (4.08e+01)−	6.08e+02 (9.35e−01)−	4.81e+02 (2.76e+00)+	4.81e+02 (2.32e+00)≈	5.01E+02 (3.10E+01)−	4.81e+02 (2.27e+00)
F26	1.17e+03 (3.18e+01)≈	1.35e+03 (7.03e+01)−	2.59e+03 (3.14e+02)−	**3.07e+02 (1.41e+00)**+	1.37e+03 (8.62e+01)−	1.21e+03 (6.10e+01)−	1.33e+03 (9.31e+01)−	1.18e+03 (5.27e+01)
F27	5.42e+02 (1.67e+01)−	**5.11e+02 (1.23e+01)**+	6.30e+02 (5.51e+01)−	7.17e+02 (2.20e+01)−	5.13e+02 (2.37e+01)+	5.39e+02 (1.76e+01)≈	5.13e+02 (1.30e+01)+	5.35e+02 (2.02e+01)
F28	4.61e+02 (1.03e+01)+	4.59e+02 (1.95e−13)+	4.95e+02 (2.03e+01)−	5.57e+02 (8.12e+00)−	**4.59e+02 (1.92e−13)**+	4.80e+02 (2.43e+01)≈	4.62e+02 (1.23e+01)−	4.66e+02 (1.70e+01)
F29	3.88e+02 (4.90e+01)−	3.93e+02 (3.26e+01)−	1.01e+03 (2.64e+02)−	1.11e+03 (1.14e+02)−	3.89e+02 (2.11e+01)−	3.50e+02 (1.09e+01)−	**3.37e+02 (1.74e+01)**≈	3.41e+02 (1.10e+01)
F30	6.90e+05 (6.93e+04)−	6.10e+05 (4.68e+04)+	6.25e+05 (4.85e+04)≈	4.03e+06 (6.09e+05)−	6.08e+05 (3.45e+04)+	6.68e+05 (7.25e+04)≈	**5.92e+05 (1.97e+04)**+	6.22e+05 (3.18e+04)
+/−/≈	11/11/8	11/14/5	0/29/1	1/28/1	10/15/5	1/12/17	5/22/3	

Table 6. Experimental results ($D = 50$).

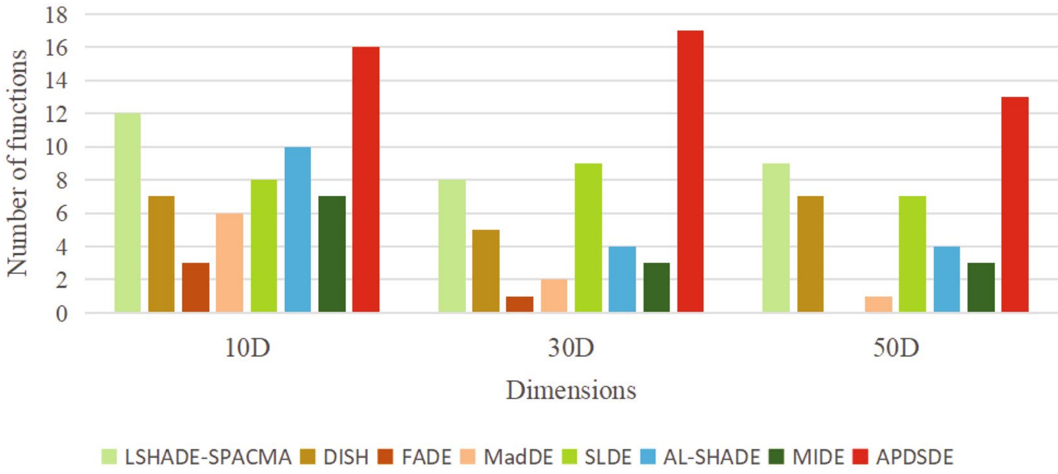

Figure 1. The number of functions.

Algorithm	Symbols	10D	30D	50D	Total
LSHADE-SPACMA	−	13	12	11	36
	+	3	6	11	20
	≈	14	12	8	34
DISH	−	16	17	14	47
	+	0	4	11	15
	≈	14	9	5	28
FADE	−	24	29	29	82
	+	0	1	0	1
	≈	6	0	1	7
MadDE	−	20	28	28	76
	+	4	1	1	6
	≈	6	1	1	8
SLDE	−	12	17	15	44
	+	0	7	10	17
	≈	18	6	5	29
AL-SHADE	−	5	9	12	26
	+	2	1	1	4
	≈	23	20	17	60
MIDE	−	16	24	22	62
	+	2	2	5	9
	≈	12	4	3	19
Total	−	106	136	131	373
	+	11	22	39	72
	≈	93	52	40	185

Table 7. The statistical results of Wilcoxon rank sum tests on CEC2017.

other 7 comparison algorithms in 131 functions and underperforms other algorithms in 39 functions. On more than 62.4% of the functions, APDSDE performs better than the competitors. Therefore, it can be concluded that APDSDE algorithm performs better on high-dimensional functions.

In Table 7, compared to LSHADE-SPACMA, APDSDE is significantly better on 36 test functions, which accounts for 40% of the total functions. Compared to DISH, APDSDE is significantly better on 47 test functions, which accounts for 52.2% of the total functions. Compared to FADE, APDSDE is significantly better on 82 test functions, which accounts for 91.1% of the total functions. Compared to MadDE, APDSDE is significantly better on 76 test functions, which accounts for 84.4% of the total functions. Compared to SLDE, APDSDE is significantly better on 44 test functions, which accounts for 48.9% of the total functions. Compared to AL-SHADE, APDSDE is significantly better on 26 test functions, which accounts for 28.9% of the total functions. Compared to MIDE, APDSDE is significantly better on 62 test functions, which accounts for 68.9% of the total functions. Therefore, it can be seen that APDSDE can achieve better performance in more functions than other algorithms.

To further comprehensively assess the performance of various algorithms, Friedman test is adopted for 10D, 30D and 50D. On the basis of its calculation results, we can get the average rankings of each method for all functions. The Friedman ranking of each algorithm in each dimension is shown in Table 8. The higher the Friedman ranking value, the worse the performance. In Table 8, APDSDE has the smallest Friedman ranking values on 10D ,30D and 50D. APDSDE performs better than the comparison algorithms for all dimensions. According to the value of "Mean Ranking", the "Rank" of all 8 algorithms in Table 8 is obtained, and APDSDE ranks first. Based on the above analysis, it can be seen that the proposed APDSDE generally outperforms the seven comparison methods on the CEC2017 test suite.

To assess the convergence capacity of each method, the convergence curves are shown in Fig. 2. Due to the limitation of space, only the convergence curves of 6 functions on 30D are shown in this paper, which are F1, F6, F9, F13, F19, and F28. As can be seen from Fig. 2, no notable disparity in convergence performance exists between APDSDE and the comparative methods. Among them, the convergence of APDSDE is slightly worse than LSHADE-SPACMA and AL-SHADE, but slightly better than DISH, SLDE, MadDE and MIDE. Therefore, it can be seen that APDSDE has achieved good convergence performance in CEC2017 test suite.

Algorithm complexity

Algorithm complexity is one of the vital references to evaluate algorithm performance. Referring to the method described in reference[42], we calculate the algorithm complexity based on the algorithm running time. The algorithm complexity is calculated by $(\hat{T}_2 - T_1)/T_0$, where T_0 is the time consumed to run the test program, T_1 is the running time for D-dimensional F19 with 200,000 function evaluations, T_2 is the running time of the algorithm to solve the D-dimensional F19 with 200,000 evaluations, and $\hat{T}_2$ is the average of T_2 from 5 runs. Table 9 lists the algorithm complexity of APDSDE and seven comparison algorithms on 10D, 30D and 50D. We can see that the proposed APDSDE has more time consuming compared with AL-SHADE, but has less time consuming compared to the other six methods. The proposed APDSDE method is improved in mutation strategy, control parameters

Algorithm	10D ranking	30D ranking	50D ranking	Mean ranking	Rank
LSHADE-SPACMA	4.40	3.58	3.27	3.75	3
DISH	5.02	4.12	3.63	4.26	5
FADE	6.55	7.10	6.93	6.86	8
MadDE	4.45	6.93	7.37	6.25	7
SLDE	4.12	4.03	3.80	3.98	4
AL-SHADE	4.12	3.27	3.50	3.63	2
MIDE	4.40	4.58	4.67	4.55	6
APDSDE	2.95	2.38	2.83	2.72	1

Table 8. The rankings of APDSDE and the compared methods according to the Friedman test.

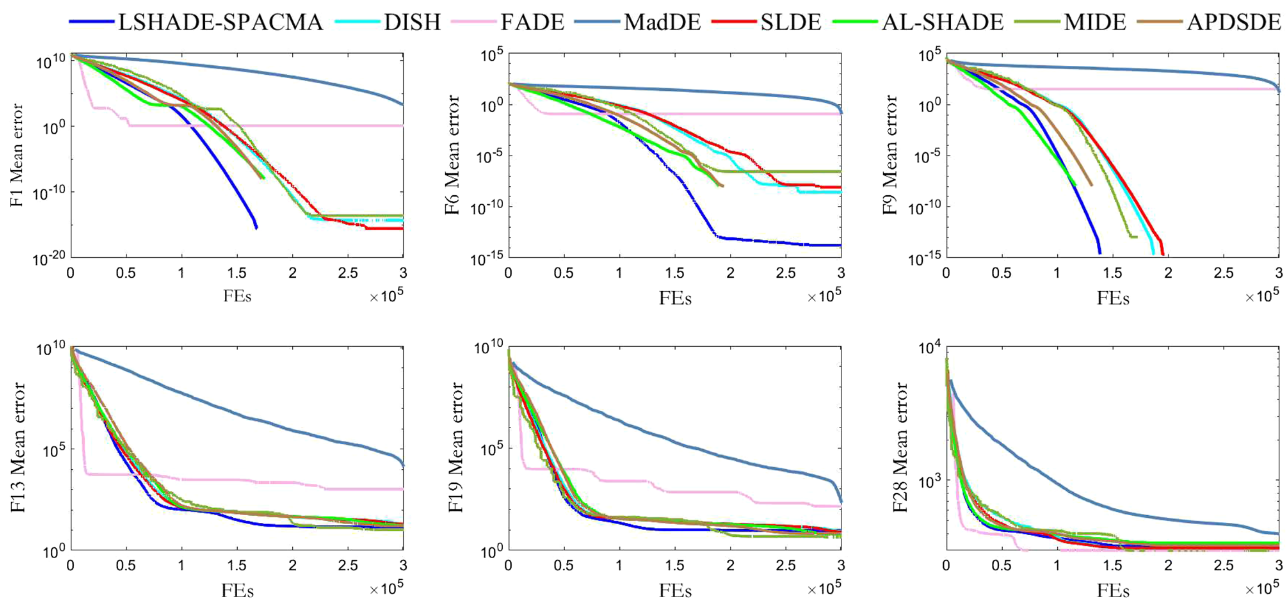

Figure 2. The convergence curve of all methods.

Algorithm	T_0	T_1			$\hat{T}_2$			$(\hat{T}_2 - T_1)/T_0$		
		10D	30D	50D	10D	30D	50D	10D	30D	50D
LSHADE-SPACMA	0.0194	0.3117	0.9684	1.6859	0.7941	1.7678	3.1744	24.8630	41.2105	76.7252
DISH	0.0194	0.3152	0.9870	1.6927	2.3718	3.3386	4.3206	106.0115	121.2136	135.4575
FADE	0.0194	1.3724	4.2548	7.4255	7.4432	12.2208	17.2521	312.9288	410.6171	506.5242
MadDE	0.0194	0.3171	0.9597	1.7034	0.7918	1.7563	2.7043	24.4680	41.0645	51.5978
SLDE	0.0194	0.3308	0.9610	1.6913	14.8781	20.9776	26.8541	749.8614	1031.7839	1297.0475
AL-SHADE	0.0194	0.3081	0.9473	1.6957	0.4782	1.2482	2.0726	8.7721	15.5113	19.4294
MIDE	0.0194	0.3128	0.9747	1.6957	10.1049	10.8802	11.8191	504.7474	510.5916	521.1794
APDSDE	0.0194	0.3165	0.9575	1.6970	0.5261	1.3039	2.1618	10.8232	17.8939	24.0008

Table 9. The complexity of all algorithms.

and population size, which leads to more computation time for APDSDE. This is well worth it because of the significant improvement in optimization performance.

Applications of APDSDE on engineering problems

To assess the efficacy of APDSDE in solving real-world engineering problems, we apply APDSDE to solve two practical problems with constraints[1]: pressure vessel design problem (PVDP) and the tension/compression spring design problem (TCSDP). To deal with constraints in engineering problems, the objective function is calculated using the penalty function. The expression of the penalty function is as follows:

$$\min F(x) = f(x) + \sum_{i=1}^{n} \alpha_i (max(0, g_i(x)))^2, \tag{24}$$

where $F(x)$ is the objective function, n is the number of the constraints, α_i is penalty coefficients, $g_i(x)$ is constraint functions.

As a classical engineering optimization problem, the purpose of the PVDP is to minimize the cost of the vessel design by determining the optimal parameters of the vessel. These parameters include the wall thickness of the vessel (T_v), the thickness of the head (T_h), the inner radius (R) and the length of the cylindrical part of the vessel (L). The objective function of this optimization problem is as follows

$$\text{Minimize:} \quad F(X) = 0.6224 T_v RL + 1.7781 T_h R^2 + 3.1661 T_v^2 L + 19.84 T_v^2 R, \tag{25}$$

$$\text{Subject to:} \quad g_1(X) = 0.0193R - T_v \leq 0, \tag{26}$$

$$g_2(X) = 0.00954R - T_h \leq 0, \tag{27}$$

$$g_3(X) = 1296000 - \frac{4\pi R^3}{3} - \pi R^2 L \leq 0, \tag{28}$$

$$g_4(X) = L - 240 \leq 0. \tag{29}$$

where $X = [T_v, T_h, R, L]$, $0 \leq T_v, T_h \leq 99$ and $10 \leq R, L \leq 200$.

To reduce the random error, the algorithm is executed 51 times independently. Table 10 shows the optimization results of APDSDE algorithm and other comparison algorithms for the pressure vessel design problem,

Algorithm	Best	Worst	Avg	Std
LSHADE-SPACMA	6288.7411	8178.9941	6731.5043	376.0825
DISH	6277.0168	6277.8941	6277.0603	0.1256
FADE	5940.2461	6775.8173	6231.4306	180.0394
MadDE	5885.6722	6742.1060	6178.3585	212.1642
SLDE	6277.0180	6279.9020	6277.1831	0.4352
AL-SHADE	5885.3353	5987.8983	5888.4947	15.7556
MIDE	5885.4031	5933.3069	5893.3592	10.7898
APDSDE	5885.3331	5919.1279	5886.6339	5.3771

Table 10. The optimization results of the APDSDE and comparison algorithms for the PVDP.

Algorithm	Best	Worst	Avg	Std
LSHADE-SPACMA	0.012666021	0.013554022	0.012781831	0.000180974
DISH	0.012666021	0.012727105	0.012674299	1.32e-05
FADE	0.012714957	0.013381029	0.012895922	0.000145861
MadDE	0.012666021	0.013181139	0.012730184	0.000108465
SLDE	0.012666021	0.012810772	0.012674658	2.22e-05
AL-SHADE	0.012666021	0.012727105	0.012670129	1.01e-05
MIDE	0.012666021	0.012754458	0.012670216	1.25e-05
APDSDE	0.012666021	0.012682683	0.012668837	5.71e-06

Table 11. The optimization results of the APDSDE and comparison algorithms for the TCSDP.

including the best value (Best), the worst value (Worst), the average value (Avg) and standard deviation (Std). As can be seen from Table 10, APDSDE has obtained the lowest average cost value, and compared with other algorithms, APDSDE has stronger competitiveness in practical engineering optimization problems.

The TCSDP is another common engineering problem. The aim of this engineering problem is to obtain the minimum weight of the spring under certain constraints. The design parameters of the optimization problem include diameter of the wire (d), diameter of the coil (D) and number of active coils (N). The objective function of this spring design problem is described as follows

$$\text{Minimize:} \quad F(X) = (N+2)Dd^2, \tag{30}$$

$$\text{Subject to:} \quad g_1(X) = 1 - \frac{D^3N}{71785d^4} \le 0, \tag{31}$$

$$g_2(X) = \frac{1}{5108d^2} + \frac{4D^2 - dD}{12566(Dd^3 - d^4)} - 1 \le 0, \tag{32}$$

$$g_3(X) = 1 - \frac{140.45d}{D^2N} \le 0, \tag{33}$$

$$g_4(X) = \frac{d+D}{1.5} - 1 \le 0. \tag{34}$$

where $X = [d, D, N]$, $0.05 \le d \le 2$, $0.25 \le D \le 1.3$ and $2 \le N \le 15$.

Table 11 lists the statistical results obtained by APSDE and comparison algorithms to solve this design problem. The result was achieved by performing 51 runs and using 5,000 function evaluations. Experimental results show that the proposed method is superior to other comparison algorithms in finding the minimum spring weight for the spring design problem.

Conclusions

To augment the performance of DE, a new algorithm, APDSDE, has been introduced. It features dual mutation operators and an adaptive update of control parameters. Within APDSDE, a novel weight update formula for F and CR is implemented. Additionally, APDSDE employs a switching mechanism that integrates dual mutation strategies, effectively harmonizing these improved approaches to balance search diversity and convergence speed. A unique nonlinear population size reduction method is also introduced. CEC2017 test set in 10, 30 and 50 dimensions are utilized to assess the effectiveness of APDSDE, and comparative analysis with LSHADE-SPACMA, DISH, FADE, MadDE, SLDE, AL-SHADE, and MIDE indicates superior overall performance of APDSDE.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 12 April 2024; Accepted: 20 August 2024
Published online: 24 August 2024

References

1. Zhao, W., Wang, L. & Zhang, Z. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl.-Based Syst.* **163**, 283–304 (2019).
2. Hashim, F. A., Hussain, K., Houssein, E. H., Mabrouk, M. S. & Al-Atabany, W. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Appl. Intell.* **51**, 1531–1551 (2021).
3. Gad, A. G. Particle swarm optimization algorithm and its applications: A systematic review. *Arch. Comput. Methods Eng.* **29**, 2531–2561 (2022).

4. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014).
5. Salih, S. Q. & Alsewari, A. A. A new algorithm for normal and large-scale optimization problems: Nomadic people optimizer. *Neural Comput. Appl.* **32**, 10359–10386 (2020).
6. Ahmadianfar, I., Heidari, A. A., Gandomi, A. H., Chu, X. & Chen, H. Run beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Syst. Appl.* **181**, 115079 (2021).
7. Kuo, R.-J. & Zulvia, F. E. The gradient evolution algorithm: A new metaheuristic. *Inf. Sci.* **316**, 246–265 (2015).
8. Simon, D. Biogeography-based optimization. *IEEE Trans. Evolut. Comput.* **12**, 702–713 (2008).
9. Mehmood, K. *et al.* Novel knacks of chaotic maps with Archimedes optimization paradigm for nonlinear ARX model identification with key term separation. *Chaos Solitons Fractals* **175**, 114028 (2023).
10. Mehmood, K., Chaudhary, N. I., Khan, Z. A., Cheema, K. M. & Zahoor Raja, M. A. Atomic physics-inspired atom search optimization heuristics integrated with chaotic maps for identification of electro-hydraulic actuator systems. *Mod. Phys. Lett. B* 2450308 (2024).
11. Mehmood, K., Chaudhary, N. I., Khan, Z. A., Cheema, K. M. & Raja, M. A. Z. Variants of chaotic grey wolf heuristic for robust identification of control autoregressive model. *Biomimetics* **8**, 141 (2023).
12. Tang, H. *et al.* A novel hybrid algorithm based on PSO and FOA for target searching in unknown environments. *Appl. Intell.* **49**, 2603–2622 (2019).
13. Dadgar, M., Jafari, S. & Hamzeh, A. A PSO-based multi-robot cooperation method for target searching in unknown environments. *Neurocomputing* **177**, 62–74 (2016).
14. Kuo, R. & Li, S.-S. Applying particle swarm optimization algorithm-based collaborative filtering recommender system considering rating and review. *Appl. Soft Comput.* **135**, 110038 (2023).
15. Khan, T. A. *et al.* Design of Runge–Kutta optimization for fractional input nonlinear autoregressive exogenous system identification with key-term separation. *Chaos Solitons Fractals* **182**, 114723 (2024).
16. Storn, R. & Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**, 341–359 (1997).
17. Gao, Z., Zhang, M. & Zhang, L. Ship-unloading scheduling optimization with differential evolution. *Inf. Sci.* **591**, 88–102 (2022).
18. Wang, X., Wang, Y., Wong, K.-C. & Li, X. A self-adaptive weighted differential evolution approach for large-scale feature selection. *Knowl.-Based Syst.* **235**, 107633 (2022).
19. Chen, Q., Ding, J., Chai, T. & Pan, Q. Evolutionary optimization under uncertainty: The strategies to handle varied constraints for fluid catalytic cracking operation. *IEEE Trans. Cybern.* **52**, 2249–2262 (2020).
20. Zheng, L. M., Zhang, S. X., Zheng, S. Y. & Pan, Y. M. Differential evolution algorithm with two-step subpopulation strategy and its application in microwave circuit designs. *IEEE Trans. Indus. Inform.* **12**, 911–923 (2016).
21. Fan, Q. & Yan, X. Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies. *IEEE Trans. Cybern.* **46**, 219–232 (2015).
22. Meng, Z. & Yang, C. Hip-DE: Historical population based mutation strategy in differential evolution with parameter adaptive mechanism. *Inf. Sci.* **562**, 44–77 (2021).
23. Zeng, Z., Zhang, M., Zhang, H. & Hong, Z. Improved differential evolution algorithm based on the sawtooth-linear population size adaptive method. *Inf. Sci.* **608**, 1045–1071 (2022).
24. Brest, J., Maučec, M. S. & Bošković, B. Single objective real-parameter optimization: Algorithm jSO. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. 1311–1318 (IEEE, 2017).
25. Xia, X. *et al.* A fitness-based adaptive differential evolution algorithm. *Inf. Sci.* **549**, 116–141 (2021).
26. Poláková, R., Tvrdík, J. & Bujok, P. Differential evolution with adaptive mechanism of population size according to current population diversity. *Swarm Evolut. Comput.* **50**, 100519 (2019).
27. Mohamed, A. K. & Mohamed, A. W. Real-parameter unconstrained optimization based on enhanced AGDE algorithm. In *Machine Learning Paradigms: Theory and Application*. 431–450 (2019).
28. Tanabe, R. & Fukunaga, A. S. Improving the search performance of SHADE using linear population size reduction. In *2014 IEEE Congress on Evolutionary Computation (CEC)*. 1658–1665 (IEEE, 2014).
29. Viktorin, A., Senkerik, R., Pluhacek, M., Kadavy, T. & Zamuda, A. Distance based parameter adaptation for success-history based differential evolution. *Swarm Evolut. Comput.* **50**, 100462 (2019).
30. Mohamed, A. W., Hadi, A. A. & Jambi, K. M. Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization. *Swarm Evolut. Comput.* **50**, 100455 (2019).
31. Zhang, J. & Sanderson, A. C. JADE: Adaptive differential evolution with optional external archive. *IEEE Trans. Evolut. Comput.* **13**, 945–958 (2009).
32. Zheng, L. M., Zhang, S. X., Tang, K. S. & Zheng, S. Y. Differential evolution powered by collective information. *Inf. Sci.* **399**, 13–29 (2017).
33. Wang, H.-B., Ren, X.-N., Li, G.-Q. & Tu, X.-Y. APDDE: Self-adaptive parameter dynamics differential evolution algorithm. *Soft Comput.* **22**, 1313–1333 (2018).
34. Li, Y., Wang, S., Yang, H., Chen, H. & Yang, B. Enhancing differential evolution algorithm using leader-adjoint populations. *Inf. Sci.* **622**, 235–268 (2023).
35. Qin, A. K., Huang, V. L. & Suganthan, P. N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evolut. Comput.* **13**, 398–417 (2008).
36. Wang, Y., Cai, Z. & Zhang, Q. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans. Evolut. Comput.* **15**, 55–66 (2011).
37. Li, Y., Wang, S. & Yang, B. An improved differential evolution algorithm with dual mutation strategies collaboration. *Expert Syst. Appl.* **153**, 113451 (2020).
38. Tanabe, R. & Fukunaga, A. Success-history based parameter adaptation for differential evolution. In *2013 IEEE Congress on Evolutionary Computation*. 71–78 (IEEE, 2013).
39. Mohamed, A. W., Hadi, A. A., Fattouh, A. M. & Jambi, K. M. LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. 145–152 (IEEE, 2017).
40. Brest, J., Maučec, M. S. & Bošković, B. iL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. 1188–1195 (IEEE, 2016).
41. Deng, L., Li, C., Han, R., Zhang, L. & Qiao, L. TPDE: A tri-population differential evolution based on zonal-constraint stepped division mechanism and multiple adaptive guided mutation strategies. *Inf. Sci.* **575**, 22–40 (2021).
42. Li, Y., Han, T., Zhou, H., Tang, S. & Zhao, H. A novel adaptive L-SHADE algorithm and its application in UAV swarm resource configuration problem. *Inf. Sci.* **606**, 350–367 (2022).
43. Biswas, S. *et al.* Improving differential evolution through Bayesian hyperparameter optimization. In *2021 IEEE Congress on Evolutionary Computation (CEC)*. 832–840 (IEEE, 2021).
44. Yang, Q., Yuan, S., Gao, H. & Zhang, W. Differential evolution with migration mechanism and information reutilization for global optimization. *Expert Syst. Appl.* **238**, 122076 (2024).

Acknowledgements
This work was supported in part by the National Natural Science Foundation of China under Grant 62363010, and in part by Jiangxi Double Thousand Plan under Grant SSQ2023018.

Author contributions
Z.Z. Editing, Interpretation, Software. J.Z. Conceptualization, Methodology, Writing-review. F.N. Supervision.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to J.Z.

Reprints and permissions information is available at www.nature.com/reprints.

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