Appl Intell (2017) 47:644–658
DOI 10.1007/s10489-017-0914-3

Self-adaptive differential evolution algorithm with improved mutation mode

Shihao Wang[1,2] · Yuzhen Li[3] · Hongyu Yang[1,2]

Published online: 13 April 2017

Abstract The optimization performance of the Differential Evolution algorithm (DE) is easily affected by its control parameters and mutation modes, and their settings depend on the specific optimization problems. Therefore, a Self-adaptive Differential Evolution algorithm with Improved Mutation Mode (IMMSADE) is proposed by improving the mutation mode of DE and introducing a new control parameters adaptation strategy. In IMMSADE, each individual in the population has its own control parameters, and they would be dynamically adjusted according to the population diversity and individual difference. IMMSADE is compared with the basic DE and the other state-of-the-art DE algorithms by using a set of 22 benchmark functions. The experimental results show that the overall performance of the proposed IMMSADE is better than the basic DE and the other compared DE algorithms.

Keywords Differential evolution · Global optimization · Population diversity · Improved mutation mode · Control parameters adaptation

✉ Shihao Wang
wsh19871024@126.com

[1] School of Aeronautics and Astronautics, Sichuan University, Chengdu 610065, China

[2] National Key Laboratory of Air Traffic Control Automation System Technology, Sichuan University, Chengdu 610065, China

[3] Shanghai Electrical Apparatus Research Institute, Shanghai 200063, China

1 Introduction

Differential evolution, proposed by Storn and Price [1], is a competitive global optimization algorithm to solve real-world optimization problems. DE requires few control parameters and is easy to implement. So far, DE has been widely applied in many fields, such as image processing [2], industrial control [3, 4] and function optimization [1, 5, 6].

However, the optimization performance of DE is very sensitive to control parameters (i.e., population size NP, amplification factor F and crossover probability CR) and mutation modes. Generally, their settings should be first determined when using DE to solve optimization problems. However, choosing appropriate settings is often time-consuming. Although many researchers have given different choice rules [5–7], these rules are not general and unsuitable for practical applications.

In order to deal with the above problems, researchers have proposed many improved DE algorithms. Some work mainly focused on the control parameters adaptation strategy. Liu and Lampien [8] introduced a Fuzzy Adaptive Differential Evolution algorithm (FADE), which uses fuzzy logic controllers to adapt the control parameters for mutation and crossover. Brest et al. [9] proposed an adaptive Differential Evolution algorithm (jDE), which assigns control parameters values to each individual and dynamically adjusts them according to two specified thresholds. Guo et al. [10] proposed a Chaos Differential Evolution algorithm (chDE), where chaos theory is used to dynamically adjust control parameters values. Nasimul et al. [11] developed a new Adaptive Differential Evolution algorithm (aDE), wherein the control parameters values are adaptively adjusted by comparing individual fitness of the offspring with average fitness of the parent population. Brest and

Maucec [12] proposed a novel adaptive DE algorithm that uses a self-adaptation scheme for the control parameters F and CR, and a dynamic population size reduction scheme. Zhu et al. [13] proposed an adaptive population tuning scheme for DE, which uses the solution-searching status and population distribution to dynamically adjust the population size.

Many researchers have also focused on the mutation mode of DE. Qin et al. [14] proposed a Self-adaptive Differential Evolution algorithm (SaDE), in which the adaptation of multiple mutation modes is implemented and the control parameters values are generated according to two normal distributions. Zhang and Sanderson [15] proposed an Adaptive Differential Evolution algorithm with Optional Archive (JADE), which uses a new mutation mode (DE/current-to-$p$best/1) and employs an adaptive strategy for control parameters. Wang et al. [16] introduced a Composite Differential Evolution algorithm (CoDE), which randomly combines three mutation modes with three fixed parameters settings to generate three trail vectors. Thus, the best among them will enter the next generation if it is better than its target vector. Mallipeddi et al. [17] proposed an ensemble of mutation strategies and control parameters of DE algorithm, wherein different mutation strategies are incorporated into a pool and different control parameters settings are incorporated into the other pool. Islam et al. [18] presented an adaptive DE algorithm that uses a novel mutation mode and a modified crossover operation. Elsayed et al. [19] proposed two novel self-adaptive DE algorithms, which incorporate a heuristic mixing of operators and utilize the strengths of multiple mutation and crossover operations. Elsayed et al. [20] proposed a DE with self-adaptive multi-combination strategies, in which the strengths of four mutation strategies, two crossover strategies and two constraint handling techniques are combined. Yi et al. [21] utilized a hybrid mutation operation and self-adaptive control parameters to propose a new DE algorithm, which categorizes the population into two parts. Fan and Yan [22] introduced a Self-adaptive Differential Evolution algorithm with Discrete Mutation Control Parameters (DMPSADE), in which each individual has its own mutation mode and crossover probability, and each variable of the individual has its own amplification factor.

Furthermore, many attempts have also been made to improve the performance of DE by using other methods. Poikolainen et al. [23] proposed a cluster-based population initialization mechanism for DE that consists of three stages, including multiple local search, K-means clustering and composition of the population. Tang et al. [24] introduced an individual-dependent scheme for DE, including individual-dependent control parameters setting and individual-dependent mutation mode. Guo et al. [25] proposed a successful-parent-selecting framework for DE, which can adapt to the selection of parents by storing successful solutions into an archive and select parents from the archive when the stagnation occurs. Wu et al. [26] introduced a multi-population ensemble DE algorithm, in which a multi-population based approach is used to realize an ensemble of three mutation modes.

To further improve DE's performance and make the selections of control parameters values independent of specific optimization problems, based on encoding for control parameters in jDE, a Self-adaptive Differential Evolution algorithm with Improved Mutation Mode (IMMSADE) is proposed by improving "DE/rand/1" mutation mode of the basic DE. In IMMSADE, each individual has its own control parameters, and they will be dynamically adjusted according to the population diversity and individual difference.

The proposed IMMSADE is different from the rest of the improved DE algorithms in two aspects. First and foremost, most work in previous literature mainly focused on the effect of differential vector of "DE/rand/1" mode on DE's performance and neglected the effect of the mutated individual. By contrast, both are simultaneously considered in this paper. Second, most of the previous studies did not quantitatively analyze the population diversity or generate parameter settings in a random manner, while in IMMSADE the population diversity and individual difference are computed quantitatively and utilized to guide the generation of control parameters values throughout the evolution process, and thus it is beneficial to maintain the population diversity and improve the global exploration capability. Therefore, we believe the proposed IMMSADE will be an effective approach to global optimization.

In this paper, a total of 22 benchmark functions from the literatures [27, 28] are used to investigate the performance of the proposed IMMSADE. The experimental results show that the overall performance of IMMSADE on the tested problems is better than the basic DE and the other improved

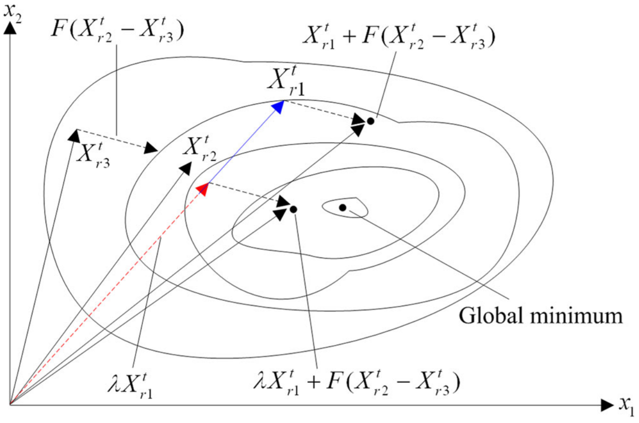

Fig. 1 Improved mutation mode in IMMSADE

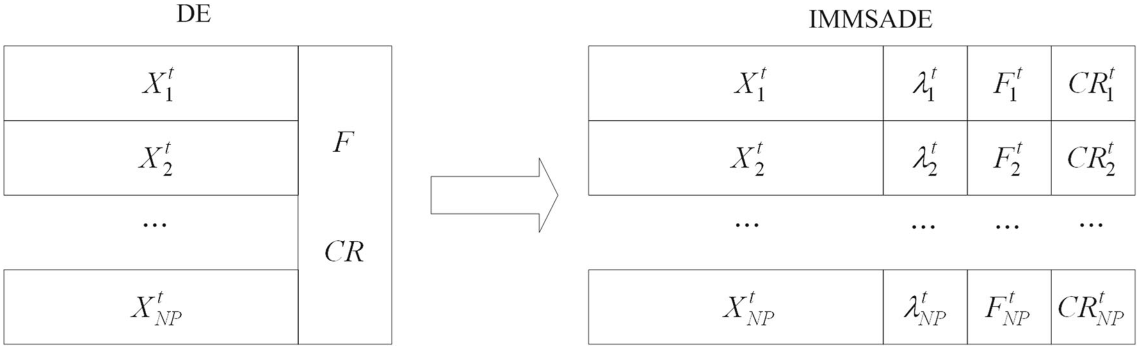

Fig. 2 Encoding for control parameters in IMMSADE

DE algorithms (i.e., jDE, chDE, aDE, SaDE, JADE, CoDE and DMPSADE) on the same problems.

The rest of the paper is organized as follows. Section 2 briefly introduces the basic DE. Section 3 presents the proposed IMMSADE. Section 4 reports the experiments and results analysis. Finally, Section 5 summarizes the conclusions.

2 Differential evolution algorithm

Suppose that the objective function of the optimization problem is min $f(x)$ and the search space is $S \subset \mathrm{R}^D$. The population of the t^{th} generation is denoted by using $\mathbf{X}^t = \{\mathbf{X}_1^t,\ \mathbf{X}_2^t, \cdots, \mathbf{X}_{NP}^t\}$, which contains NP D-dimensional vectors called individuals $\mathbf{X}_i^t = \{x_{i1}^t,\ x_{i2}^t, \cdots, x_{iD}^t\} \in S$ $(i = 1, 2, \cdots, NP)$, where D is the dimension of the problem and NP is the population size. The mutation, crossover and selection operations of the basic DE are shown as follows.

(1) Mutation. For each target individual $\mathbf{X}_i^t$ in the parent population, a mutant vector $\mathbf{V}_i^{t+1} = \left[v_{i1}^{t+1}, v_{i2}^{t+1}, \cdots, v_{iD}^{t+1}\right]$ is generated according to the following equation, which is called “DE/rand/1” mutation mode.

$$\mathbf{V}_i^{t+1} = \mathbf{X}_{r1}^t + F \cdot (\mathbf{X}_{r2}^t - \mathbf{X}_{r3}^t) \tag{1}$$

where the indexes $r1$, $r2$ and $r3$ are distinct integers randomly chosen from the set $\{1, 2, \cdots, NP\}\backslash\{i\}$. The amplification factor $F \in [0, 2]$ is a real number that controls the amplification of the difference vector $\mathbf{X}_{r2}^t - \mathbf{X}_{r3}^t$.

Table 1 The rules for adjusting control parameters in the IMMSADE

α^t	β_i^t	
	negative	non-negative
smaller	retain	adjust
larger	retain	retain

(2) Crossover. A trial vector $\mathbf{U}_i^{t+1} = \left[u_{i1}^{t+1}, u_{i2}^{t+1}, \cdots, u_{iD}^{t+1}\right]$ is generated using the following equation.

$$u_{ij}^{t+1} = \begin{cases} v_{ij}^{t+1} & rand_{ij} \le CR \ or \ j = j_{rand} \\ x_{ij}^{t} & rand_{ij} > CR \ and \ j \ne j_{rand} \end{cases} \tag{2}$$

where CR is the crossover probability within the range [0, 1]. $rand_{ij} \in [0, 1]$ $(j = 1, 2, \cdots, D)$ is a uniform random number. j_{rand} is a randomly chosen integer within [1, D], which ensures that $\mathbf{U}_i^{t+1}$ gets at least one element from $\mathbf{V}_i^{t+1}$, and avoids evolutionary stagnation.

During the evolution process, some components of the trial vector $\mathbf{U}_i^{t+1}$ may violate the search space S. Thus, the invalid components of $\mathbf{U}_i^{t+1}$ should be processed as follows.

$$u_{ij}^{t+1} = \begin{cases} S_l + rand_{ij} \cdot (S_u - S_l) & u_{ij}^{t+1} \notin [S_l,\ S_u] \\ u_{ij}^{t+1} & otherwise \end{cases} \tag{3}$$

where $rand_{ij}$ is uniform random number within the range [0, 1]. S_l and S_u are the lower and upper bounds for the search space S, respectively.

(3) Selection. A greedy strategy is used to perform the selection operation of DE.

$$\mathbf{X}_i^{t+1} = \begin{cases} \mathbf{U}_i^{t+1} & f(\mathbf{U}_i^{t+1}) < f(\mathbf{X}_i^t) \\ \mathbf{X}_i^t & otherwise \end{cases} \tag{4}$$

If and only if, the trial vector $\mathbf{U}_i^{t+1}$ yields a smaller function value than the target vector $\mathbf{X}_i^t$, $\mathbf{X}_i^{t+1}$ is set to $\mathbf{U}_i^{t+1}$, otherwise $\mathbf{X}_i^t$ is retained to enter the next generation.

Table 2 Benchmark functions

Function	D	S	$f_{\min}$
$f_1(x) = \sum_{i=1}^{D} x_i^2$	30	[−100, 100]	0
$f_2(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	30	[−100, 100]	0
$f_3(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$	30	[−100, 100]	0
$f_4(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	30	[−10, 10]	0
$f_5(x) = \max \{ \lvert x_i \rvert ,\ 1 \le i \le D \}$	30	[−100, 100]	0
$f_6(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$	30	[−100, 100]	0
$f_7(x) = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2$	30	[−100, 100]	0
$f_8(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	30	[−100, 100]	0
$f_9(x) = \sum_{i=1}^{D} i x_i^4 + random[0, 1)$	30	[−1.28, 1.28]	0
$f_{10}(x) = \sum_{i=1}^{D-1} \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right)$	30	[−100, 100]	0
$f_{11}(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	30	[−600, 600]	0
$f_{12}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right) - \exp\left(\frac{1}{D} \sum_{i=1}^{D} \cos 2\pi x_i \right) + 20 + e$	30	[−32, 32]	0
$f_{13}(x) = 10D + \sum_{i=1}^{D} \left(x_i^2 - 10 \cos(2\pi x_i) \right)$	30	[−5.12, 5.12]	0
$f_{14}(x) = \sum_{i=1}^{D} (\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))]) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k \cdot 0.5)]$ $a = 0.5,\ b = 3,\ k_{\max} = 20$	30	[−0.5, 0.5]	0
$f_{15}(x) = \sum_{i=1}^{D} \left(0.5 + \frac{\sin^2(\sqrt{x_i^2 + x_{i+1}^2}) - 0.5}{(1 + 0.001(x_i^2 + x_{i+1}^2))^2} \right) \quad x_{D+1} = x_1$	30	[−0.5, 0.5]	0
$f_{16}(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^{D} x_i^2} \right) + 0.1 \sqrt{\sum_{i=1}^{D} x_i^2}$	30	[−100, 100]	0
$f_{17}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right\}$ $+ \sum_{i=1}^{D} u(x_i, 10, 100, 4) \qquad y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[−50, 50]	0
$f_{18}(x) = 0.1 \Big\{ 10 \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] +$ $(x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \Big\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	30	[−50, 50]	0
$f_{19}(x) = 418.9829 \times D - \sum_{i=1}^{D} g(z_i) \quad z_i = x_i + 4.209687462275036e + 02$ $g(z_i) = \begin{cases} z_i \sin(\lvert z_i \rvert^{1/2}) & \lvert z_i \rvert \le 500 \\ (500 - \mathrm{mod}(z_i, 500)) \sin(\sqrt{\lvert 500 - \mathrm{mod}(z_i, 500) \rvert}) - (z_i - 500)^2/10000D & z_i > 500 \\ (\mathrm{mod}(\lvert z_i \rvert, 500) - 500) \sin(\sqrt{\lvert \mathrm{mod}(\lvert z_i \rvert, 500) - 500 \rvert}) - (z_i + 500)^2/10000D & z_i < -500 \end{cases}$	30	[−100, 100]	0
$f_{20}(x) = \frac{10}{D} \prod_{i=1}^{D} (1 + i \sum_{j=1}^{32} \frac{\lvert 2^j x_i - round(2^j x_i) \rvert}{2^j})^{\frac{10}{D^{1.2}}} - \frac{10}{D^2}$	30	[−100, 100]	0

Table 2 (continued)

Function	D	S	$f_{\min}$
$f_{21}(x) = \left\| \sum_{i=1}^{D} x_i^2 - D \right\|^{1/4} + \left(0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i \right) / D + 0.5$	30	[−100, 100]	0
$f_{22}(x) = \left\| \left(\sum_{i=1}^{D} x_i^2 \right)^2 - \left(\sum_{i=1}^{D} x_i \right)^2 \right\|^{1/2} + \left(0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i \right) / D + 0.5$	30	[−100, 100]	0

3 IMMSADE

During the evolution process, each individual $\mathbf{X}_i^t$ represents a solution to the optimization problem, and the final goal of DE is to make $\mathbf{X}_i^t$ infinitely close to the global optimum. The updating of $\mathbf{X}_i^t$ is derived from the mutant vector $\mathbf{V}_i^{t+1}$, which is composed of two components. One is the individual $\mathbf{X}_{r1}^t$ to be mutated, and the other is the difference vector $F \cdot (\mathbf{X}_{r2}^t - \mathbf{X}_{r3}^t)$, called the perturbation. So far, most of the improved DE algorithms were based on the latter, while few algorithms studied the effect of the former on DE's performance.

As illustrated in Fig. 1 (two-dimensional example), a large value of $\mathbf{X}_{r1}^t$ may lead to individual divergence and thus is not beneficial to converge. To address this issue, an improved "DE/rand/1" mode described in (5) is proposed to serve as the basis of IMMSADE.

$$\mathbf{V}_i^{t+1} = \lambda \cdot \mathbf{X}_{r1}^t + F \cdot (\mathbf{X}_{r2}^t - \mathbf{X}_{r3}^t) \tag{5}$$

where λ is a benchmark factor within the range (0, 1]. It is clear that (1) is a special case of (5) by setting λ to 1.

To make the selections of control parameters values independent of specific optimization problems, this paper proposes an improved control parameters adaptation strategy. As illustrated in Fig. 2, the control parameters values of DE are applied at the population level while the control parameters values of IMMSADE are applied at the individual level. In IMMSADE, each individual has its own benchmark factor λ_i^t, amplification factor F_i^t and crossover

Table 3 The mean and standard deviation of the solutions obtained by DE, chDE, jDE, aDE and IMMSADE

Function	DE Mean(STD)	chDE Mean(STD)	jDE Mean(STD)	aDE Mean(STD)	IMMSADE Mean(STD)
f_1	6.45E−37(9.42E−37)	1.02E−56(2.80E−56)	3.24E−39(3.48E−39)	2.63E−56(2.23E−56)	**5.68E−165(0.00E+00)**
f_2	1.27E−05(1.41E−05)	1.65E−04(1.90E−04)	2.58E+00(1.28E+00)	1.23E−01(7.11E−02)	**1.71E−137(9.20E−137)**
f_3	1.08E−33(1.55E−33)	7.16E−53(2.08E−52)	2.11E−36(2.58E−36)	2.93E−53(2.68E−53)	**5.44E−161(2.93E−160)**
f_4	4.16E−18(3.83E−18)	3.06E−34(4.40E−34)	3.40E−23(2.49E−23)	6.75E−33(4.42E−33)	**4.64E−48(2.50E−47)**
f_5	1.99E−01(3.39E−01)	1.02E+01(5.06E+00)	1.91E−05(1.12E−05)	2.77E−04(1.10E−03)	**3.14E−88(1.69E−87)**
f_6	8.13E−31(8.51E−31)	9.73E−51(1.97E−50)	1.57E−33(1.69E−33)	1.71E−50(1.74E−50)	**3.99E−165(0.00E+00)**
f_7	1.49E−36(1.93E−36)	7.19E−56(3.33E−55)	5.34E−39(4.91E−39)	5.15E−56(5.00E−56)	**9.09E−125(4.89E−124)**
f_8	**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_9	4.32E−03(1.30E−03)	3.87E−03(1.02E−03)	4.88E−03(1.42E−03)	4.85E−03(1.41E−03)	**1.33E−04(7.24E−05)**
f_{10}	5.67E−02(1.64E−01)	4.34E+01(2.89E+01)	1.87E+01(2.12E+01)	1.82E+01(1.99E+01)	**0.00E+00(0.00E+00)**
f_{11}	**0.00E+00(0.00E+00)**	2.46E−04(1.33E−03)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
f_{12}	5.06E−15(1.63E−15)	6.13E−15(1.74E−15)	5.89E−15(1.77E−15)	4.59E−15(1.32E−15)	**3.99E−15(0.00E+00)**
f_{13}	1.44E+02(2.20E+01)	2.52E+00(2.13E+00)	4.59E+00(3.58E+00)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
f_{14}	1.18E−09(1.22E−09)	**1.14E−11(1.15E−11)**	9.53E−11(5.87E−11)	1.46E−10(9.32E−11)	3.95E−11(2.16E−11)
f_{15}	**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_{16}	1.74E−01(4.23E−02)	2.30E−01(5.26E−02)	2.00E−01(4.12E−10)	1.90E−01(3.00E−02)	**9.98E−02(5.65E−07)**
f_{17}	**1.35E−19(2.40E−35)**	1.35E−19(3.13E−29)	1.35E−19(2.41E−35)	1.35E−19(2.41E−35)	1.35E−19(2.95E−32)
f_{18}	**1.29E−19(4.81E−35)**	1.29E−19(2.49E−28)	**1.29E−19(4.81E−35)**	**1.29E−19(4.81E−35)**	1.29E−19(5.40E−33)
f_{19}	**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_{20}	1.47E+00(3.40E−01)	**1.18E−02(2.69E−03)**	1.34E−01(6.24E−03)	1.43E−02(1.93E−03)	2.84E−01(1.53E−02)
f_{21}	3.41E−01(4.50E−02)	1.60E−01(4.55E−02)	3.09E−01(4.01E−02)	3.18E−01(4.11E−02)	**6.10E−02(1.72E−02)**
f_{22}	2.72E−01(3.10E−02)	3.33E−01(1.39E−01)	3.03E−01(9.61E−02)	2.97E−01(9.14E−02)	**9.03E−02(3.04E−02)**

Best results are presented in bold

Table 4 SR and MNEG obtained by DE, chDE, jDE, aDE and IMMSADE

Function	DE		chDE		jDE		aDE		IMMSADE	
	MNEG	SR	MNEG	SR	MNEG	SR	MNEG	SR	MNEG	SR
f_1	912	100%	622	100%	870	100%	628	100%	**111**	**100%**
f_2	N/A	0%	N/A	0%	N/A	0%	N/A	0%	**188**	**100%**
f_3	1137	100%	778	100%	1078	100%	785	100%	**141**	**100%**
f_4	1550	100%	843	100%	1217	100%	873	100%	**193**	**100%**
f_5	N/A	0%	N/A	0%	N/A	0%	N/A	0%	**151**	**100%**
f_6	1354	100%	907	100%	1271	100%	916	100%	**154**	**100%**
f_7	932	100%	630	100%	888	100%	643	100%	**126**	**100%**
f_8	344	100%	225	100%	309	100%	227	100%	**47**	**100%**
f_9	N/A	0%	N/A	0%	N/A	0%	N/A	0%	**502**	**100%**
f_{10}	N/A	0%	N/A	0%	N/A	0%	N/A	0%	**169**	**100%**
f_{11}	956	100%	N/A	97%	937	100%	674	100%	**114**	**100%**
f_{12}	1434	100%	946	100%	1340	100%	976	100%	**167**	**100%**
f_{13}	N/A	0%	N/A	20%	N/A	0%	1388	100%	**443**	**100%**
f_{14}	1679	100%	739	100%	1175	100%	818	100%	**174**	**100%**
f_{15}	607	100%	402	100%	571	100%	412	100%	**79**	**100%**
f_{16}	N/A	7%	N/A	3%	N/A	0%	N/A	10%	**833**	**100%**
f_{17}	850	100%	548	100%	793	100%	571	100%	**123**	**100%**
f_{18}	916	100%	602	100%	858	100%	613	100%	**122**	**100%**
f_{19}	849	100%	565	100%	806	100%	580	100%	**105**	**100%**
f_{20}	N/A	17%	120	100%	158	100%	**78**	100%	239	**100%**
f_{21}	N/A	0%	N/A	3%	N/A	0%	N/A	0%	**900**	**100%**
f_{22}	N/A	0%	N/A	0%	N/A	0%	N/A	0%	N/A	**67%**

Best results are presented in bold

probability CR_i^t, and they will be dynamically adjusted during the evolution process. Compared to the fixed parameters values used in the DE, self-adaptively adjusting parameters values is more beneficial to population evolution.

In view of the above analysis, a new differential evolution algorithm, IMMSADE, is proposed in this paper. The mutation and crossover operations of IMMSADE are shown in (6) and (7), respectively.

$$\mathbf{V}_i^{t+1} = \lambda_i^t \cdot \mathbf{X}_{r1}^t + F_i^t \cdot (\mathbf{X}_{r2}^t - \mathbf{X}_{r3}^t) \tag{6}$$

$$u_{ij}^{t+1} = \begin{cases} v_{ij}^{t+1} & rand_j \le CR_i^t \ or \ \ j = j_{rand} \\ x_{ij}^t & rand_j > CR_i^t \ and \ j \ne j_{rand} \end{cases} \tag{7}$$

The contradiction between global exploration and local exploitation affects the performance of DE. Chakrabory et al. [29] considered that the local exploitation capability of DE is strong, while the global exploration capability is poor. Thus, it is more likely to get stuck at a local optimum and thereby lead to premature convergence due to the reduced population diversity. Therefore, properly controlling the population diversity can effectively improve global exploration capability.

During the evolution process, the population fitness diversity can be described using the following equation.

$$\alpha^t = \frac{f_{max}^t - f_{avg}^t}{f_{max}^t} \tag{8}$$

where f_{max}^t and f_{avg}^t denote the maximum fitness and the average fitness of the t^{th} generation, respectively. α^t is a real number between 0 and 1. A larger value of α^t indicates that the individual similarity is lower and the population is more diverse. On the contrary, a smaller value of α^t indicates that the individuals are more similar and the population is less varied.

Additionally, the individual difference in the population can be described using the (9).

$$\beta_i^t = f_i^t - f_{avg}^t \tag{9}$$

If $\beta_i^t < 0$, it indicates that the new offspring $\mathbf{X}_i^t$ of the target individual $\mathbf{X}_i^{t-1}$ is better. On the contrary, if $\beta_i^t \ge 0$, it indicates that the new offspring is worse.

Based on the above analysis, the rules for adjusting control parameters values are obtained, as shown in Table 1.

Important results can be observed from Table 1. First, when the current population presents good diversity whatever the individual's fitness, retaining the current control parameters values is beneficial to improve the global exploration capability. Second, if the population diversity decreases and the individual fitness is less than the average fitness of the current population, the current control parameters values should be protected so that the individual can continue to evolve in the right direction. Third, when the population diversity decreases and the individual fitness is greater than or equal to the average fitness of the current population, the control parameters values should be adjusted so that the new offspring is generated. These new offspring will be beneficial to improve the current population diversity and help the individual escape from a local optimum, and thus avoid premature convergence. Therefore, the rules for adjusting control parameters values not only maintain the population diversity, but also ensure the convergence performance of the algorithm.

If the individual fitness is much greater than the average fitness of the current population, a smaller λ_i^{t+1} and F_i^{t+1}, and a larger CR_i^{t+1} are beneficial to promote the generation of new individual and accelerate the convergence speed. If the individual fitness is close to the average fitness of the current population, a larger λ_i^{t+1} and F_i^{t+1}, and a smaller CR_i^{t+1} are beneficial to maintain the population diversity and enhance the global exploration capability. Thus, λ_i^{t+1}, F_i^{t+1} and CR_i^{t+1} are calculated as follows.

$$\lambda_i^{t+1} = \begin{cases} \lambda_{max} - \frac{f_i^{t+1} - f_{avg}^{t+1}}{f_{max}^{t+1} - f_{avg}^{t+1}} \cdot (\lambda_{max} - \lambda_{min}) & \alpha^{t+1} < \tau \ and \ \beta_i^{t+1} \geq 0 \\ \lambda_i^t & otherwise \end{cases} \tag{10}$$

$$F_i^{t+1} = \begin{cases} F_{max} - \frac{f_i^{t+1} - f_{avg}^{t+1}}{f_{max}^{t+1} - f_{avg}^{t+1}} \cdot (F_{max} - F_{min}) & \alpha^{t+1} < \tau \ and \ \beta_i^{t+1} \geq 0 \\ F_i^t & otherwise \end{cases} \tag{11}$$

$$CR_i^{t+1} = \begin{cases} CR_{min} + \frac{f_i^{t+1} - f_{avg}^{t+1}}{f_{max}^{t+1} - f_{avg}^{t+1}} \cdot (CR_{max} - CR_{min}) & \alpha^{t+1} < \tau \ and \ \beta_i^{t+1} \geq 0 \\ CR_i^t & otherwise \end{cases} \tag{12}$$

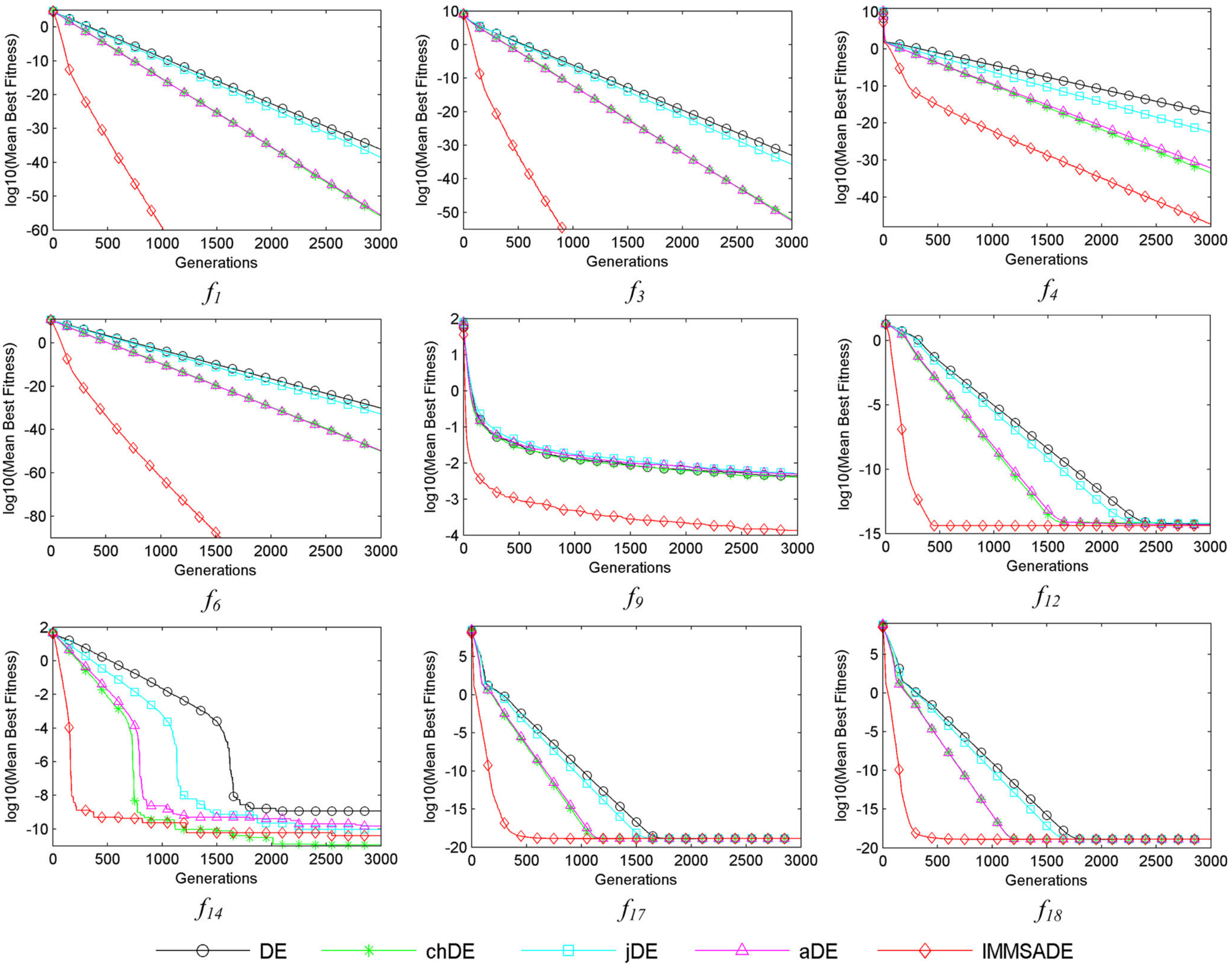

Fig. 3 Evolution curves of DE, chDE, jDE, aDE and IMMSADE for test functions

Table 5 The mean and standard deviation of the optimal solutions obtained by SaDE, JADE, CoDE, DMPSADE and IMMSADE on 30-dimensional problems

Function	SaDE Mean(STD)	JADE Mean(STD)	CoDE Mean(STD)	DMPSADE Mean(STD)	IMMSADE Mean(STD)
f_1	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	4.99E−32(2.49E−29)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
f_2	1.09E−05(2.53E−05)	7.26E−29(8.61E−29)	1.07E−15(1.96E−15)	1.51E−11(2.84E−10)	**0.00E+00(0.00E+00)**
f_3	5.42E+05(2.41E+05)	7.89E+03(5.69E+03)	1.22E+05(7.91E+04)	1.96E+05(9.99E+04)	**0.00E+00(0.00E+00)**
f_{10}	4.67E+01(3.23E+01)	2.94E+00(8.17E+00)	1.42E−01(6.10E−01)	7.48E−06(2.24E−05)	**0.00E+00(0.00E+00)**
f_{11}	2.22E−02(2.13E−02)	8.96E−03(7.99E−03)	8.99E−03(8.76E−03)	7.01E−03(7.92E−03)	**0.00E+00(0.00E+00)**
f_{12}	2.09E+01(4.26E−03)	2.09E+01(1.67E−01)	2.02E+01(1.60E−01)	2.09E+01(5.35E−02)	**3.99E−15(0.00E+00)**
f_{13}	8.56E−02(2.15E−01)	**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_{14}	1.65E+01(2.63E+00)	2.54E+01(1.77E+00)	1.28E+01(3.52E+00)	2.49E+01(4.42E+00)	**1.20E−10(8.15E−11)**
f_{15}	1.26E+01(2.75E−01)	1.23E+01(3.04E−01)	1.24E+01(3.83E−01)	1.24E+01(2.95E−01)	**7.69E+00(4.34E−01)**

Best results are presented in bold

where λ_{min} and λ_{max} are the minimum and maximum values of λ, respectively. F_{min} and F_{max} are the minimum and maximum values of F, respectively. CR_{min} and CR_{max} are the minimum and maximum values of CR, respectively. $\tau \in [0, 1]$ denotes the probability for adjusting population diversity. In this paper, $\lambda_{min} = 0.7$, $\lambda_{max} = 1.0$, $F_{min} = 0.1$, $F_{max} = 0.8$, $CR_{min} = 0.3$, $CR_{max} = 1.0$ and $\tau = 0.7$ are set. Note that the updating of f_{avg}^{t+1}, f_{max}^{t+1}, α^{t+1} and β_i^{t+1} is real-time, and they will be recalculated when a new offspring is generated.

We have made a decision about the values of λ_{min}, λ_{max}, F_{min}, F_{max}, CR_{min} and CR_{max}, which are based on the suggested values in the literature [9] and the experimental results. Additionally, the value of τ can refer to Section 4.4.

λ_i^{t+1}, F_i^{t+1} and CR_i^{t+1} are updated before the mutation operation is performed, and thus these new control parameters values will affect the mutation, crossover and selection operations of the next generation. Moreover, compared with the basic DE, although the space complexity of IMMSADE increases from $O(NP*D)$ to $O(NP*(D+3))$, the time complexity is still $O(T*NP*D)$.

IMMSADE tracks the current population diversity and individual difference in real time and uses them to guide the generation of control parameters values, it is helpful to diversify the difference vector, and thus it effectively avoids premature convergence and is beneficial to search a relatively large space so that more promising regions can be found. Therefore, IMMSADE is able to obtain higher convergence precision within a limited number of evolution generations, and utilize the lesser number of evolution generations to reach the specified convergence precision. Furthermore, the proposed IMMSADE is more reliable.

4 Experiments and results analysis

In this section, IMMSADE is applied to minimize a total of 22 benchmark functions from the literatures [27, 28], as

Table 6 The mean and standard deviation of the optimal solutions obtained by SaDE, JADE, CoDE, DMPSADE and IMMSADE on 50-dimensional problems

Function	SaDE Mean(STD)	JADE Mean(STD)	CoDE Mean(STD)	DMPSADE Mean(STD)	IMMSADE Mean(STD)
f_1	2.96E-30(1.05E-29)	**0.00E+00(0.00E+00)**	3.12E-31(1.83E-30)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
f_2	9.33E-02(8.72E-02)	4.15E-27(3.12E-27)	4.25E-09(6.44E-09)	2.34E-05(2.33E-05)	**0.00E+00(0.00E+00)**
f_3	8.48E+05(1.94E+05)	2.01E+04(1.03E+04)	1.87E+05(6.04E+04)	3.50E+05(1.34E+05)	**0.00E+00(0.00E+00)**
f_{10}	8.24E+01(3.60E+01)	6.70E-01(1.59E+00)	9.86E-01(1.94E+00)	6.23E-01(1.60E+00)	**0.00E+00(0.00E+00)**
f_{11}	5.23E-03(1.02E-02)	4.10E-03(7.14E-03)	5.11E-03(8.32E-03)	4.61E-04(1.73E-03)	**0.00E+00(0.00E+00)**
f_{12}	2.11E+01(3.22E-02)	2.10E+01(4.18E-01)	2.01E+01(8.55E-02)	2.11E+01(4.29E-02)	**3.99E-15(0.00E+00)**
f_{13}	1.34E+00(1.13E+00)	**0.00E+00(0.00E+00)**	8.36E-01(9.37E-01)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
f_{14}	3.65E+01(5.17E+00)	5.31E+01(2.88E+00)	3.08E+01(4.91E+00)	3.99E+01(7.50E+00)	**6.41E-11(1.53E-11)**
f_{15}	2.22E+01(2.65E-01)	2.18E+01(6.78E-01)	2.20E+01(6.13E-01)	2.19E+01(4.21E-01)	**1.68E+01(6.37E-01)**

Best results are presented in bold

Table 7 The ranks obtained by Friedman test and Kruskal-Wallis test for SaDE, JADE, CoDE, DMPSADE and IMMSADE

Algorithm	$D = 30$		$D = 50$	
	Friedman(Rank)	Kruskal-Wallis(Rank)	Friedman(Rank)	Kruskal-Wallis(Rank)
SaDE	4.39	29.39	4.67	30.33
JADE	3.11	25.00	2.89	23.78
CoDE	3.00	24.89	3.22	26.78
DMPSADE	3.17	24.50	3.00	24.11
IMMSADE	**1.33**	**11.22**	**1.22**	**10.00**

Best results are presented in bold

shown in Table 2. IMMSADE is compared with the basic DE and 7 other state-of-the-art DE algorithms, i.e., jDE, chDE, aDE, SaDE, JADE, CoDE and DMPSADE.

Different characteristics for benchmark functions are summarized as follows. $f_1 \sim f_7$ are unimodal functions. f_8 has one minimum and is discontinuous. f_9 is a noisy quadratic function. $f_{10} \sim f_{22}$ are multimodal functions.

4.1 Comparison with the basic DE and 7 improved DE

In this experiment, IMMSADE is compared with the basic DE, jDE [9], chDE [10] and aDE [11]. The parameters of DE are set to $F = 0.5$ and $CR = 0.9$, as recommended in [9]. The parameters settings for jDE, chDE and aDE are the same as in their original papers. The initial parameters λ_i^0, F_i^0 and CR_i^0 of IMMSADE are randomly chosen from [0.7, 1.0], [0.1, 0.8] and [0.3, 1.0], respectively. In all experiments, we set the dimension $D = 30$ and the population size $NP = 100$. For each algorithm and each function, 30 independent runs with 3000 maximum number of evolution generations are conducted.

The Mean and the Standard Deviation (STD) of the optimal solutions obtained by each algorithm for each function are summarized in Table 3. The Success Rate (SR) of each algorithm and the Mean Number of Evolution Generations (MNEG) required to reach the specified convergence precision for each function are reported in Table 4, where "N/A" denotes Not Applicable. The convergence precisions are set as follows: 1.0 for f_{20}, 10^{-1} for f_{16}, f_{21} and f_{22}, 10^{-3} for f_9, and 10^{-8} for all others. MNEG and SR are mainly used to compare the convergence speed and the reliability of each algorithm, respectively. For convenience of illustration, the evolution curves for some functions are plotted in Fig. 3.

Some important observations can be obtained from Fig. 3, Tables 3 and 4:

Table 8 Multiple Sign test results

IMMSADE	DE	chDE	jDE	aDE
Wins(+)	20	20	20	19
Loses(−)	2	2	2	3

(1) In terms of convergence precision and convergence speed, IMMSADE is the best among the five DE algorithms. This can be because IMMSADE can dynamically adjust the population diversity and improve the convergence performance by improving "DE/rand/1" mode and introducing control parameters adaptation strategy. DE performs worst because it converges very slowly and results in the optimal solutions found with lower precisions. jDE, chDE and aDE outperform DE on most functions due to the use of parameters adaptation strategies. It is interesting to note that chDE is slightly better than IMMSADE on f_{14} and f_{20}, and DE is slightly better than IMMSADE on f_{17} and f_{18}.

(2) In terms of the reliability, it is clear that DE, jDE, chDE and aDE perform worse than IMMSADE. For DE, the use of fixed parameters setting is not suitable for various test functions. For example, better results are obtained on unimodal functions f_1 and f_3, while DE fails on multimodal functions f_{10}, f_{13} and f_{16}. For chDE and jDE, although the parameters adaptation strategy is used to improve DE's performance, the updating of control parameters values is a random manner and lacks the guidance of the evolution process. For aDE, the differences of the individuals are not considered when generating new control parameters values, thus easily leading to premature convergence. In contrast, the control parameters values in IMMSADE will be automatically adjusted according to the population diversity, and the differences of the individuals are used to guide the

Table 9 The ranks obtained by Friedman test and Kruskal-Wallis test for DE, chDE, jDE, aDE and IMMSADE

Algorithms	Friedman(Rank)	Kruskal-Wallis(Rank)
DE	3.66	60.52
chDE	3.14	61.16
jDE	3.55	60.68
aDE	2.93	53.73
IMMSADE	**1.73**	**41.41**

Best results are presented in bold

Table 10 Optimization results of DE and IMMSADE

Function	DE			IMMSADE without adaptation			IMMSADE		
	Mean(STD)	MNEG	SR	Mean(STD)	MNEG	SR	Mean(STD)	MNEG	SR
f_1	6.45E−37(9.42E−37)	912	100 %	*3.48E−165(0.00E+00)*	*222*	100 %	**5.68E−165(0.00E+00)**	**111**	100 %
f_2	1.27E−05(1.41E−05)	N/A	0 %	*2.09E−40(4.33E−40)*	*838*	100 %	**1.71E−137(9.20E−137)**	**188**	100 %
f_3	1.08E−33(1.55E−33)	1137	100 %	*5.56E−161(1.96E−160)*	*283*	100 %	**5.44E−161(2.93E−160)**	**141**	100 %
f_4	4.16E−18(3.83E−18)	1550	100 %	**8.89E−81(1.03E−80)**	*369*	100 %	*4.64E−48(2.50E−47)*	**193**	100 %
f_5	1.99E−01(3.39E−01)	N/A	0 %	*1.91E−49(1.62E−49)*	*583*	100 %	**3.14E−88(1.69E−87)**	**151**	100 %
f_6	8.13E−31(8.51E−31)	1354	100 %	*5.00E−159(1.51E−158)*	326	100 %	**3.99E−165(0.00E+00)**	**154**	100 %
f_7	1.49E−36(1.93E−36)	932	100 %	**1.70E−164(0.00E+00)**	231	100 %	*9.09E−125(4.89E−124)*	**126**	100 %
f_8	0.00E+00(0.00E+00)	344	100 %	**0.00E+00(0.00E+00)**	*80*	100 %	**0.00E+00(0.00E+00)**	**47**	100 %
f_9	4.32E−03(1.30E−03)	N/A	0 %	*6.43E−04(1.80E−04)*	N/A	97 %	**1.33E−04(7.24E−05)**	**502**	100 %
f_{10}	5.67E−02(1.64E−01)	N/A	0 %	**0.00E+00(0.00E+00)**	*376*	100 %	**0.00E+00(0.00E+00)**	**169**	100 %
f_{11}	0.00E+00(0.00E+00)	956	100 %	**0.00E+00(0.00E+00)**	*229*	100 %	**0.00E+00(0.00E+00)**	**114**	100 %
f_{12}	5.06E−15(1.63E−15)	1434	100 %	**3.99E−15(0.00E+00)**	*346*	100 %	**3.99E−15(0.00E+00)**	**167**	100 %
f_{13}	1.44E+02(2.20E+01)	N/A	0 %	*1.24E+02(3.50E+01)*	N/A	0.7 %	**0.00E+00(0.00E+00)**	**443**	100 %
f_{14}	1.18E−09(1.22E−09)	1679	100 %	*1.02E−10(2.15E−11)*	*460*	100 %	**3.95E−11(2.16E−11)**	**174**	100 %
f_{15}	0.00E+00(0.00E+00)	607	100 %	**0.00E+00(0.00E+00)**	*145*	100 %	**0.00E+00(0.00E+00)**	**79**	100 %
f_{16}	1.74E−01(4.23E−02)	N/A	7 %	**9.98E−02(7.09E−11)**	**672**	100 %	*9.98E−02(5.65E−07)*	*833*	100 %
f_{17}	**1.35E−19(2.40E−35)**	850	100 %	8.52E−19(5.82E−19)	194	100 %	*1.35E−19(2.95E−32)*	**123**	100 %
f_{18}	**1.29E−19(4.81E−35)**	916	100 %	5.63E−19(3.88E−19)	210	100 %	*1.29E−19(5.40E−33)*	**122**	100 %
f_{19}	0.00E+00(0.00E+00)	849	100 %	**0.00E+00(0.00E+00)**	*206*	100 %	**0.00E+00(0.00E+00)**	**105**	100 %
f_{20}	*1.47E+00(3.40E−01)*	N/A	17 %	1.82E+00(7.67E−02)	N/A	0 %	**2.84E−01(1.53E−02)**	**239**	100 %
f_{21}	3.41E−01(4.50E−02)	N/A	0 %	*1.36E−01(1.90E−02)*	N/A	7 %	**6.10E−02(1.72E−02)**	**900**	100 %
f_{22}	2.72E−01(3.10E−02)	N/A	0 %	**8.62E−02(1.81E−02)**	N/A	77 %	*9.03E−02(3.04E−02)*	N/A	67 %

Best results are presented in bold

generation of new parameters, thus it is beneficial to improve the ability to escape a local minimum and promote the individuals to search towards the right direction. Therefore, IMMSADE performs best.

Additionally, IMMSADE is further compared with SaDE [14], JADE [15], CoDE [16] and DMPSADE [22]. 50 independent runs with 300000 function evaluations are performed and [−100, 100] for f_{15} is used in this simulation. Mean and STD for $D = 30$ and $D = 50$ are summarized in Tables 5 and 6, respectively. The results of SaDE, JADE, CoDE and DMPSADE are taken from [22]. Moreover, two nonparametric statistical tests with a significance level of 5 %, namely, Friedman test and Kruskal-Wallis test [30], are used to carry out multiple comparisons. Table 7 depicts the ranks computed through these two tests. It can be seen from Tables 5, 6 and 7 that IMMSADE significantly performs better than the other competitive DE algorithms.

4.2 Multiple comparisons using nonparametric statistical tests

Recently, nonparametric statistical tests have been widely used to compare the performance of a new proposed algorithm with the existing algorithms, and Derrac et al. [30] also gave a set of nonparametric procedures for multiple comparisons. In this section, to further compare IMMSADE with DE, jDE, chDE and aDE, we label IMMSADE as the control algorithm and reuse the results shown in Tables 3 and 4 to apply the Multiple Sign test with a significance level of 5 %. Table 8 summarizes the statistical analysis results.

Reference to [30] (Table A.21) for $m = 4$ (number of algorithms excluding IMMSADE) and $n = 22$ (number of test functions) reveals the critical value is 5. Since the number of minus signs in the pairwise comparison between IMMSADE and DE, chDE, jDE and aDE is less than the critical value, it may be concluded that IMMSADE has a significantly better performance than the comparison algorithms.

Table 11 The ranks obtained by Friedman test and Kruskal-Wallis test for DE and IMMSADE

Algorithms	Friedman(Rank)	Kruskal-Wallis(Rank)
DE	2.64	39.73
IMMSADE without adaptation	1.89	32.14
IMMSADE	**1.48**	**28.64**

Best results are presented in bold

Moreover, Friedman test and Kruskal-Wallis test in [30] are also used to detect significant differences for multiple algorithms. The ranks obtained are depicted in Table 9. As can be seen in the table, IMMSADE is the best among these five algorithms.

4.3 Effectiveness study of the improved mutation mode in IMMSADE

As shown in Section 4.1, IMMSADE improves DE's performance by improving "DE/rand/1" mutation mode and introducing control parameters adaptation strategy. However, the effectiveness of the improved mutation mode shown in Eq. (5) is not evaluated. To address this issue, IMMSADE without parameters adaptation is compared with the basic DE and IMMSADE. For DE, $F = 0.5$ and $CR = 0.9$ are set. For IMMSADE without parameters adaptation, $\lambda = 0.8$, $F = 0.5$ and $CR = 0.9$ are set. The parameters settings of IMMSADE are the same as in Section 4.1. The optimization results are shown in Table 10, where the results of the best and second best

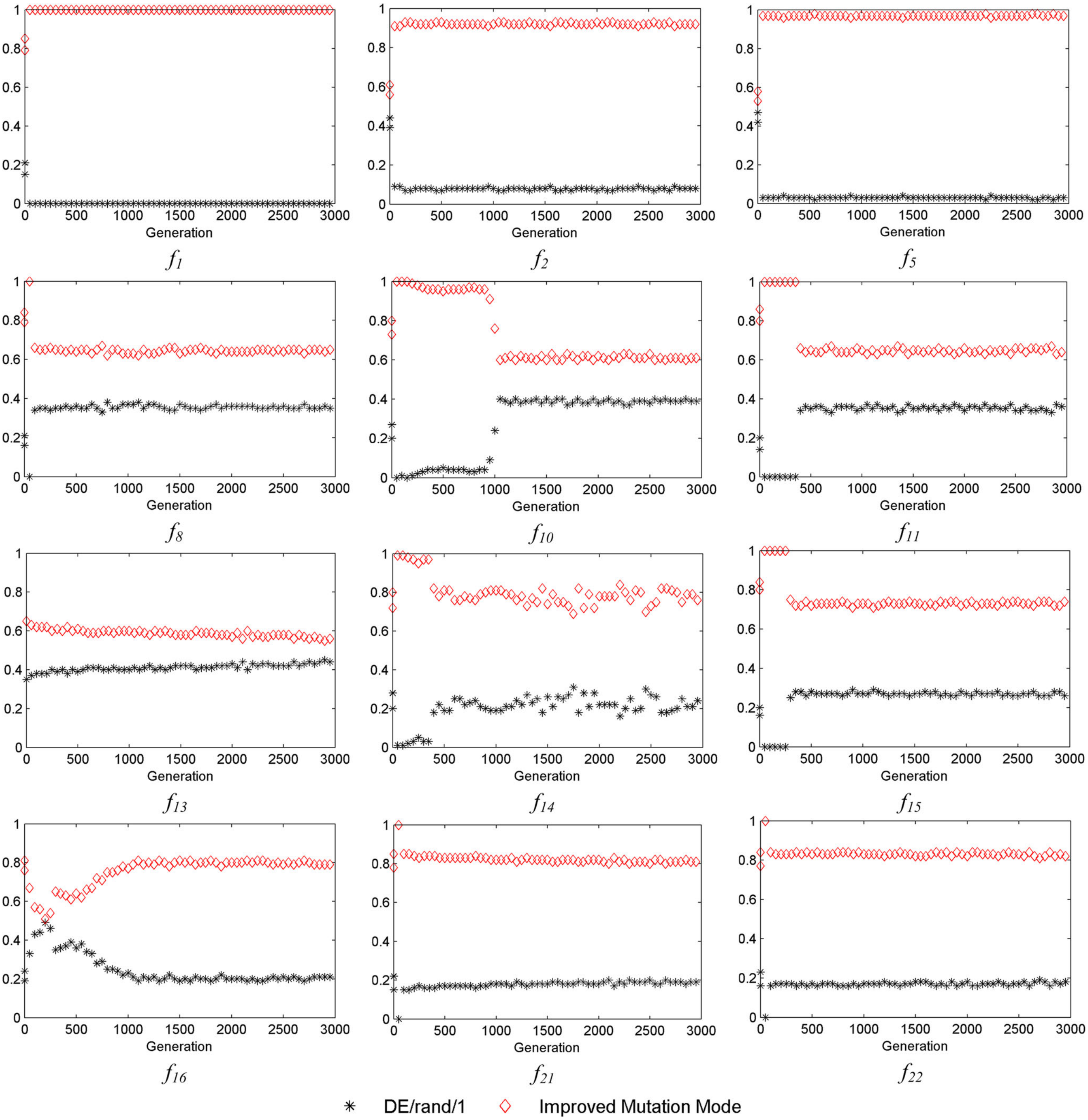

Fig. 4 The probability distribution for DE/rand/1 and the improved mutation mode

Table 12 Experimental results of IMMSADE with different τ

Function	τ = 0.1 Mean(STD)	τ = 0.2 Mean(STD)	τ = 0.3 Mean(STD)	τ = 0.4 Mean(STD)	τ = 0.5 Mean(STD)
f_1	3.29E−53(1.77E−52)	4.38E−51(1.73E−50)	1.50E−53(5.12E−53)	5.77E−80(2.29E−79)	9.42E−93(4.99E−92)
f_2	1.21E−07(4.37E−07)	2.46E−05(8.74E−05)	5.39E−06(1.79E−05)	1.17E−07(3.18E−07)	6.69E−10(1.99E−09)
f_3	3.75E−48(2.01E−47)	1.16E−47(5.74E−47)	6.72E−51(3.45E−50)	2.95E−50(1.27E−49)	8.67E−62(2.96E−61)
f_4	1.36E−30(5.07E−30)	5.27E−29(1.70E−28)	1.70E−33(6.39E−33)	1.81E−54(9.12E−54)	1.97E−50(5.30E−49)
f_5	2.03E−24(8.10E−24)	7.72E−37(3.18E−36)	3.87E−41(2.07E−40)	5.64E−47(1.86E−46)	9.38E−50(5.05E−49)
f_6	3.28E−52(1.71E−51)	1.63E−50(6.09E−50)	1.75E−47(7.32E−47)	4.11E−76(2.14E−75)	3.65E−93(1.51E−92)
f_7	2.26E−53(1.02E−52)	9.69E−54(3.87E−53)	4.75E−52(2.49E−51)	1.79E−47(9.45E−47)	1.94E−59(9.53E−59)
f_8	**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_9	8.30E−04(3.34E−04)	7.74E−04(2.80E−04)	7.77E−04(2.89E−04)	7.55E−04(2.46E−04)	6.05E−04(2.33E−04)
f_{10}	6.51E−21(2.84E−20)	2.36E−21(6.65E−21)	2.18E−19(1.07E−18)	8.95E−19(4.82E−18)	1.32E−26(7.10E−26)
f_{11}	**0.00E+00(0.00E+00)**	**0.00E+0 (0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
f_{12}	**3.99E−15(0.00E+00)**	**3.99E−15(0.00E+00)**	**3.99E−15(0.00E+00)**	**3.99E−15(0.00E+00)**	**3.99E−15(0.00E+00)**
f_{13}	5.57E+00(1.47E+01)	6.21E+00(1.90E+01)	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**	**0.00E+00 (0.00E+00)**
f_{14}	1.85E−10(1.58E−10)	5.72E−11(4.74E−11)	9.52E−11(7.42E−11)	6.52E−11(2.80E−11)	6.36E−11(4.14E−11)
f_{15}	**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_{16}	9.98E−02(1.69E−05)	1.03E−01(1.79E−02)	9.98E−02(6.60E−11)	9.99E−02(1.01E−04)	9.99E−02(1.93E−04)
f_{17}	1.35E−19(7.95E−30)	1.35E−19(1.94E−30)	1.35E−19(8.87E−30)	1.35E−19(7.47E−30)	1.35E−19(1.76E−30)
f_{18}	1.29E−19(2.67E−31)	1.29E−19(1.26E−30)	1.29E−19(1.01E−30)	1.29E−19(1.80E−30)	1.29E−19(9.58E−31)
f_{19}	**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_{20}	4.31E−01(7.94E−02)	4.72E−01(2.11E−02)	4.27E−01(1.89E−02)	3.18E−01(7.30E−03)	3.62E−01(3.19E−02)
f_{21}	1.54E−01(3.11E−02)	1.65E−01(2.97E−02)	1.46E−01(3.82E−02)	6.46E−02(1.76E−02)	5.78E−02(1.68E−02)
f_{22}	1.13E−01(2.33E−02)	1.16E−01(2.23E−02)	1.23E−01(2.45E−02)	7.44E−02(2.41E−02)	**5.86E−02(2.35E−02)**

Function	τ = 0.6 Mean(STD)	τ = 0.7 Mean(STD)	τ = 0.8 Mean(STD)	τ = 0.9 Mean(STD)	τ = 1.0 Mean(STD)
f_1	4.18E−122(1.57E−121)	**5.68E−165(0.00E+ 00)**	4.69E−136(2.52E−135)	8.22E−137(4.43E−136)	2.46E−142(1.33E−141)
f_2	2.04E−116(1.10E−115)	1.71E−137(9.20E−137)	5.16E−208(0.00E+00)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
f_3	5.11E−87(2.75E−86)	5.44E−161(2.93E−160)	2.82E−152(1.52E−151)	**2.45E−167(0.00E+00)**	1.98E−164(0.00E+00)
f_4	2.74E−48(1.47E−47)	4.64E−48(2.50E−47)	5.60E−51(1.80E−50)	2.70E−51(1.03E−50)	**6.94E−52(3.08E−51)**
f_5	1.49E−77(8.01E−77)	3.14E−88(1.69E−87)	6.24E−61(3.36E−60)	5.72E−93(2.78E−92)	**5.67E−97(3.01E−96)**
f_6	7.29E−123(3.93E−122)	3.99E−165(0.00E+00)	**3.98E−169(0.00E+00)**	2.96E−155(1.59E−154)	7.60E−148(4.09E−147)
f_7	2.11E−120(1.13E−119)	9.09E−125(4.89E−124)	9.05E−150(4.86E−149)	3.01E−153(1.62E−152)	**2.30E−159(1.24E−158)**
f_8	**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_9	2.56E−04(1.25E−04)	1.33E−04(7.24E−05)	**1.21E−04(5.25E−05)**	1.28E−04(7.32E−05)	1.21E−04(6.18E−05)
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)**
f_{11}	**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_{12}	**3.99E−15(0.00E+00)**	**3.99E−15(0.00E+00)**	**3.99E−15(0.00E+00)**	**3.99E−15(0.00E+00)**	**3.99E−15(0.00E+00)**
f_{13}	**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_{14}	1.30E−10(5.87E−11)	**3.95E−11(2.16E−11)**	1.25E−10(9.74E−11)	1.01E−10(6.55E−11)	1.07E−10(1.59E−11)
f_{15}	**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_{16}	9.98E−02(3.93E−05)	**9.98E−02(5.65E−07)**	9.99E−02(2.94E−04)	9.99E−02(1.86E−04)	9.99E−02(8.49E−05)
f_{17}	1.35E−19(1.08E−31)	1.35E−19(2.95E−32)	1.35E−19(1.23E−32)	**1.35E−19(1.20E−32)**	1.35E−19(2.20E−30)
f_{18}	1.29E−19(4.20E−32)	1.29E−19(5.40E−33)	1.29E−19(6.81E−33)	1.29E−19(7.28E−33)	**1.29E−19(4.81E−33)**
f_{19}	**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_{20}	2.97E−01(2.26E−02)	2.84E−01(1.53E−02)	2.96E−01(1.70E−02)	**2.82E−01(2.85E−02)**	3.04E−01(1.24E−02)
f_{21}	5.87E−02(1.18E−02)	6.10E−02(1.72E−02)	**5.10E−02(1.21E−02)**	5.57E−02(1.36E−02)	6.11E−02(1.14E−02)
f_{22}	7.43E−02(2.64E−02)	9.03E−02(3.04E−02)	9.41E−02(3.52E−02)	8.81E−02(2.59E−02)	9.06E−02(2.42E−02)

Best results are presented in bold

Table 13 The ranks obtained by Friedman test and Kruskal-Wallis test for IMMSADE with different τ

Method	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.3$	$\tau = 0.4$	$\tau = 0.5$	$\tau = 0.6$	$\tau = 0.7$	$\tau = 0.8$	$\tau = 0.9$	$\tau = 1.0$
Friedman(Rank)	7.59	7.73	7.09	5.98	5.23	5.00	4.20	4.36	**3.77**	4.05
Kruskal-Wallis(Rank)	128.09	128.50	120.30	116.50	112.86	103.14	100.05	99.80	**97.68**	98.09

Best results are presented in bold

algorithms are marked in bold and italic, respectively. Moreover, Table 11 gives the ranks computed by using Friedman test and Kruskal-Wallis test.

From Table 10, it can be concluded that compared to DE, IMMSADE without parameters adaptation performs remarkably better because the improved mutation mode improves the convergence performance by speeding the optimization search in the promising direction. In addition, as can be seen from Table 11, IMMSADE is the best, IMMSADE without parameters adaptation ranks second, whereas DE is the worst. It indicates that the improved mutation mode is effective.

Compared to IMMSADE without parameters adaptation, IMMSADE performs better in terms of convergence precision and convergence speed. It indicates a beneficial cooperation between the improved mutation mode and the parameters adaptation. The parameters adaptation strategy is able to adapt parameters to appropriate values and thus improves the convergence performance. Interestingly, IMMSADE without parameters adaptation is better than IMMSADE on functions f_4, f_7, f_{16} and f_{22}. This might be due to the fact that IMMSADE implements the control parameters adaptation strategy, which may need to frequently adjust the parameters values owing to the reduced population diversity in the later stage of the evolution. However, these parameters values may not be beneficial to the algorithm convergence within a limited number of evolution generations and thereby lead to the final solution with a lower precision. In addition, the parameters adaptation also further improves the success rates on functions f_9, f_{13}, f_{20} and f_{21}. This can be because the parameters adaptation could help the algorithm search better solutions as well as maintain higher convergence speed when solving the functions with a number of local minima.

To further confirm the superiority of the improved mutation mode shown in (5), DE/rand/1 shown in (1) and the improved mutation mode are used to generate two trail vectors for each target vector at each generation. Then, we choose the better one to compare with its target vector. It needs to record n_t at generation t, which is the number of the trail vectors generated by the improved mutation mode which are better than DE/rand/1. Thus, the probability p_t of the better trail vectors generated by the improved mutation mode is calculated according to the following equation, and the probability of DE/rand/1 is $1 - p_t$.

$$p_t = \frac{n_t}{NP} \tag{13}$$

Figure 4 gives the probability distribution for DE/rand/1 and the improved mutation mode in the evolution process. In Fig. 4, X-axis represents the generation count, while Y-axis represents the probabilities of the improved mutation mode and DE/rand/1, which are the average values of 30 independent runs. It can be seen from this figure that the probability of the improved mutation mode is always larger than DE/rand/1. Accordingly, the improved mutation mode can generate better trail vectors and is more beneficial to search the optimal solutions.

4.4 Parameter study of IMMSADE

IMMSADE introduces the parameter τ which determines the updating rates of control parameters (λ_i^t, F_i^t and CR_i^t). Therefore, it is interesting to determine the range of τ which is appropriate for different optimization problems. Table 12 reports the experimental results obtained by setting τ to 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0, respectively. Moreover, the ranks obtained by Friedman test and Kruskal-

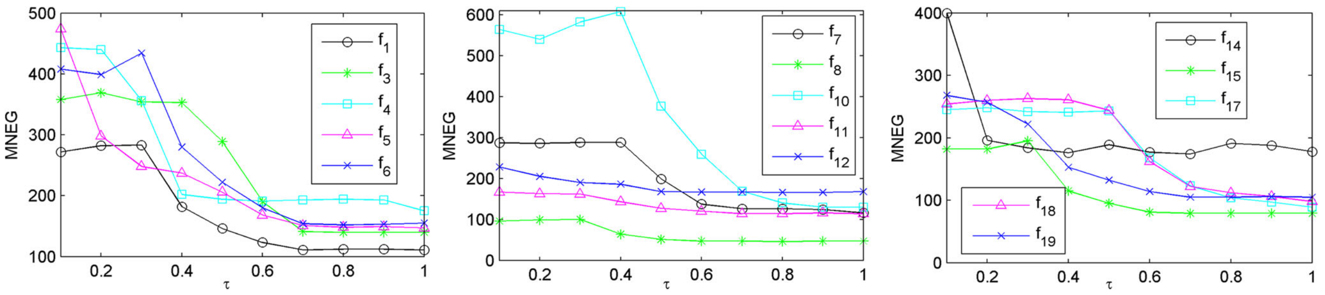

Fig. 5 Curves of MNEG obtained by IMMSADE for benchmark functions

Wallis test are summarized in Table 13 and the curves of MNEG for some functions are plotted in Fig. 5.

As shown in Table 12 and 13, and Fig. 5, IMMSADE performs worse when τ is set to $\tau \leq 0.6$. A smaller value of τ decreases the updating rates of control parameters and thus is not beneficial to generate new individuals and maintain the population diversity. The reduced population diversity leads to premature convergence and thereby decreases convergence precision and convergence speed. In contrast, IMMSADE performs better by setting τ to $\tau \geq 0.7$. A larger value of τ is beneficial to adjust the population diversity and thus improves the global exploration capability. Therefore, IMMSADE can work best with parameter $\tau \in [0.7,\ 1.0]$.

Based on the above empirical analysis, it can be concluded that the proposed IMMSADE has faster convergence speed, higher convergence precision and stronger robustness, thus meeting the requirements for the real time, accuracy and stability of process optimization. Moreover, IMMSADE has been successfully applied to solve arrival flights scheduling, including single runway optimization and dual-runways optimization. The optimization results indicate that IMMSADE can effectively decrease the total delay time of arrival flights sequence, improve the throughput of runways, and relieve the scheduling pressure of controllers.

5 Conclusions

The performance of DE is very sensitive to its control parameters and mutation modes, and choosing appropriate settings for different problems is very time-consuming. Therefore, the paper proposed an IMMSADE to improve DE's performance. A set of test functions from the literatures are used to validate the performance of IMMSADE. Furthermore, IMMSADE is compared with the basic DE and the other improved DE algorithms (i.e., jDE, chDE, aDE, SaDE, JADE, CoDE and DMPSADE). The experimental results show that IMMSADE performs better than the basic DE and the other improved DE algorithms on most test functions.

There are 10 low-dimensional functions in [28], and most algorithms can locate the global optimum within the lesser number of evolution generations. Therefore, we mainly focus on the optimization of high-dimensional functions, and the results of low-dimensional functions are not reported in this paper. Moreover, for some hybrid and composition functions shown in [27], the results of IMMSADE are not satisfactory at this point, but the improved experiments incorporating the advantages of the other evolutionary algorithms are being carried out.

For our future work, the proposed IMMSADE will be used to solve additional optimization problems from various fields, such as PID controller [31], image segmentation [32], pattern recognition [33], financial time series prediction [34], waveform inversion [35] and power system planning [36]. In addition, since the objective function is more than one in many real-world applications, we intend to extend the current research to multi-objective optimization (MOO), especially the conditions optimization of multi-runways arrival flights scheduling. Finally, it is also worthwhile to incorporate the advantage of the other evolutionary algorithms (i.e., particle swarm optimization, simulated annealing and bacterial foraging algorithm) to further improve the performance of the proposed algorithm.

Acknowledgments The authors would like to thank the reviewers for their critical and constructive review of the manuscript.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent This study does not involve any human participants.

Funding This study was funded by National Natural Science Foundation of China (71573184).

References

1. Storn R, Price K (1995) Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, California: University of California, Berkeley
2. Ghosh A, Datta A, Ghosh S (2013) Self-adaptive differential evolution for feature selection in hyperspectral image data. Appl Soft Comput 13(4):1969–1977
3. Marcic T, Stumberger B, Stumberger G (2014) Differential evolution based parameter identification of a line-start IPM synchronous motor. IEEE Trans Ind Electron 61(11):5921–5929
4. Kadhar KMA, Baskar S, Amali SMJ (2015) Diversity controlled self-Adaptive differential evolution based design of non-fragile multivariable PI controller. Eng Appl Artif Intell 46:209–222
5. Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
6. Gamperle R, Muller SD, Koumoutsakos P (2002) A parameter study for differential evolution. WSEAS International Conference on Advances in Intelligent Systems, Fuzzy Systems. Evol Comput 10(10):293–298
7. Zielinski K, Weitkemper P, Laur R et al (2006) Parameter study for differential evolution using a power allocation problem including interference cancellation. IEEE Congress on Evolutionary Computation 1857–1864
8. Liu J, Lampinen J (2005) A fuzzy adaptive differential evolution algorithm. Soft Comput 9(6):448–462

9. Brest J, Greiner S, Boskovic B et al (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10(6):646–657
10. Guo Z, Bo C, Min Y et al (2006) Self-adaptive chaos differential evolution Proceedings of international conference on natural computation 972–975
11. Nasimul N, Danushka B, Hitoshi I (2011) An adaptive differential evolution algorithm. IEEE Congress on Evolutionary Computation 2229–2236
12. Brest J, Maucec MS (2008) Population size reduction for the differential evolution algorithm. Appl Intell 29:228–247
13. Zhu W, Tang Y, Fang JA et al (2013) Adaptive population tuning scheme for differential evolution. Inf Sci 223:164–191
14. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaption for global numerical optimization. IEEE Trans Evol Comput 13:398–417
15. Zhang J, Sanderson AC (2009) JADE: Adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 13:945–958
16. Wang Y, Cai Z, Zhang Q (2011) Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans Evol Comput 15:55–66
17. Mallipeddi R, Suganthan P, Pan Q et al (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. Appl Soft Comput 11:1679–1696
18. Islam SM, Das S, Ghosh S et al (2012) An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. IEEE Trans Syst Man Cybern B Cybern 42:482–500
19. Elsayed SM, Sarker RA, Essam DL (2013) Self-adaptive differential evolution incorporating a heuristic mixing of operators. Comput Optim Appl 54:771–790
20. Elsayed SM, Sarker RA, Essam DL (2014) A self-adaptive combined strategies algorithm for constrained optimization using differential evolution. Appl Math Comput 241:267–282
21. Yi W, Gao L, Li X et al (2015) A new differential evolution algorithm with a hybrid mutation operator and self-adapting control parameters for global optimization problems. Appl Intell 42(4):642–660
22. Fan Q, Yan X (2015) Self-adaptive differential evolution algorithm with discrete mutation control parameters. Expert Syst Appl 42:1551–1572
23. Poikolainen I, Neri F, Caraffini F (2015) Cluster-based population initialization for differential evolution frameworks. Inf Sci 297:216–235
24. Tang L, Dong Y, Liu J (2015) Differential evolution with an individual-dependent mechanism. IEEE Trans Evol Comput 19(4):560–574
25. Guo S, Yang C, Hsu P et al (2015) Improving differential evolution with successful-parent-selecting framework. IEEE Trans Evol Comput 19(5):717–730
26. Wu G, Mallipeddi R, Suganthan PN et al (2016) Differential evolution with multi-population based ensemble of mutation strategies. Inf Sci 329:329–345
27. Liang JJ, Qu BY, Suganthan PN et al (2014) Problem denifition and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization. Zhengzhou University, China, and Nanyang Technological University, Singapore
28. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102
29. Chakrabory UK, Das S, Konar A (2006) Differential evolution with local neighborhood. IEEE Congress on Evolutionary Computation 2042–2049
30. Derrac J, Garcia S, Molina D et al (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18
31. Moharam A, El-Hosseini MA, Ali HA (2016) Design of optimal PID controller using hybrid differential evolution and particle swarm optimization with an aging leader and challengers. Appl Soft Comput 38:727–737
32. Khan A, Jaffar MA, Shao L (2015) A modified adaptive differential evolution algorithm for color image segmentation. Knowl Inf Syst 43(3):583–597
33. Bhadra T, Bandyopadhyay S (2015) Unsupervised feature selection using an improved version of differential evolution. Expert Syst Appl 42(8):4042–4053
34. Dash R, Dash PK, Bisoi R (2014) A self-adaptive differential harmony search based optimized extreme learning machine for financial time series prediction. Swarm Evol Comput 19:25–42
35. Gao Z, Pan Z, Gao J (2014) A new highly efficient differential evolution scheme and its application to wave inversion. IEEE Geosci Remote Sens Lett 11(10):1702–1706
36. Yang G, Dong Z, Wong K (2008) A modified differential evolution algorithm with fitness sharing for power system planning. IEEE Trans Power Syst 23(2):514–522

Shihao Wang received M.S. degree in computer science and technology from Sichuan University, Chengdu, China, in 2014. He has been worked toward the Ph.D. degree in the School of Aeronautics and Astronautics, Sichuan University since 2014. His current research interests include evolutionary algorithms, intelligence optimization and global optimization.

Yuzhen Li received M.S. degree in information engineering from Sichuan University, Chengdu, China in 2014. Her current research interests include intelligence optimization and intelligent traffic.

Hongyu Yang is a professor and Ph.D. supervisor at School of Aeronautics and Astronautics, Sichuan University, Chengdu, China. Her research interests include intelligence optimization, air traffic management and image processing.