ARTICLE IN PRESS

Swarm and Evolutionary Computation xxx (2018) 1–21

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

An improved differential evolution with information intercrossing and sharing mechanism for numerical optimization☆

Mengnan Tian, Xingbao Gao *

School of Mathematics and Information Science, Shaanxi Normal University, Xi'an, Shaanxi 710062, China

ARTICLE INFO

Keywords:
Differential evolution
Information intercrossing and sharing
Mutation strategy
Population classification
Stochastic approach

ABSTRACT

This paper presents a novel differential evolution algorithm by designing a stochastic mixed mutation strategy and an information intercrossing and sharing mechanism. To effectively avoid the premature convergence and enhance the information dissemination between subpopulations under the work specialization, a stochastic mixed mutation strategy is first proposed by incorporating a cosine perturbation into the probability parameter setting and using two mutation strategies with this probability to balance the exploration and exploitation. Then, an information intercrossing and sharing mechanism is developed to make good use of the information of individuals by dividing the population into superior and inferior subpopulations according to their fitness values and exchanging or sharing their information. Furthermore, a simple and efficient approach is applied to adjust control parameters. Finally, the proposed algorithm is compared with twelve typical algorithms by numerical experiments on 55 benchmark functions from both CEC2005 and CEC2014, and is applied to solve the Parameter Estimation for Frequency-Modulated Sound Waves. Experimental results show that the proposed algorithm is very competitive.

1. Introduction

Differential evolution (DE) proposed by Price and Storn in 1995 [1] is a kind of population-based heuristic global search technique. Because of the simple principle, a few control parameters and strong robustness, DE has been applied in many fields such as mechanical engineering design [2], signal processing [3], chemical engineering [4], pattern recognition [5] and so on. However, the optimization capability of DE is still insufficient specially for multimodal problems, where it may easily get stuck at local optima or cause stagnation [6]. Thus, it is necessary to further improve the performance of DE.

Similar to other evolutionary algorithms (EAs) [7–9], the performance of DE heavily depends on the balance between the exploitation and exploration, and different trial vector generation strategies and parameter settings have quite different search characteristics and effects during the evolutionary process. Then, it is important to assign the appropriate strategies or parameters for different problems and evolutionary stages. Consequently, many variants of DE are proposed over the last decades [10–36]. Specially, a survey of the recent advances on DE can be founded in Ref. [10], where the advances on DE at the recent five years are reviewed by different categories including single objective optimization, complex optimization, theoretical analysis, engineering application and so on, and some important future directions of research in the area of DE are also pointed out. Differing from Ref. [10], we classify the improved DE methods simply as follows: a) design a new mutation strategy [11,13–16,18] or parameter setting method [17,19–21]; b) combine the mutation strategies and parameters with different characteristics [11,22–28], which have been proven to be effective in Ref. [45]; c) design the restart mechanism to ensure the population diversity and avoid premature convergence or stagnation [11,12]; d) divide the population under the concept of work specialization and design a suitable mutation strategy for each subgroup [29–32]; and e) combine with other heuristic algorithms to compromise the advantages of them [33–36].

In particular, in order to overcome the difficulty of choosing the best choice in different mutation strategies for a specific problem, many DE variants with adaption strategies have been developed [11,22–28]. Among them, Qin *et al.* [23] proposed a variant of DE (SaDE) by using a self-adaptive probability to choose one trial vector generation strategy to generate offspring from four given strategies based on their pre-

☆ This work is supported by the National Natural Science Foundation of China No. 61273311 and 61502290, and by the Fundamental Research Funds For the Central Universities No. 2017TS002.

* Corresponding author.
E-mail addresses: mengnan_tian@snnu.edu.cn (M. Tian), xinbaog@snnu.edu.cn (X. Gao).

https://doi.org/10.1016/j.swevo.2017.12.010
Received 13 April 2017; Received in revised form 21 December 2017; Accepted 21 December 2017
Available online XXX

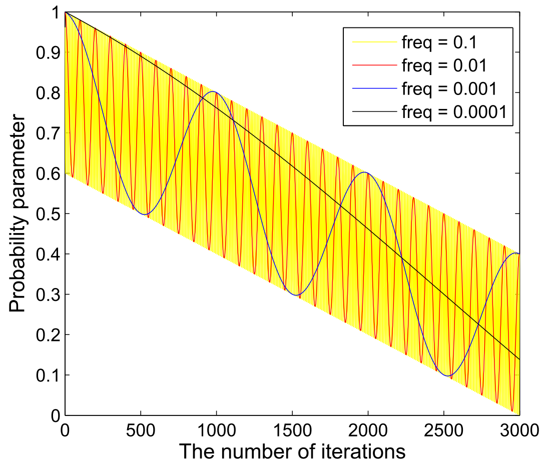


Fig. 1. Curves of ξ_1 with various *freq*.

vious success rate. Unlike [23], by employing single strategy parameter, Gong *et al.* [27] presented a strategy adaptation mechanism to choose the trial vector generation strategy during the evolutionary process. Wang *et al.* [22] designed a DE with composite trial vector generation strategies and control parameters (CoDE), where three mutation strategies are employed in parallel to generate offspring. By using cheap surrogate models, Gong *et al.* [25] presented a cheap surrogate model-based multioperator search strategy. This algorithm first generates several trial individuals by predefining mutation strategies, and then chooses the best one as the offspring by evaluating these trial individuals according to the surrogate model. Even though these methods above have made great progress on improving the performance of DE, they are at the expense of increasing store spaces or computational costs. Meanwhile, by introducing a linear time-varying function to decide the mutation strategy at the phases of mutation, Xiang *et al.* [28] presented a new combined mutation strategy which is composed of two mutation strategies with different searching characteristics to accelerate standard DE and prevent DE from clustering around the global best individual. Mohamed [11] proposed an improved variant of DE by using a iteration-based and decreasing monotonically probability parameter to choose a more suitable mutation strategy. Although the numerical experiments show that the algorithms in Refs. [11,28] are effective, each probability parameter setting might not be suitable since the mutation strategy with more explorative capability would be chosen with a smaller probability as the iteration increases such that the premature convergence might easily occur during the later evolutionary process. Therefore, it is necessary to develop a new adaptive method to choose a more suitable strategy.

Moreover, inspired by the phenomenon that the work efficiency can be improved by using the concept of work specialization, many DE variants with multipopulation have been also developed to effectively balance the exploration and exploitation of DE (see Refs. [29–32]). In particular, Han *et al.* [31] divided the population into better and worse subpopulations dynamically according to their fitness values and assigned the local and global search operators to them, respectively. Cui *et al.* [30] developed an adaptive DE based on multiple subpopulations, where the population is split into three subpopulations and three DE strategies are designed and employed for them respectively to balance the exploitation and exploration. Ali *et al.* [29] proposed a new multipopulation DE algorithm with an ensemble of different mutation and crossover strategies by using a competitive success-based scheme to determine the life cycle of each tribe and its participation rate for next generation. Even though the concept of work specialization is helpful to balance the exploration and exploitation and the methods in Refs. [29–31] have obtained more promising performance, each subgroup only performs its special task during evolutionary process and the information dissemination between different subpopulations has not been

Table 1
Experimental results of ISDE with different values for k with $D = 30$.

Function	statistic	$k = 50$	$k = 100$	$k = 200$	$k = 300$	$k = 500$	$k = 1500$	$k = 3000$
f_1	Mean Error	8.08E-30	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std Dev (Rank)	4.04E-29 (7)	**0.00E+00 (1)**	**0.00E+00 (1)**	**0.00E+00 (1)**	**0.00E+00 (1)**	**0.00E+00 (1)**	**0.00E+00 (1)**
f_2	Mean Error	5.39E-26	9.08E-27	8.09E-27	7.74E-27	5.09E-27	**3.42E-27**	5.56E-27
	Std Dev (Rank)	1.41E-21 (7)	1.06E-26 (6)	3.03E-25 (5)	1.01E-26 (4)	1.07E-26 (2)	**2.66E-27 (1)**	1.75E-26 (3)
f_3	Mean Error	7.75E+04	5.02E+04	**4.04E+04**	4.53E+04	4.66E+04	4.48E+04	4.60E+04
	Std Dev (Rank)	3.41E+04 (7)	2.36E+04 (6)	**1.88E+04 (1)**	2.47E+04 (3)	2.68E+04 (5)	2.50E+04 (2)	2.25E+04 (4)
f_4	Mean Error	2.04E-04	**1.64E-05**	1.05E-04	3.41E-04	4.01E-05	4.32E-04	4.95E-05
	Std Dev (Rank)	5.11E-04 (4)	**4.08E-05 (1)**	2.89E-04 (3)	1.27E-03 (5)	1.26E-04 (6)	4.37E-03 (7)	1.10E-04 (2)
f_5	Mean Error	1.49E+03	9.33E+02	8.37E+02	6.37E+02	7.09E+02	**6.29E+02**	6.46E+02
	Std Dev (Rank)	4.98E+02 (7)	4.10E+02 (6)	3.95E+02 (5)	4.37E+02 (2)	3.44E+02 (4)	**3.07E+02 (1)**	4.82E+02 (3)
f_6	Mean Error	**1.59E-01**	**1.59E-01**	2.57E-01	3.19E-01	3.19E-01	9.57E-01	1.44E+00
	Std Dev (Rank)	**7.97E-01 (1)**	**7.97E-01 (1)**	1.74E+00 (3)	1.10E+00 (4)	1.10E+00 (4)	1.74E+00 (6)	1.95E+00 (7)
f_7	Mean Error	1.87E-02	1.49E-03	**1.26E-03**	1.72E-02	1.78E-02	1.84E-02	1.97E-02
	Std Dev (Rank)	1.83E-02(6)	8.24E-03 (2)	**6.22E-03 (1)**	1.25E-02 (3)	1.38E-02 (4)	1.41E-02 (5)	1.26E-02 (7)
f_8	Mean Error	2.08E+01	2.08E+01	**2.08E+01**	2.08E+01	2.08E+01	2.09E+01	2.09E+01
	Std Dev (Rank)	1.47E-01 (4)	1.36E-01 (2)	**1.03E-01 (1)**	2.37E-01 (5)	1.38E-01 (3)	1.28E-01 (6)	2.15E-01 (7)
f_9	Mean Error	1.19E-01	**0.00E+00**	**0.00E+00**	3.18E-01	5.97E-01	1.99E-01	3.98E-01
	Std Dev (Rank)	3.30E-01 (3)	**0.00E+00 (1)**	**0.00E+00 (1)**	5.54E-01 (5)	6.42E-01 (7)	4.06E-01 (4)	1.99E-01 (6)
f_{10}	Mean Error	4.01E+01	**2.57E+01**	3.85E+01	4.18E+01	3.95E+01	4.33E+01	4.42E+01
	Std Dev (Rank)	8.63E+00 (4)	**6.53E+00 (1)**	1.07E+01 (2)	8.85E+00 (5)	1.16E+01 (3)	9.67E+00 (6)	1.24E+01 (7)
f_{11}	Mean Error	1.46E+01	**1.21E+01**	1.33E+01	1.37E+01	1.43E+01	1.50E+01	1.54E+01
	Std Dev (Rank)	4.64E+00 (5)	**3.38E+00 (1)**	3.26E+00 (2)	3.35E+00 (3)	4.38E+00 (4)	3.32E+00 (6)	3.61E+00 (7)
f_{12}	Mean Error	2.49E+03	**8.74E+02**	3.08E+03	2.61E+03	3.05E+03	2.52E+03	4.30E+03
	Std Dev (Rank)	3.95E+03 (2)	**1.22E+03 (1)**	3.48E+03 (6)	1.71E+03 (4)	2.82E+03 (5)	3.94E+03 (3)	4.14E+03 (7)
f_{13}	Mean Error	1.54E+00	**1.36E+00**	1.60E+00	1.57E+00	1.69E+00	1.69E+00	1.71E+00
	Std Dev (Rank)	2.57E-01 (2)	**1.17E-01 (1)**	2.51E-01 (4)	2.76E-01 (3)	3.17E-01 (6)	2.41E-01 (5)	3.78E-01 (7)
f_{14}	Mean Error	1.23E+01	**1.22E+01**	1.22E+01	1.23E+01	1.24E+01	1.26E+01	1.27E+01
	Std Dev (Rank)	3.72E-01 (5)	**4.35E-01 (1)**	5.86E-01 (2)	4.39E-01 (3)	5.81E-01 (4)	7.94E-01 (6)	3.49E-01 (7)
Sum Rank		64	31	37	50	58	59	75
Aver Rank		4.57	2.21	2.64	3.57	4.14	4.21	5.36

Table 2
Experimental results of ISDE with various values for α with $D = 30$.

Function	$\alpha = 0$	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank
f_1	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**
f_2	1.59E-26(2.69E-26)/5	9.62E-27(6.14E-27)/2	1.19E-26(1.34E-26)/3	**9.08E-27(1.06E-26)/1**	1.53E-26(1.51E-26)/4
f_3	**1.37E+04(8.60E+03)/1**	2.13E+04(9.86E+03)/4	1.81E+04(1.46E+04)/3	5.02E+04(2.36E+04)/5	1.76E+04(8.92E+03)/2
f_4	1.08E-04(2.01E-04)/3	1.55E-04(3.04E-04)/4	3.69E-05(7.25E-05)/2	**1.64E-05(4.08E-05)/1**	1.76E-04(3.80E-04)/5
f_5	2.79E+02(2.12E+02)/2	3.18E+02(2.33E+02)/3	**2.75E+02(2.92E+02)/1**	9.33E+02(4.10E+02)/5	3.62E+02(3.55E+02)/4
f_6	3.58E+00(8.05E+00)/2	4.43E+00(8.71E+00)/5	3.69E+00(8.25E+00)/3	**1.59E-01(7.97E-01)/1**	3.86E+00(8.50E+00)/4
f_7	1.24E-02(1.43E-02)/5	6.99E-03(9.83E-03)/4	5.32E-03(1.11E-02)/2	**1.49E-03(8.24E-03)/1**	6.70E-03(9.20E-03)/3
f_8	2.10E+01(3.58E-02)/5	2.09E+01(4.45E-02)/2	2.09E+01(7.11E-02)/2	**2.08E+01(1.36E-01)/1**	2.09E+01(5.43E-01)/2
f_9	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**
f_{10}	**2.28E+01(5.77E+00)/1**	2.44E+01(5.65E+00)/3	2.29E+01(6.15E+00)/2	2.57E+01(6.53E+00)/4	2.58E+01(6.55E+00)/5
f_{11}	1.60E+01(4.17E+00)/3	1.54E+01(3.34E+00)/2	1.67E+01(4.42E+00)/5	**1.21E+01(3.38E+00)/1**	1.65E+01(4.90E+00)/4
f_{12}	1.38E+03(1.41E+03)/5	1.05E+03(9.79E+02)/3	1.01E+03(1.41E+03)/2	**8.74E+02(1.22E+03)/1**	1.34E+03(1.99E+03)/4
f_{13}	**1.27E+00(1.82E-01)/1**	1.35E+00(1.47E-01)/4	1.29E+00(1.57E-01)/2	1.36E+00(1.17E-01)/5	1.31E+00(2.15E-01)/3
f_{14}	1.25E+01(4.92E-01)/4	1.24E+01(2.89E-01)/3	**1.22E+01(4.76E-01)/1**	**1.22E+01(4.35E-01)/1**	1.25E+01(2.75E-01)/4
Sum Rank	39	41	30	29	46
Aver Rank	2.79	2.93	2.14	2.07	3.29

considered. Obviously, the information intercrossing and sharing is useful to exploit or explore the promising individuals. Thus it is vital to design a new information intercrossing and sharing mechanism under the concept of work specialization.

Based on above discussions, this paper presents a novel differential evolution algorithm (ISDE) by developing a stochastic mixed mutation strategy and an information intercrossing and sharing mechanism. In particular, inspired by Ref. [17], where a sinusoidal formulas is used to overcome the shortcomings of linear settings and improve the robustness of DE, we a) incorporate a cosine perturbation into the probability parameter setting, and develop a stochastic mixed mutation strategy by randomly selecting a suitable mutation operator with this probability; b) design an information intercrossing and sharing mechanism to promote the information dissemination between different subpopulations by dividing the population into superior and inferior subpopulations and exchanging or sharing their information with the opposite operation and binomial crossover operation respectively; c) employ a simple and efficient method to adjust the control parameters. Compared with the existing DE variants in Refs. [11,22–32],

1) in the proposed stochastic mixed mutation strategy, the probability parameter is only related to the number of iteration, and a cosine perturbation is incorporated into the probability parameter setting such that the mutation strategy with more exploration or exploitation could have more possibility to be chosen at the later or earlier evolutionary stage. Then the proposed mutation strategy does not require more store spaces or computational costs (see Refs. [22–27]) and could adapt different evolutionary stages and complex problems (see Refs. [11,28]).
2) in the proposed information intercrossing and sharing mechanism, the population is divided into the superior and inferior subpopulations according to their fitness values, and the opposite operation is employed to share the information of individual in the superior subpopulation. Furthermore, the binomial crossover operation is used to exchange the information of individual in the superior subpopulation with the individual randomly generated from search space or the best individual according to its fitness value. Then this mechanism promotes the information dissemination between different subpopulations (see Refs. [29–32]).

Therefore, ISDE could not only enhance the performance to avoid premature convergence, but also make good use of the information of individuals during the evolutionary process. Finally, numerical experiments are carried out to evaluate the performance of ISDE by comparing with twelve typical algorithms on 55 test functions from both CEC2005 [42] and CEC2014 [43]. The experimental results show

Table 3
Experimental results of ISDE with various values for β with $D = 30$.

Function	$\beta = 0.1$	$\beta = 0.3$	$\beta = 0.5$	$\beta = 0.7$	$\beta = 0.9$
	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank
f_1	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**
f_2	8.16E-26(4.26E-26)/3	9.55E-27(5.06E-27)/2	**9.08E-27(1.06E-26)/1**	9.13E-26(2.40E-25)/4	2.91E-23(7.46E-23)/5
f_3	**1.49E+04(6.82E+03)/1**	1.66E+04(9.27E+03)/2	5.02E+04(2.36E+04)/5	2.82E+04(1.84E+04)/4	2.56E+04(1.80E+04)/3
f_4	1.16E-02(4.26E-02)/2	5.39E-04(2.23E-03)/5	**1.64E-05(4.08E-05)/1**	1.39E-04(5.36E-04)/3	1.66E-04(3.20E-04)/4
f_5	4.11E+02(3.39E+02)/4	**3.18E+02(3.74E+02)/1**	9.33E+02(4.10E+02)/5	3.25E+02(3.08E+02)/2	3.97E+02(3.35E+02)/3
f_6	1.14E+00(4.92E+00)/3	1.04E+00(4.44E+00)/2	**1.59E-01(7.97E-01)/1**	6.66E+00(1.71E+01)/4	8.35E+00(1.05E+01)/5
f_7	7.78E-03(8.62E-03)/4	7.78E-03(1.12E-02)/4	**1.49E-03(8.24E-03)/1**	5.32E-03(6.38E-03)/2	5.71E-03(1.10E-02)/3
f_8	2.09E+01(1.12E-01)/2	2.10E+01(4.10E-02)/5	**2.08E+01(1.36E-01)/1**	2.09E+01(6.48E-02)/2	2.09E+01(5.01E-02)/2
f_9	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**
f_{10}	3.65E+01(6.27E+00)/4	3.03E+01(8.19E+00)/2	**2.57E+01(6.53E+00)/1**	3.12E+01(4.73E+00)/3	3.95E+01(3.98E+00)/5
f_{11}	2.10E+01(4.03E+00)/5	1.73E+01(3.44E+00)/4	**1.21E+01(3.38E+00)/1**	1.49E+01(4.44E+00)/3	1.37E+01(4.07E+00)/2
f_{12}	1.61E+03(1.71E+03)/5	1.10E+03(1.55E+03)/3	8.74E+02(1.22E+03)/2	**5.82E+02(7.34E+03)/1**	1.37E+03(1.85E+03)/4
f_{13}	**1.26E+00(1.94E-01)/1**	1.30E+00(1.19E-01)/2	1.36E+00(1.17E-01)/4	1.33E+00(1.46E-01)/3	1.38E+00(1.17E-01)/5
f_{14}	1.23E+01(4.36E-01)/3	**1.22E+01(3.73E-01)/1**	**1.22E+01(4.35E-01)/1**	1.25E+01(3.34E-01)/5	1.24E+01(3.54E-01)/4
Sum Rank	41	35	26	38	47
Aver Rank	2.93	2.5	1.86	2.71	3.36

Table 4
Experimental results of ISDE with various values for γ with $D = 30$.

Function	$\gamma = 0.1$	$\gamma = 0.3$	$\gamma = 0.5$	$\gamma = 0.7$	$\gamma = 0.9$
	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank	Mean Error(Std Dev)/Rank
f_1	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**
f_2	8.04E-26(5.14E-26)/5	1.48E-26(1.15E-26)/4	**9.08E-27(1.06E-26)/1**	9.28E-27(1.02E-26)/2	9.83E-27(4.91E-26)/3
f_3	**2.01E+04(9.29E+03)/1**	2.62E+04(1.56E+04)/4	5.02E+04(2.36E+04)/5	2.06E+04(1.49E+04)/2	2.20E+04(1.17E+04)/3
f_4	4.63E-03(2.24E-02)/5	2.69E-04(6.47E-04)/3	**1.64E-05(4.08E-05)/1**	1.62E-04(3.03E-04)/2	6.65E-04(2.08E-03)/4
f_5	**2.66E+02(1.96E+02)/1**	4.50E+02(3.20E+02)/4	9.33E+02(4.10E+02)/5	4.08E+02(3.26E+02)/2	4.09E+02(3.61E+02)/3
f_6	1.79E+01(6.73E+00)/2	1.91E+01(1.05E+00)/3	**1.59E-01(7.97E-01)/1**	3.63E+01(2.67E+01)/5	3.19E+01(2.73E+01)/4
f_7	1.68E-03(3.03E-03)/2	2.56E-03(5.88E-03)/3	**1.49E-03(8.24E-03)/1**	5.81E-03(7.69E-03)/4	8.86E-03(1.05E-02)/5
f_8	2.10E+01(4.36E-02)/2	2.10E+01(5.55E-02)/2	**2.08E+01(1.36E-01)/1**	2.10E+01(4.68E-02)/2	2.10E+01(4.55E-02)/2
f_9	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**	**0.00E+00(0.00E+00)/1**
f_{10}	3.20E+01(8.27E+00)/3	3.00E+01(6.91E+00)/2	**2.57E+01(6.53E+00)/1**	3.48E+01(7.12E+00)/4	3.71E+01(1.05E+01)/5
f_{11}	1.92E+01(4.58E+00)/5	1.78E+01(4.02E+00)/3	**1.21E+01(3.38E+00)/1**	1.79E+01(4.23E+00)/4	1.44E+01(3.30E+00)/2
f_{12}	1.35E+03(1.38E+03)/2	2.11E+03(2.24E+03)/4	**8.74E+02(1.22E+03)/1**	1.74E+03(2.04E+03)/3	3.23E+03(4.55E+03)/5
f_{13}	1.38E+00(1.85E-01)/3	1.44E+00(2.34E-01)/4	1.36E+00(1.17E-01)/2	**1.35E+00(1.91E-01)/1**	1.64E+00(1.28E-01)/5
f_{14}	1.25E+01(4.49E-01)/2	1.25E+01(3.88E-01)/2	**1.22E+01(4.35E-01)/1**	1.25E+01(3.49E-01)/2	1.25E+01(3.04E-01)/2
Sum Rank	37	40	23	35	45
Aver Rank	2.64	2.86	1.64	2.50	3.21

that the proposed algorithm is very competitive. Furthermore, ISDE is applied to Parameter Estimation for Frequency-Modulated Sound Waves.

The reminder of this paper is organized as follows. Section 2 introduces briefly the classical DE algorithm. The proposed algorithm is presented in Section 3. Section 4 provides and discusses the experimental results. Finally, conclusions are drawn in Section 5.

2. Classical DE algorithm

In this section, we consider the minimization problem $\min\{f(\vec{x}) \mid a_j \le x_j \le b_j, j = 1, 2, \ldots, D\}$, where $\vec{x} = (x_1, x_2, \ldots, x_D)$ represents the decision vector, D is the dimension of the decision space, a_j and b_j are the lower and upper bounds of the j-th component of decision space, respectively. In particular, the main steps of DE algorithm are described as follows.

2.1. Initialization

At the beginning of DE algorithm, an initial population X^0 with NP individuals is randomly generated in the decision space, where NP is the size of population. For the i-th individual $\vec{x}_i^0 = (x_{i,1}^0, x_{i,2}^0, \ldots, x_{i,D}^0)$, its j-th component is generated as follow:

$$x_{i,j}^0 = a_j + rand(0,1) \cdot (b_j - a_j) \tag{1}$$

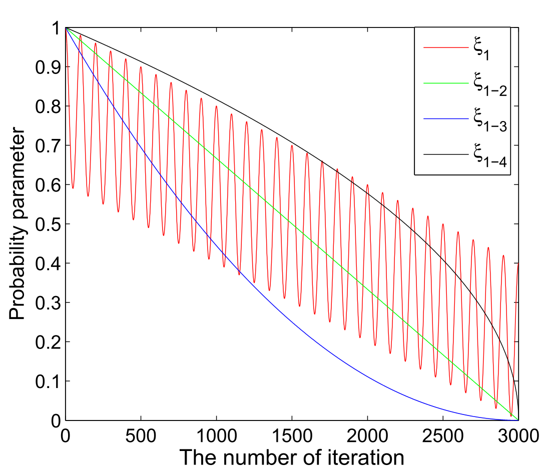


Fig. 2. Curves of strategy chosen probability parameters.

for $i = 1, 2, \ldots, NP$ and $j = 1, 2, \ldots, D$, where $rand(0, 1)$ returns a uniformly distributed random variable within the range [0, 1].

2.2. Mutation

After the initialization of population, the mutation operation will be performed for each individual to generate its mutant individual. In particular, for the i-th individual $\vec{x}_i^g$ at generation g, five mutation strategies below are commonly used to create the mutant vector $\vec{v}_i^g$:

$$DE/best/1: \quad \vec{v}_i^g = \vec{x}_{best}^g + F \cdot (\vec{x}_{r1}^g - \vec{x}_{r2}^g), \tag{2}$$

$$DE/best/2: \quad \vec{v}_i^g = \vec{x}_{best}^g + F \cdot (\vec{x}_{r1}^g - \vec{x}_{r2}^g) + F \cdot (\vec{x}_{r3}^g - \vec{x}_{r4}^g), \tag{3}$$

$$DE/rand/1: \quad \vec{v}_i^g = \vec{x}_{r1}^g + F \cdot (\vec{x}_{r2}^g - \vec{x}_{r3}^g), \tag{4}$$

$$DE/rand/2: \quad \vec{v}_i^g = \vec{x}_{r1}^g + F \cdot \left(\vec{x}_{r2}^g - \vec{x}_{r3}^g\right) + F \cdot \left(\vec{x}_{r4}^g - \vec{x}_{r5}^g\right) \tag{5}$$

and

$$DE/current-to-best/1: \quad \vec{v}_i^g = \vec{x}_i^g + F \cdot \left(\vec{x}_{best}^g - \vec{x}_i^g\right) + F \cdot \left(\vec{x}_{r1}^g - \vec{x}_{r2}^g\right), \tag{6}$$

where r_1, r_2, r_3, r_4 and r_5 are the distinct integers randomly generated from $[1, NP]$ and not equal to i, $\vec{x}_{best}^g$ represents the individual with the best fitness value among X^g, and F is a scaling factor.

2.3. Crossover

For the target individual $\vec{x}_i^g$ and its corresponding mutant individual $\vec{v}_i^g$, the crossover operation is then performed to generate the trial individual $\vec{u}_i^g = (u_{i,1}^g, u_{i,2}^g, \ldots, u_{i,D}^g)$. Specially, the binomial crossover method is described as:

$$u_{i,j}^g = \begin{cases} v_{i,j}^g, & \text{if } rand \le Cr \text{ or } j = randn(i), \\ x_{i,j}^g, & \text{otherwise} \end{cases} \tag{7}$$

for $j = 1, 2, \ldots, D$, where $Cr \in [0, 1]$ is crossover rate, and $randn(i) \in \{1, 2, \ldots, D\}$ is a random index to ensure that $\vec{u}_i^g$ has at least one component from $\vec{v}_i^g$.

Table 5
Experimental results of ISDE with four probability rules with $D = 30$.

Function	ISDE-ξ_1	ISDE-ξ_{1-2}	ISDE-ξ_{1-3}	ISDE-ξ_{1-4}
	Mean Error (Std Dev)/Rank	Mean Error (Std Dev)/Rank	Mean Error (Std Dev)/Rank	Mean Error (Std Dev)/Rank
f_6	**1.59E-01 (7.97E-01)/1**	5.38E-01 (1.40 + 00)/2	9.26E-01 (1.90E+00)/3	1.24E+00 (2.62E+00)/4
f_7	1.49E-03 (8.24E-03)/3	**1.36E-03 (3.37E-03)/1**	2.76E-02 (1.65E-02)/4	1.42E-03 (1.45E-03)/2
f_8	2.08E+01 (1.36E-01)/1	2.09E+01 (8.87E-02)/3	2.09E+01 (5.96E-02)/2	2.09E+01 (1.30E-01)/4
f_9	**0.00E+00 (0.00E+00)/1**	1.99E-01 (4.12E-01)/3	1.33E-01 (3.50E-01)/2	4.38E-01 (5.80E-01)/4
f_{10}	2.57E+01 (6.53E+00)/2	**2.38E+01 (1.27E+00)/1**	3.82E+01 (1.13E+01)/4	3.76E+01 (1.13E+01)/3
f_{11}	**1.21E+01 (3.38E+00)/1**	1.40E+01 (3.92E+00)/2	1.46E+01 (6.37E+00)/3	1.50E+01 (6.30E+00)/4
f_{12}	**8.74E+02 (1.22E+03)/1**	3.40E+03 (5.66E+03)/4	3.10E+03 (5.13E+03)/2	3.18E+03 (4.65E+03)/3
f_{13}	**1.36E+00 (1.17E-01)/1**	2.16E+00 (2.96E-01)/4	2.10E+00 (3.07E-01)/2	2.16E+00 (2.84E-01)/3
f_{14}	**1.22E+01 (4.35E-01)/1**	1.23E+01 (4.52E-01)/2	1.24E+01 (3.94E-01)/3	1.23E+01 (4.71E-01)/3
Sum Rank	12	22	25	30
Aver Rank	1.33	2.44	2.78	3.33

2.4. Selection

Finally, the selection operation is used to generate the next population, and the greedy selection strategy [1] and tournament selection strategy [37] are the two commonly used operators. In particular, for the objective individual $\vec{x}_i^g$ and its offspring $\vec{u}_i^g$, the greedy selection strategy is

$$\vec{x}_i^{g+1} = \begin{cases} \vec{u}_i^g, & \text{if } f\left(\vec{u}_i^g\right) \le f\left(\vec{x}_i^g\right) \\ \vec{x}_i^g, & \text{otherwise.} \end{cases} \tag{8}$$

Note that DE with (8) will either get better or remain the same fitness status, but never deteriorates. Moreover, the mutation, crossover and selection operators will still execute in turn until the termination crite-

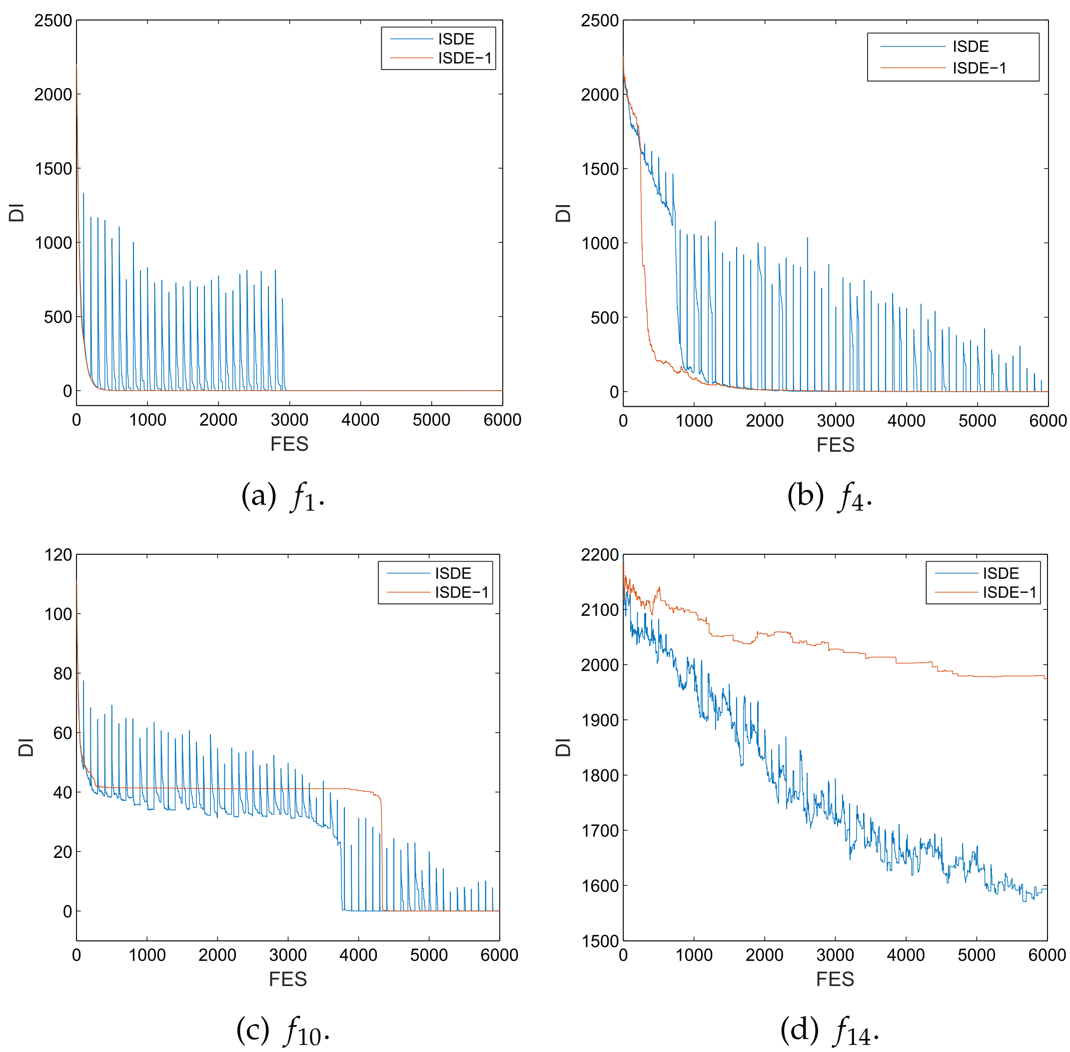
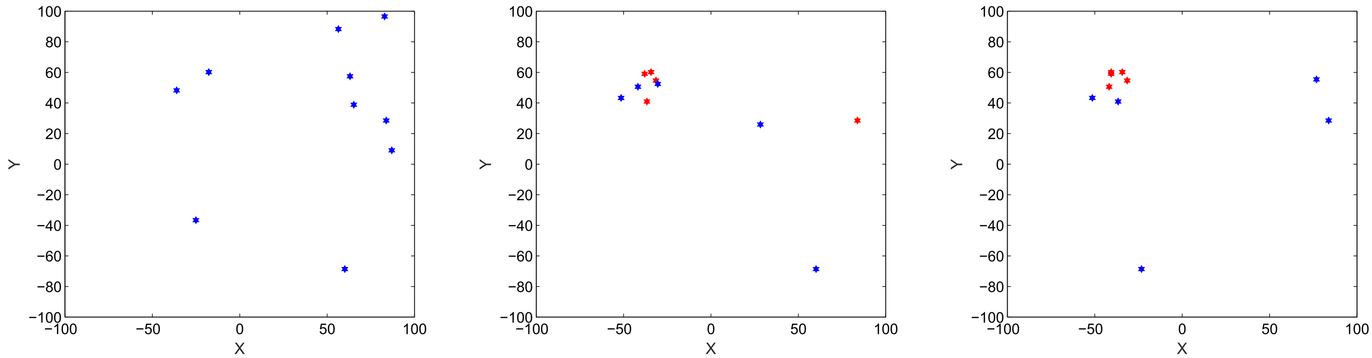


Fig. 3. DI curves of ISDE and ISDE-1. (a) f_1, (b) f_4, (c) f_{10} and (d) f_{14}.



(a) $G = 0$. (b) $G = 19$. (c) $G = 20$.

Fig. 4. Distribution of population on f_1 with $D = 2$ and $NP = 10$. (a) The initialization population. (b) Before the IS mechanism. (c) The IS mechanism.

Table 6
Experimental results of ISDE and ISDE-1 with $D = 30$.

Function	f_1 Mean Error(Std Dev)	f_2 Mean Error(Std Dev)	f_3 Mean Error(Std Dev)	f_4 Mean Error(Std Dev)	f_5 Mean Error(Std Dev)
ISDE-1	**0.00E+00 (0.00E+00)**	**5.56E-27 (4.74E-27)**	**1.92E+04 (1.05E+04)**	1.34E-03 (3.41E-03)	**3.02E+02 (2.96E+02)**
ISDE	**0.00E+00(0.00E+00)**	9.08E-27(1.06E-26)	5.02E+04(2.36E+04)	**1.64E-05(4.08E-05)**	9.33E+02(4.10E+02)
Function	f_6 Mean Error(Std Dev)	f_7 Mean Error(Std Dev)	f_8 Mean Error(Std Dev)	f_9 Mean Error(Std Dev)	f_{10} Mean Error(Std Dev)
ISDE-1	1.44E+00 (1.95E+00)	1.97E-02 (1.26E-02)	2.09E+01 (5.55E-02)	**0.00E+00 (0.00E+00)**	4.26E+01 (1.92E+01)
ISDE	**1.59E-01 (7.97E-01)**	**1.49E-03 (8.24E-03)**	**2.08E+01 (1.36E-01)**	**0.00E+00 (0.00E+00)**	**2.57E+01 (6.53E+00)**
Function	f_{11} Mean Error(Std Dev)	f_{12} Mean Error(Std Dev)	f_{13} Mean Error(Std Dev)	f_{14} Mean Error(Std Dev)	
ISDE-1	2.08E+01 (7.16E+00)	1.96E+03 (2.53E+03)	1.55E+00 (1.20E-01)	1.29E+01 (4.42E-01)	
ISDE	**1.21E+01 (3.38E+00)**	**8.74E+02 (1.22E+03)**	**1.36E+00 (1.17E-01)**	**1.22E+01 (4.35E-01)**	

rion is satisfied.

3. Proposed algorithm

In this section, we shall propose a DE by designing a stochastic mixed mutation strategy and an information intercrossing and sharing mechanism.

3.1. Stochastic mixed mutation strategy

As pointed out in Ref. [6], the performance of DE depends mainly on the balance between the exploration and exploitation. Then, it is important to design a suitable mutation operator for different problems and evolutionary stages. In particular, Mohamed [11] developed a novel triangular mutation operator to enhance the exploitation ability of algorithm, and proposed a combined mutation strategy by randomly choosing a suitable operator from $DE/rand/1$ and the triangular mutation operator with a monotonically decreasing probability, while Xiang *et al.* [28] presented a new combined mutation strategy composed of two mutation strategies, where a linear time-varying function is used to decide which mutation strategy is chosen at the mutation phases. Obviously, they could balance effectively the exploration and exploitation during the evolutionary process. However, once the algorithms in Refs. [11,28] get stuck at local optimum during the later evolutionary process, it will be difficult to jump out since the global search ability has been weakened even when one operation similar to the restart mechanism is used to enhance the diversity of population.

To mitigate the difficulty above, we propose a stochastic mixed mutation strategy by using “DE/current-to-pbest/1” and “DE/pbest/1”

Table 7
Parameters setting.

Algorithms	Parameter setting
jDE [40]	$NP = 100$, $\tau_1 = \tau_2 = 0.1$, $F_l = 0.1$, $F_u = 0.9$
SaDE [23]	$NP = 50$, $K = 4$, $Lp = 50$
EPSDE [24]	$NP = 50$, $F \in [0.4,\ 0.9]$ and $CR \in [0.1,\ 0.9]$ with stepsize = 0.1
JADE [14]	$NP = 100$, $\mu F_0 = \mu Cr_0 = 0.5$, $c = 0.1$, $p = 0.05$
CoDE [22]	$NP = 30$, $[F = 1.0,\ CR = 0.1]$, $[F = 1.0,\ CR = 0.9]$, $[F = 0.8,\ CR = 0.2]$
SHADE [21]	$NP = 100$, $H = 2$, $c = 0.1$, $p = rand(0.02, 0.2)$
MPEDE [32]	$NP = 250$, $c = 0.1$, $\lambda_1 = \lambda_2 = \lambda_3 = 0.2$, $ng = 20$
SinDE [17]	$NP = 40$, $freq = 0.25$
CIPDE [18]	$NP = 100$, $c = 0.1$, $\mu_F = 0.7$, $\mu_{CR} = 0.5$, $T = 90$
CLPSO [7]	$NP = 30$, $c_1 = c_2 = 1.494$, $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, $m = 5$
CMA-ES [44]	$NP = 4 + \lfloor 3\ \ln(D) \rfloor$, $\mu = \lfloor NP/2 \rfloor$, $\omega_i = \ln((NP+1)/2) - \ln(i)(i = 1, 2, \cdots, \mu)$, $C_c = C_\sigma = 4/(D+4)$
GL-25 [8]	$NP = 60$, $\alpha = 1$, $\omega = 5$, $n_T = 2$
ISDE	$NP = 50$, $K = 100$, $\alpha = 0.6$, $\beta = \gamma = 0.5$, $freq = 0.01$

Table 8

Experimental results of ISDE and ten existing algorithms on CEC 2005 contest test instances with $D = 30$.

Func	Statistic	jDE	SaDE	EPSDE	JADE	CoDE	SHADE	MPEDE	CLPSO	CMA-ES	GL-25	ISDE
f_1	Mean Error	**0.00E+00≈**	**0.00E+00≈**	**0.00E+00≈**	**0.00E+00≈**	**0.00E+00≈**	**0.00E+00≈**	**0.00E+00≈**	**0.00E+00≈**	1.58E-25+	5.60E-27+	**0.00E+00**
	Std Dev (Rank)	**0.00E+00 (1)**	**0.00E+00 (1)**	**0.00E+00 (1)**	**0.00E+00 (1)**	**0.00E+00 (1)**	**0.00E+00 (1)**	**0.00E+00 (1)**	**0.00E+00 (1)**	3.35E-26 (8)	1.76E-26 (7)	**0.00E+00 (1)**
f_2	Mean Error	1.11E-06+	8.26E-06+	4.23E-26+	1.26E-28-	1.69E-15+	**5.97E-29-**	1.01E-26+	8.40E+02+	1.12E-24+	4.04E+01+	9.08E-27
	Std Dev (Rank)	1.96E-06 (8)	1.65E-05 (9)	4.07E-26 (5)	1.22E-28(2)	3.95E-15(7)	**8.31E-29(1)**	2.05E-26 (4)	1.90E+02 (11)	2.93E-25 (6)	6.28E+01 (10)	1.06E-26 (3)
f_3	Mean Error	1.98E+05+	4.27E+05+	8.74E+05+	8.42E+03-	1.05E+05+	6.87E+03-	1.01E+01-	1.42E+07+	**5.54E-21-**	2.19E+06+	5.02E+04
	Std Dev (Rank)	1.10E+05 (7)	2.08E+05 (8)	3.28E+06 (9)	6.58E+03(4)	6.25E+04(6)	5.93E+03(3)	8.32E+00 (2)	4.19E+06 (11)	**1.69E-21 (1)**	1.08E+06 (10)	2.36E+04 (5)
f_4	Mean Error	4.40E-02+	1.77E+02+	3.49E+02+	4.13E-16-	5.81E-03+	**2.71E-16-**	6.61E-16-	6.99E+03+	9.15E+05+	9.07E+02+	1.64E-05
	Std Dev (Rank)	1.26E-01 (6)	2.67E+02 (7)	2.23E+03 (8)	3.45E-16(2)	1.38E-02(5)	**9.37E-16(1)**	5.68E-16 (3)	1.73E+03 (10)	2.16E+06 (11)	4.25E+02 (9)	4.08E-05 (4)
f_5	Mean Error	5.11E+02-	3.25E+03+	1.40E+03+	7.59E-08-	3.31E+02-	9.70E-10-	7.21E-06-	3.86E+03+	**2.77E-10-**	2.51E+03+	9.33E+02
	Std Dev (Rank)	4.40E+02 (6)	5.90E+02 (10)	7.12E+02 (8)	5.65E-07(3)	3.44E+02(5)	2.87E-09(2)	5.12E-06 (4)	4.35E+02 (11)	**5.04E-11 (1)**	1.96E+02 (9)	4.10E+02 (7)
f_6	Mean Error	2.35E+01+	5.31E+01+	6.38E-01+	1.16E+01+	1.60E-01+	1.28E+00+	9.65E+00+	4.16E+00+	4.78E-01+	2.15E+01+	**1.59E-01**
	Std Dev (Rank)	2.50E+01 (10)	3.25E+01 (11)	1.49E+00 (4)	3.16E+01(8)	7.85E-01(2)	1.90E+00(5)	4.65E+00 (7)	3.48E+00 (6)	1.32E+00 (3)	1.17E+00 (9)	**7.97E-01 (1)**
f_7	Mean Error	1.18E-02+	1.57E-02+	1.77E-02+	8.27E-03+	7.46E-03+	5.90E-03+	2.36E-03+	4.51E-01+	1.82E-03+	2.78E-02+	**1.49E-03**
	Std Dev (Rank)	7.78E-03 (7)	1.38E-02 (8)	1.34E-02 (9)	8.22E-03(6)	8.55E-03(5)	9.25E-03(4)	1.15E-03 (3)	8.47E-02 (11)	4.33E-03 (2)	3.62E-02 (10)	**8.24E-03 (1)**
f_8	Mean Error	2.09E+01+	2.09E+01+	2.09E+01+	2.09E+01+	**2.01E+01-**	2.04E+01-	2.09E+01+	2.09E+01+	2.03E+01-	2.09E+01+	2.08E+01
	Std Dev (Rank)	4.86E-02 (5)	4.95E-02 (5)	5.81E-02 (5)	1.68E-01(5)	**1.41E-01(1)**	4.15E-01(3)	5.87E-01 (5)	4.41E-02 (5)	5.72E-01 (2)	5.94E-02 (5)	1.36E-01 (4)
f_9	Mean Error	**0.00E+00≈**	2.39E-01+	3.98E-02+	**0.00E+00≈**	**0.00E+00≈**	**0.00E+00≈**	**0.00E+00≈**	**0.00E+00≈**	4.45E+02+	2.45E+01+	**0.00E+00**
	Std Dev (Rank)	**0.00E+00 (1)**	4.33E-01 (9)	1.99E-01 (8)	**0.00E+00(1)**	**0.00E+00(1)**	**0.00E+00(1)**	**0.00E+00 (1)**	**0.00E+00 (1)**	7.12E+01 (11)	7.35E+00 (10)	**0.00E+00 (1)**
f_{10}	Mean Error	5.54E+01+	4.72E+01+	5.36E+01+	2.42E+01-	4.15E+01+	2.17E+01-	**1.52E+01-**	1.04E+02+	4.63E+01+	1.42E+02+	2.57E+01
	Std Dev (Rank)	8.46E+00 (9)	1.01E+01 (7)	3.03E+01 (8)	5.44E+00(3)	1.16E+01(5)	7.17E+00(2)	**2.98E+00 (1)**	1.53E+01 (10)	1.16E+01 (6)	6.45E+01 (11)	6.53E+00 (4)
f_{11}	Mean Error	2.79E+01+	1.65E+01+	3.56E+01+	2.57E+01+	1.18E+01-	2.55E+01+	2.58E+01+	2.60E+01+	**7.11E+00-**	3.27E+01+	1.21E+01
	Std Dev (Rank)	1.61E+00 (9)	2.42E+00 (4)	3.88E+00 (11)	2.21E+00(6)	3.40E+00(2)	1.48E+00(5)	3.11E+00 (7)	1.63E+00 (8)	**2.14E+00 (1)**	7.79E+00 (10)	3.38E+00 (3)
f_{12}	Mean Error	8.63E+03+	3.02E+03+	3.58E+04+	6.45E+03+	3.05E+03+	8.15E+03+	1.17E+03+	1.79E+04+	1.26E+04+	6.53E+04+	**8.74E+02**
	Std Dev (Rank)	8.31E+03 (7)	2.33E+03 (3)	7.05E+03 (10)	2.89E+03(5)	3.80E+03(4)	4.66E+03(6)	8.66E+02 (2)	5.24E+03 (9)	1.74E+04 (8)	4.69E+04 (11)	**1.22E+03 (1)**
f_{13}	Mean Error	1.66E+00+	3.94E+00+	1.94E+00+	1.47E+00+	1.57E+00+	1.38E+00+	2.92E+00+	2.06E+00+	3.43E+00+	6.23E+00+	**1.36E+00**
	Std Dev (Rank)	1.35E-01 (5)	2.81E-01 (10)	1.46E-01 (6)	1.15E-01(3)	3.27E+01(4)	9.28E-02(2)	6.33E-01 (8)	2.15E-01 (7)	7.60E-01 (9)	4.88E+00 (11)	**1.17E-01 (1)**
f_{14}	Mean Error	1.30E+01+	1.26E+01+	1.35E+01+	1.23E+01+	1.23E+01+	1.25E+01+	1.23E+01+	1.28E+01+	1.47E+01+	1.31E+01+	**1.22E+01**
	Std Dev (Rank)	2.00E-01 (8)	2.83E-01 (6)	2.09E-01 (10)	3.21E-01(2)	4.81E-01(2)	2.50E-01(5)	4.22E-01 (2)	2.48E-01 (7)	3.31E-01 (11)	1.84E-01 (9)	**4.35E-01 (1)**
f_{15}	Mean Error	3.77E+02+	3.76E+02+	2.12E+02-	3.61E+02+	3.88E+02+	3.44E+02+	3.78E+02+	**5.77E+01-**	5.55E+02+	3.04E+02-	3.36E+02
	Std Dev (Rank)	8.02E+01 (8)	7.83E+01 (7)	1.98E+01 (2)	2.24E+02(6)	6.85E+01(10)	9.14E+01(5)	6.32E+01 (9)	**2.76E+01 (1)**	3.32E+02 (11)	1.99E+01 (3)	9.52E+01 (4)
f_{16}	Mean Error	7.94E+01+	8.57E+01+	1.22E+02+	9.33E+02+	7.37E+01+	1.07E+02+	**3.77E+01-**	1.74E+02+	2.98E+02+	1.32E+02+	4.66E+01
	Std Dev (Rank)	2.96E+01 (4)	6.94E+01 (5)	9.19E+01 (7)	1.31E+02(11)	5.13E+01(3)	1.32E+02(6)	**5.22E+00 (1)**	2.82E+01 (9)	2.08E+02 (10)	7.60E+01 (8)	6.44E+00 (2)
f_{17}	Mean Error	1.37E+02+	7.83E+01+	1.69E+02+	1.21E+02+	6.67E+01-	1.52E+02+	**4.36E+01-**	2.46E+02+	4.43E+02+	1.61E+02+	7.46E+01
	Std Dev (Rank)	3.80E+01 (6)	3.76E+01 (4)	1.02E+02 (9)	1.08E+02(5)	2.12E+01(2)	1.33E+02(7)	**6.35E+00 (1)**	4.81E+01 (11)	3.34E+02 (10)	6.80E+01 (8)	8.00E+01 (3)
f_{18}	Mean Error	9.04E+02≈	8.68E+02≈	**8.20E+02-**	9.04E+02≈	9.04E+02≈	9.05E+02+	9.04E+02≈	9.13E+02+	9.04E+02≈	9.07E+02+	9.04E+02
	Std Dev (Rank)	1.08E+01 (2)	6.23E+01 (2)	**3.35E+00 (1)**	1.24E+00(2)	1.04E+00(2)	1.27E+00(9)	1.21E+00 (2)	1.42E+00 (11)	3.01E-01(2)	1.48E+00 (10)	3.73E-01 (2)
f_{19}	Mean Error	9.04E+02≈	8.74E+02≈	**8.21E+02-**	9.04E+02≈	9.04E+02≈	9.05E+02+	9.04E+02≈	9.14E+02+	9.16E+02+	9.06E+02+	9.04E+02
	Std Dev (Rank)	1.11E+00 (2)	6.22E+01 (2)	**3.35E+00 (1)**	8.23E+00(2)	9.42E-01(2)	1.42E+00(8)	1.24E+00 (2)	1.45E+00 (10)	6.03E+01 (11)	1.24E+00 (9)	2.92E-01 (2)
f_{20}	Mean Error	9.04E+02≈	8.78E+02≈	**8.22E+02-**	9.04E+02≈	9.04E+02≈	9.04E+02≈	9.04E+02≈	9.14E+02+	9.04E+02≈	9.07E+02+	9.04E+02
	Std Dev (Rank)	1.10E+00 (2)	6.03E+01 (2)	**4.17E+00 (1)**	7.80E-01(2)	9.01E-01(2)	8.91E-01(2)	1.18E+00 (2)	3.62E+00 (11)	2.71E-01 (2)	1.35E+00 (10)	3.30E-01 (2)

(*continued on next page*)

Table 8 *(continued)*

Func	Statistic	jDE	SaDE	EPSDE	JADE	CoDE	SHADE	MPEDE	CLPSO	CMA-ES	GL-25	ISDE
f_{21}	Mean Error	**5.00E+02≈**	5.52E+02+	8.33E+02+	**5.00E+02≈**	**5.00E+02≈**	**5.00E+02≈**	**5.00E+02≈**	**5.00E+02≈**	**5.00E+02≈**	**5.00E+02≈**	**5.00E+02**
	Std Dev (Rank)	**4.80E-13 (1)**	1.82E+02 (10)	1.00E+02 (11)	**4.67E-13(1)**	**4.88E-13(1)**	**1.31E-13(1)**	**3.54E-14 (1)**	**3.39E-13 (1)**	**2.68E-12 (1)**	**4.83E-13 (1)**	**1.01E-13 (1)**
f_{22}	Mean Error	8.75E+02+	9.36E+02+	**5.07E+02-**	8.68E+02+	8.63E+02-	8.65E+02-	8.72E+02+	9.72E+02+	8.26E+02-	9.28E+02+	8.67E+02
	Std Dev (Rank)	1.91E+01 (8)	1.83E+01 (10)	**7.26E+00 (1)**	2.24E+01(6)	2.43E+01(3)	1.89E+01(4)	2.98E+00 (7)	1.20E+01 (11)	1.46E+01 (2)	7.04E+01 (9)	2.23E+01 (5)
f_{23}	Mean Error	**5.34E+02≈**	**5.34E+02≈**	8.58E+02+	5.48E+02+	**5.34E+02≈**	5.35E+02+	**5.34E+02≈**	**5.34E+02≈**	5.36E+02+	**5.34E+02≈**	**5.34E+02**
	Std Dev (Rank)	**2.77E-04(1)**	**3.57E-03 (1)**	6.82E+01 (11)	8.62E+01(10)	**4.12E-04(1)**	2.08E+00(8)	**3.87E-04 (1)**	**2.19E-04 (1)**	5.44E+00 (9)	**4.66E-04 (1)**	**5.19E-13 (1)**
f_{24}	Mean Error	**2.00E+02≈**	**2.00E+02≈**	2.13E+02+	**2.00E+02≈**	**2.00E+02≈**	**2.00E+00≈**	**2.00E+02≈**	**2.00E+02≈**	2.12E+02+	**2.00E+02≈**	**2.00E+02**
	Std Dev (Rank)	**2.85E-14 (1)**	**6.20E-13 (1)**	1.52E+00 (11)	**2.12E-14(1)**	**2.85E-14(1)**	**6.21E-13(1)**	**2.21E-14 (1)**	**1.49E-12 (1)**	6.00E+01 (10)	**5.52E-11 (1)**	**2.90E-14 (1)**
f_{25}	Mean Error	2.11E+02+	2.14E+02+	2.13E+02+	2.11E+02+	2.11E+02+	2.11E+02+	2.09E+02≈	**2.00E+02-**	2.07E+02-	2.17E+02+	2.09E+01
	Std Dev (Rank)	7.32E-01 (5)	2.00E+00 (10)	2.55E+00 (9)	7.35E-01(5)	9.02E-01(5)	1.15E+00(5)	3.32E-01 (3)	**1.96E+00 (1)**	6.07E+00 (2)	1.36E-01 (11)	3.73E-01 (3)
Sum Rank		129	152	166	97	105	97	80	176	140	202	63
Aver Rank		5.16	6.08	6.64	3.88	4.2	3.88	3.2	7.04	5.6	8.08	2.52
+		16	19	19	13	12	13	10	18	16	21	
−		1	0	5	5	5	7	6	2	6	1	
≈		8	6	1	7	8	5	9	5	3	3	

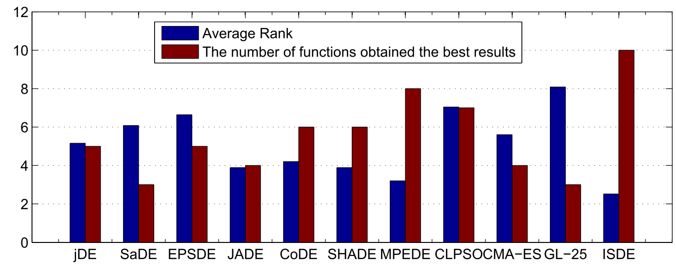


Fig. 5. Statistical results of ISDE and other compared algorithms on CEC 2005 with $D = 30$.

and incorporating the cosine perturbation into the probability parameter setting:

$$\vec{v}_i^g = \begin{cases} \vec{x}_i^g + F_1\left(\vec{x}_{pbest}^g - \vec{x}_i^g\right) + F_1\left(\vec{x}_{r1}^g - \vec{x}_{r2}^g\right), & \text{if } rand(0,1) < \xi_1 \\ \vec{x}_{pbest}^g + F_1\left(\vec{x}_{r1}^g - \vec{x}_{r2}^g\right), & \text{otherwise,} \end{cases} \tag{9}$$

where F_1 is scaling factor, r_1 and $r_2 \in [1, NP]$ are two random integers with $r1 \neq r2 \neq i$, ξ_1 is a probability parameter, and $\vec{x}_{pbest}^g$ represents the one randomly chosen from the top $\lceil p \cdot NP \rceil$ individuals with better fitness values among population.

Obviously, the first strategy “DE/current-to-pbest/1” in (9) takes the current individual as the base individual and searches the decision space along the superior individual, while the second strategy “DE/pbest/1” takes the superior individual as the base individual and searches the decision space around it. Then they have a stronger exploration or exploitation ability as the parameter p is set to large or small, and the former has the stronger global search ability than the latter. Thus, the parameter p plays an important role in the performance of (9). To effectively balance the exploration and exploitation of (9), we set

$$p = \beta(1 - \frac{g}{G}), \tag{10}$$

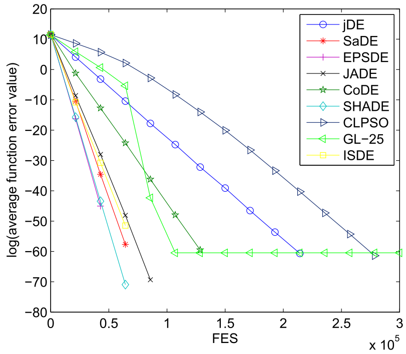
where G represents the maximin number of iterations and $\beta \in [0, 1]$. Clearly, the value of p decreases linearly as the iteration increases. Then, the proposed strategy could have a stronger exploration and exploitation during the earlier and later evolutionary stages, respectively. Specially, for β, a too large value could cause a large amount of invalid exploration during the earlier evolutionary process, while a too small value could reduce the global search ability of (9). Thus, let $\beta = 0.5$, which is shown to be a suitable choice by experiments in Subsection 4.1.

Moreover, to enhance the robustness and capability of (9) to jump out of local optimum during the later evolutionary process, a cosine perturbation is incorporated into the monotonically decreasing methods in Refs. [11,28] to set the probability parameter

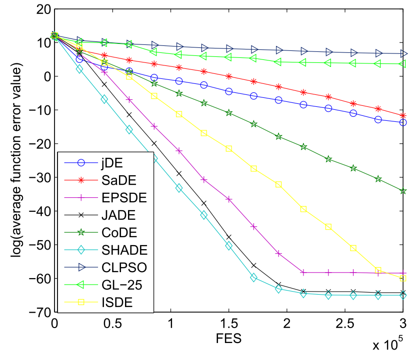
$$\xi_1 = \alpha(1 - \frac{g}{G}) + (1 - \alpha)\frac{1 + \cos(2\pi \cdot freq \cdot g)}{2}, \tag{11}$$

where $\alpha \in [0, 1]$ is a weight coefficient and $freq \in (0, 0.5)$ represents the frequency of the cosine perturbation. Obviously the value of α should not be too larger or smaller to ensure the convergence of (9) during the later evolutionary process and the validity of the cosine perturbation. From the experiments in Subsection 4.1, $\alpha = 0.6$ is a suitable choice.

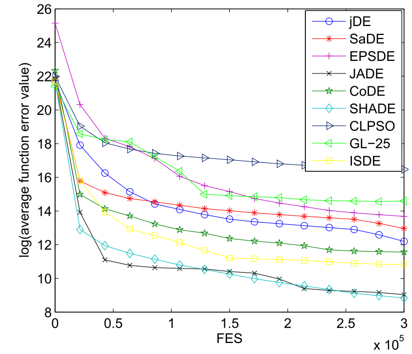
From (11), *freq* controls the property of the cosine perturbation during the evolutionary process, and its value close to 0 could lessen the oscillation of the cosine perturbation within many generations such that the explorative ability of algorithm would be reduced during the later evolutionary process. On the other hand, due to the periodicity of the cosine function, a value bigger than 0.1 could cause the severe concussion between adjacent iterations such that the exploitative ability of algorithm would be degraded since the mutation strategy with more exploration could be chosen with a larger probability. Specially, to see



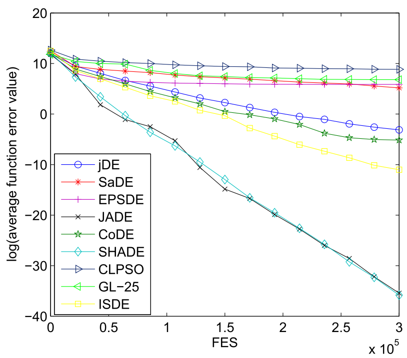
(a) f_1.



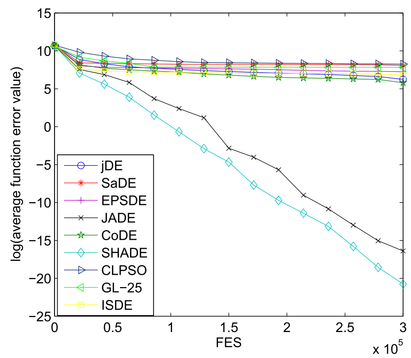
(b) f_2.



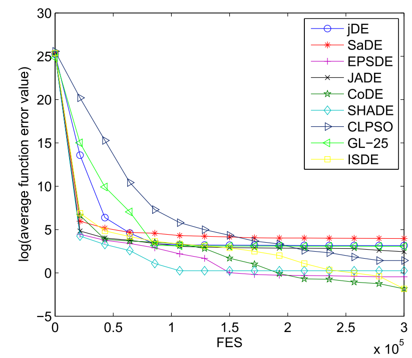
(c) f_3.



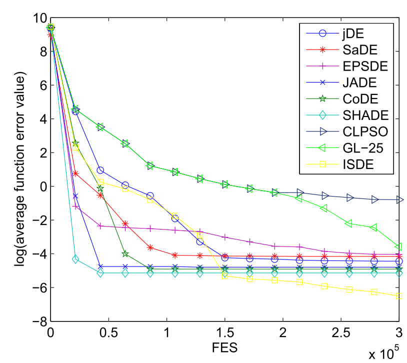
(d) f_4.



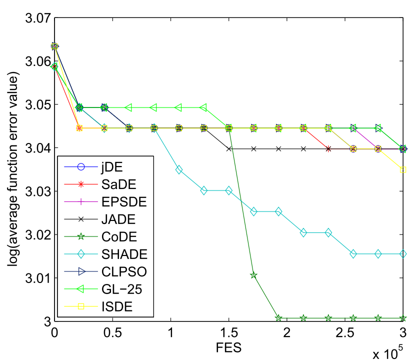
(e) f_5.



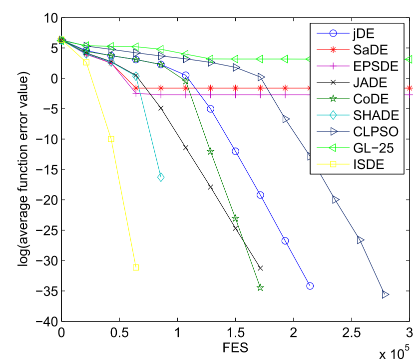
(f) f_6.



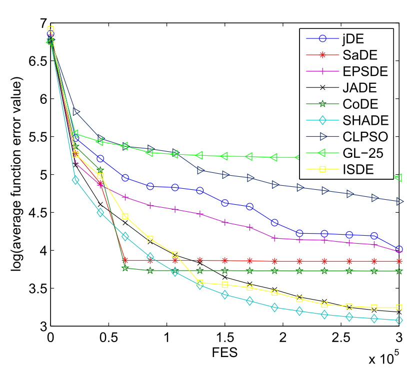
(g) f_7.



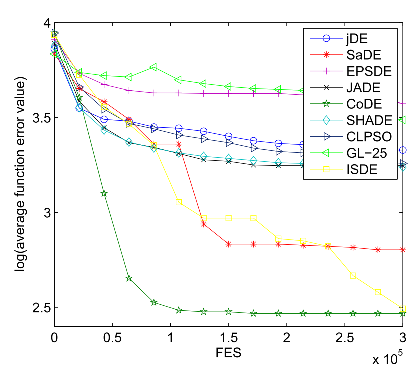
(h) f_8.



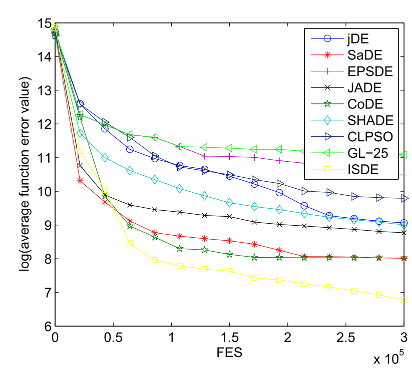
(i) f_9.



(j) f_{10}.



(k) f_{11}.



(l) f_{12}.

Fig. 6. Evolution curves of nine algorithms with $D = 30$. (a) f_1, (b) f_2, (c) f_3, (d) f_4, (e) f_5, (f) f_6, (g) f_7, (h) f_8, (i) f_9, (j) f_{10}, (k) f_{11} and (l) f_{12}.

clearly the influence of *freq*, Fig. 1 depicts the curves of ξ_1 with various *freq*. From Fig. 1, we see that a larger value for *freq* results in the severe concussion between adjacent iterations, and a smaller value causes a slower change of the cosine perturbation within many generations. Thus, $freq = 0.01$ is a suitable choice from Fig. 1 and the experiments in Subsection 4.1. In this case, ξ_1 could not only keep the change of the cosine perturbation within few iterations, but also have no severe concussion between adjacent iterations.

Therefore, the proposed strategy could effectively mitigate the difficulty to jump out of local optimum during the later evolutionary process since the cosine perturbation enhances the possibility to choose the mutation strategy with better exploration.

Overall, in the proposed strategy, the number of top individuals is decreased linearly by (10), and the probability parameter is oscillating by incorporating a cosine perturbation in (11) such that the mutation strategy has more random chance to be chosen during the evolutionary process. Then, the proposed strategy could not only balance the exploration and exploitation during the evolutionary process, but also adapt different evolutionary stages and complex problems. Unlike the adaption strategies in Refs. [22–27], the proposed method is simple and does not require extra store spaces or computational costs since the probability parameter is only related to the number of iteration. Moreover, compared with the adaption strategies in Refs. [11,28], where a iteration-based decreasing monotonically probability parameter is used to choose a more suitable mutation strategy, a suitable cosine perturbation in the proposed method enhances the randomness of choosing mutation strategy during the whole evolutionary process, and increases the possibility to choose the mutation strategy with better exploration during the later evolutionary process. Thus, the proposed strategy could enhance the capability of jumping out of local optimum and adapting different evolutionary stages and complex problems. Furthermore, the validity of the cosine perturbation is illustrated by experiments in Subsection 4.2.1.

3.2. The information intercrossing and sharing mechanism

According to the work specialization, the work efficiency could be effectively improved by dividing a task into several sub-tasks, many variants of DE with multipopulation have been proposed [29–32], where the population is divided into several subpopulations and each subgroup only performs its special task to balance the exploration and exploitation. In particular, Cui *et al.* [30] divided the population into three subpopulations according to their fitness values and three novel mutation strategies were developed and assigned for them respectively to balance the exploitation and exploration, while Han *et al.* [31] divided the population into better and worse subpopulations dynamically based on their fitness values and assigned the local and global search operator to them, respectively. Although the methods in Refs. [30,31] have made some progress on the improvement of performance of DE, the information intercrossing and sharing between different subpopulations has not been considered. It should be mentioned that the information intercrossing and sharing between different subpopulations might be very helpful to make good use of the promising information of individuals and balance effectively the exploration and exploitation. Then we propose an information intercrossing and sharing mechanism by splitting the population and exchanging or sharing the information of individual among different subpopulations. In the proposed mechanism, the population is first divided into the superior and inferior subpopulations according to their fitness values; and then the individuals from the superior subpopulation share their information with each other by the opposite operation, while the individuals from the inferior subpopulation exchange their information with the individual randomly generated from search space or the best individual according to their fitness values by using the binomial crossover operation. The detailed process of the proposed mechanism at the g generation can be described as follows:

Step 1. According to the fitness values, the population is split into the superior subpopulation X_b^g and inferior subpopulation X_w^g, where X_b^g and X_w^g are composed of the top $\lceil p \cdot NP \rceil$ individuals with better fitness values and the rest individuals respectively, and $\lceil A \rceil$ is the nearest integer larger than or equal to A.

Step 2. For X_b^g, the opposite operation [38,39] is employed to share their information. Specially, for the i-th individual $\vec{x}_{b,i}^g$, its opposite individual $\vec{x'}_{b,i}^g = (x'^g_{b,i,1}, x'^g_{b,i,2}, \ldots, x'^g_{b,i,D})$ can be generated as

$$x'^g_{b,i,j} = l_j + (u_j - x^g_{b,i,j}) \tag{12}$$

for $i = 1, 2, \ldots, \lceil p \cdot NP \rceil$ and $j = 1, 2, \ldots, D$, where l_j and u_j are the minimum and maximin of the j-th component of X_b^g, respectively. Moreover, the set of $\lceil p \cdot NP \rceil$ opposite individuals is denoted by X'^g_b.

Step 3. For X_w^g, the individuals are intercrossed with the individual $\vec{I}$ randomly generated from search space or the best individual $\vec{x}_{best}^g$ according to their fitness values by using the binomial crossover operation, and the individual with better or worse fitness value has more chance to exchange with $\vec{x}_{best}^g$ or $\vec{I}$. In particular, for the i-th individual $\vec{x}_{w,i}^g$, its corresponding individual $\vec{x'}_{w,i}^g$ would be generated as

$$\vec{x'}_{w,i}^g = \begin{cases} \vec{x'}_{Iw,i}^g & \text{if } (rand(0,1) < \xi_{2,i}) \\ \vec{x'}_{Bw,i}^g & \text{otherwise} \end{cases} \tag{13}$$

for $i = 1, 2, \ldots, NP - \lceil p \cdot NP \rceil$, where $\vec{I} = \{I_1, I_2, \ldots, I_D\}$ and its each component I_j $(j = 1, 2, \ldots, D)$ is generated randomly by (1), $\xi_{2,i}$ is the probability parameter to evaluate the characteristic of $\vec{x}_{w,i}^g$, $\vec{x'}_{Iw,i}^g = (x'^g_{Iw,i,1}, x'^g_{Iw,i,2}, \ldots, x'^g_{Iw,i,D})$ and $\vec{x'}_{Bw,i}^g = (x'^g_{Bw,i,1}, x'^g_{Bw,i,2}, \ldots, x'^g_{Bw,i,D})$ are generated by

$$\vec{x'}_{Iw,i,j}^g = \begin{cases} I_j & \text{if } (rand(0,1) < \xi_3) \\ x^g_{w,i,j} & \text{otherwise} \end{cases} \tag{14}$$

and

$$\vec{x'}_{Bw,i,j}^g = \begin{cases} x^g_{best,j} & \text{if } (rand(0,1) < \xi_3) \\ x^g_{w,i,j} & \text{otherwise} \end{cases} \tag{15}$$

for $j = 1, 2, \ldots, D$ respectively, and ξ_3 is the crossover parameter.

To accurately evaluate the characteristic of individual, the rank and fitness value of individual are employed simultaneously to set the probability parameter $\xi_{2,i}$ with the same weight, *i.e.*,

$$\xi_{2,i} = \frac{1}{2}\left(\frac{Rank_i}{NP} + \frac{fit_i - fit_{\min}}{fit_{\max} - fit_{\min}}\right), \tag{16}$$

where $Rank_i$ represents the rank of $\vec{x}_i^g$ in the ascending order according to fitness value, fit_i, $fit_{\max}$ and $fit_{\min}$ are the fitness values of $\vec{x}_i^g$, the worse and best individuals, respectively. Furthermore, to guarantee the convergence of algorithm during the later evolutionary process, the possibility of individual intercrossing with $\vec{x}_{best}^g$ or $\vec{I}$ should decrease gradually as the iteration increases. Then, let

$$\xi_3 = \gamma\left(1 - \frac{g}{G}\right), \tag{17}$$

where γ is a parameter and G is the maximum of iterations. For γ, a too large value would not make full use of the information of the current individual during the earlier evolutionary process since the information of $\vec{I}$ and $\vec{x}_{best}^g$ are almost employed to generate its corresponding individual, while a too small value would reduce the chance of exchanging the information of current individual with others. Then, let $\gamma = 0.5$, which is shown to be a suitable choice by experiments in Subsection 4.1.

Step 4. New population is generated by selecting the top $\lceil p \cdot NP \rceil$ individuals from $X_b^g \bigcup X'^g_b$ and combining them with X'^g_w.

From the above discussion, when the proposed mechanism executed, the superior individuals would share their information with each other by the opposite operation, while the worse individuals would exchange their information with $\vec{x}^g_{best}$ or $\vec{I}$ by the binomial crossover operation according to their fitness values. Moreover, for the individuals in X^g_w, the one with better and worse fitness value has more chance to exchange with $\vec{x}^g_{best}$ and $\vec{I}$ respectively, and the exchanging possibility decreases gradually as the iteration increases. Therefore, the proposed mechanism could not only make full use of the information of different individuals, but also effectively balance the exploration and exploitation during the evolutionary process. In contrast to the DE variants with the concept of work specialization in Refs. [29–32], where the population is always divided into several groups and each subpopulation is assigned with a prescribed mutation strategy, the proposed mechanism divides the population into the superior and inferior subpopulations, and employs the opposite operation and binomial crossover operation to share the information of superior individuals with each other and exchange the information of inferior individuals with the best individual or the one randomly generated from search space respectively, such that it not only keeps the concept of work specialization to effectively balance the exploration and exploitation during the evolutionary process, but also promotes the information dissemination between different subpopulations to make good use of the information of different individuals. Furthermore, the validity of the proposed mechanism is illustrated by experiments in Subsection 4.2.2.

Moreover, to further enhance the performance of DE, a simple rule is developed and adopted here as the implement criterion of the proposed mechanism, *i.e.*, the proposed mechanism will be performed every k iterations, where $k = 1/freq$, the period of the cosine perturbation in the proposed stochastic mixed mutation strategy. Obviously, this rule is helpful to enhance the ability of DE to jump out of local optimum since the diversity of population could be improved by the information intercrossing and sharing mechanism and the stochastic mixed mutation strategy. Unfortunately, although the proposed mechanism enhances the exploitation of algorithm by sharing the information of superior individuals with each other, it could reduce the performance of DE for some simple problems since the improved diversity of population decreases the convergence speed. Furthermore, the sensitivity analysis of k is demonstrated in Subsection 4.1.

3.3. Parameter setting

Same as mutation strategy, parameter settings (scaling factor F and crossover rate Cr) play an important role in the performance of DE. Then many works have been developed to them (see Refs. [1,30,32,47] and references therein). In particular, the constant method in Ref. [1] improves the running efficiency of DE algorithm, but a fixed value is often unsuitable for all problems, and requires more time to tune. Unlike constant method, the random method in Ref. [47] can avoid the redundancy of tuning a suitable value and improve its robustness. However, it cannot adapt the different evolutionary processes. Meanwhile, the adaptive methods in Refs. [30,32] can dynamically adjust parameters during the evolutionary processes and effectively balance the exploration and exploitation, yet it requires a large amount of extra store spaces and expensive cost to compute. Specially, Storn and Price [1] suggested that F should lie in $[0.4, 0.95]$, and Cr in $[0, 0.2]$ for separable problem and in $[0.9, 1.0]$ for nonseparable and multimodal problem.

Based on the statements above, to reduce the cost of DE, let F_1 be a random number in $[0.4, 1.0]$, and the adaptive methods in Refs. [30,32] are used to set Cr as follow:

For the individual $\vec{x}^g_i$ at the g-th generation, its corresponding crossover rate

$$Cr^g_i = N(Cr^g_m, 0.1), \quad i = 1, 2, \ldots, NP, \tag{18}$$

where $N(Cr^g_m, 0.1)$ is the normal distribution with mean Cr^g_m and standard deviation 0.1, and

$$Cr^g_m = w_{Cr} \cdot Cr^{g-1}_m + (1 - w_{Cr}) \cdot mean_L(Cr^{g-1}_{good}), \tag{19}$$

Cr^{g-1}_{good} is the set of all successful crossover rates at $g-1$ generation, $mean_L(\cdot)$ is the Lehmer mean [14], and w_{Cr} is a weight coefficient and randomly generated in $[0.8, 1.0]$. To ensure the validity of Cr^g_i, let

$$Cr^g_i = \begin{cases} 0, & \text{if } (Cr^g_i < 0), \\ 1, & \text{if } (Cr^g_i > 1). \end{cases} \tag{20}$$

Moreover, differing from Refs. [30,32], when Cr^{g-1}_{good} is empty, we let

$$Cr^g_m = 1 - Cr^{g-1}_m. \tag{21}$$

In contrast to setting both F and Cr by the adaptive methods in Refs. [30,32], F_1 is designed by a random number within a interval to reduce the cost of algorithm, while Cr is adaptively adjusted to enhance the performance of search by modifying the update approach of Cr^{g-1}_m whenever Cr^{g-1}_{good} is empty. Thus, this parameter setting could not only enhance the performance of DE, but also reduce the cost of algorithm during the evolutionary process.

In summary, the framework of ISDE can be shown in the algorithm below.

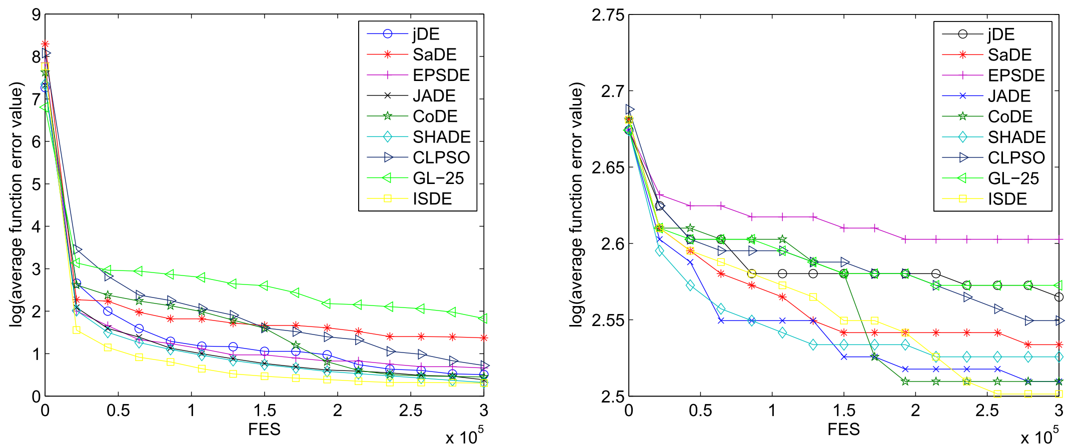
3.4. Complexity analysis

Now, we shall analyze the complexity of ISDE. Obviously, the main differences between ISDE and the classical DE algorithm are the stochastic mixed mutation strategy and the information intercrossing and sharing mechanism, and the main operations of them are the sorting of population according to their fitness values. According to [41], the complexity of the classical DE algorithm is $O(NP \cdot D \cdot G)$. Moreover, it is easy to see that the complexity of sorting the population is $O(NP \cdot \log_2 NP)$.

Because the stochastic mixed mutation strategy and the information intercrossing and sharing mechanism are performed at every generation and every k iterations respectively, the complexity of ISDE is $O\left(NP \cdot D \cdot G + NP \cdot \log_2 NP \cdot G \cdot \left(1 + 1/k\right)\right)$, which can be simplified as $O(NP \cdot D \cdot G)$ since $1/k$ and $\log_2 NP$ are often much smaller than 1 and D, respectively. Then the complexity of ISDE is almost the same as that of the classical DE algorithm.

Algorithm 1 (The framework of ISDE)

Input: the size of population NP, the initial number of iteration $g = 0$, the interval of scaling factor F_1, the initial average crossover rate Cr^0_m, and the maximin number of iteration G.
Initialize the population X^0 randomly and calculate its fitness values;
$g = 1$;
while $g \le G$ **do**
  Perform the mutation operation by Eqs. (9)-(11);
  Generate the offspring population by Eqs. (7) and (18)–(21);
  Update the population by Eq (8);
  if $g\%k == 0$ **then**
    Execute the information intercrossing and sharing mechanism;
  end if
  $g = g + 1$;
end while
Output: The best individual and its fitness value.



(a) f_{13}. (b) f_{14}.

Fig. 7. Evolution curves of nine algorithms with $D = 30$. (a) f_{13}, (b) f_{14}.

4. Numerical experiments

In this section, the performance of ISDE is evaluated on 55 well-known benchmark functions f_1-f_{25} from CEC 2005 [42] and f_{26}-f_{55} from CEC 2014 [43], where f_1-f_5 and f_{26}-f_{28} are unimodal functions, f_6-f_{12} and f_{29}-f_{41} are basic multimodal functions, f_{13}-f_{14} are expanded multimodal functions, and f_{15}-f_{25} and f_{42}-f_{55} are hybrid composition functions. Meanwhile, the sensitivities of parameters in ISDE are analyzed, and the validity of the cosine perturbation in the stochastic mixed mutation strategy and the information intercrossing and sharing mechanism are illustrated. Finally, ISDE is compared with both nine variants of DE algorithm and three non-DE algorithms, and its efficiency is also discussed.

In these experiments, all problems are run 25 times independently, and the maximum number of function evaluations ($FES_{\max}$) is set to $10000D$ for all algorithms. In ISDE, the size of population NP is set to 50. All experiments are conducted in MATLAB R2010a on a personal computer (Intel i3-4570 CUP 3.20 GHz. RAM 4.00 GB), and the average error (Mean Error) and standard deviation (Std Dev) of the function error of 25 independent runs are recorded for measuring their performances. Moreover, to have statistically sound conclusions, Wilcoxon's rank sum test at 0.05 significance level and the rank of each algorithm are conducted on their experimental results.

4.1. *The sensitivities of parameters*

In this subsection, we shall analyze the sensitivities of parameters in ISDE by a series of tuning experiments on f_1-f_{14} with $D = 30$ and $NP = 50$.

4.1.1. *The sensitivity of parameter k*

To analyze the sensitivity of parameter k, a series of tuning experiments of ISDE are conducted on f_1-f_{14} with $k = 50, 100, 200, 300, 500, 1500$ and 3000. In these experiments, other parameters in ISDE are consistent with Section 3. Table 1 reports the experimental results and rank, where the best results are marked by bold on each function, and the last two rows are the sum of rank (Sum Rank) and average rank (Aver Rank), respectively.

From Table 1, we see that the suitable value of k is closely related to the function. In particular, for unimodal functions f_1-f_5, ISDE with $k = 50$ obtains the worse results on all functions, and ISDE with $k = 200$ and 100 have the best results on f_3 and f_4 respectively, while ISDE with $k = 1500$ gets the best results on f_2 and f_5. For basic multimodal functions f_6-f_{12}, ISDE has worse results when k is larger than 300, and its performance decreases gradually as the value of k increases. Clearly, f_6 has the best results when $k = 50$ and 100, f_7 and f_8 get the best results when $k = 200$, f_9 attains the best results when $k = 100$ and 200, and f_{10}-f_{12} have the best results when $k = 100$. Similar to the basic multimodal functions, ISDE gets the best results on the expanded multimodal functions f_{13}-f_{14} when $k = 100$, but obtains the worse results on them when k is too large. Then, k should not be too small or large for simple problems and complicated problems, respectively. This might be because the frequent executions of the information intercrossing and sharing mechanism could reduce the convergence rate since the diversity of population is enhanced, while a few executions could not enhance the exploration ability to avoid premature convergence for complicated problems with a large amount of local optima. Thus, the suitable value of k should not be smaller than 100 and larger than 300 for simple and complicated problems respectively, and the intervals $[100, 1500]$ and $[50, 300]$ are suggested for them, respectively. Furthermore, according to the statistical results in Table 1, ISDE with $k = 50, 100, 200, 300, 500, 1500$ and 3000 obtain 64, 31, 37, 50, 58, 59 and 75 in term of total rank and 4.57, 2.21, 2.64, 3.57, 4.14, 4.21 and 5.36 in term of average rank on all problems, respectively. Therefore, $k = 100$ is suitable and will be used in the following experiments.

4.1.2. *The sensitivities of α, β and γ*

In this subsection, the sensitivities of parameters α, β and γ are tested on f_1-f_{14} by the same method as [46], where the tested parameter is tuned with various values, and the other parameters remain unchanged. In particular, for these experiments, α is set to 0, 0.2, 0.4, 0.6 and 0.8 respectively, both β and γ are set to 0.1, 0.3, 0.5, 0.7 and 0.9 respectively, and the other parameters are consistent with Section 3. Tables 2–4 report the experimental results and rank for α, β and γ respectively, where the best results are marked by bold on each function, Sum Rank and Aver Rank are same as Table 1.

From Table 2, we see that α should not be too small or too large. In particular, according to the statistical results in Table 2, ISDE with $\alpha = 0, 0.2, 0.4, 0.6$ and 0.8 get 39, 41, 30, 29 and 46 in term of total rank and 2.79, 2.93, 2.14, 2.07 and 3.29 in term of average rank on all problems, respectively. This might be that for α, a too small value does not ensure the convergence of algorithm during the later evolutionary process, while a too large one results in the invalidity of the cosine perturbation and the reduction of the population diversity such that the algorithm could not jump out of the local optimum during the evolutionary process. Thus, 0.6 is a reasonable choice of α.

Table 9

Experimental results of ISDE and twelve typical algorithms on CEC 2014 contest test instances with $D = 30$.

Func	Statistic	jDE	SaDE	EPSDE	JADE	CoDE	SHADE	MPEDE	SinDE	CIPDE	CLPSO	CMA-ES	GL-25	ISDE
f_{26}	Mean Error	1.82E+05+	3.44E+05+	2.80E+04+	8.75E+02-	3.64E+04+	2.35E+02-	1.06E-03-	1.33E+06+	2.86E+03-	9.15E+06+	**0.00E+00-**	1.08E+06+	2.55E+04
	Std Dev (Rank)	1.87E+05(9)	2.31E+05(10)	1.13E+05(8)	1.61E+03(3)	2.32E+04(7)	4.27E+02(2)	2.36E-03(4)	1.00E+06(12)	2.72E+03(5)	2.15E+06(13)	**1.74E-14(1)**	1.32E+06(11)	1.70E+04(6)
f_{27}	Mean Error	**0.00E+00≈**	**0.00E+00≈**	**0.00E+00≈**	2.05E-14+	**0.00E+00≈**	1.71E-14+	7.10E-06+	**0.00E+00≈**	2.96E-14+	1.31E+02+	**0.00E+00≈**	1.16E+03+	**0.00E+00**
	Std Dev (Rank)	**2.32E-14(1)**	**5.16E-14(1)**	**7.67E-14(1)**	1.30E-14(9)	**0.00E+00(1)**	1.42E-14(8)	9.03E-06(11)	**0.00E+00(1)**	5.68E-15(10)	3.14E+02(12)	**3.62E-14(1)**	2.03E+03(13)	**8.20E-15(1)**
f_{28}	Mean Error	**0.00E+00-**	5.55E+00+	9.00E-12-	1.93E-03+	2.50E-14-	3.41E-14-	7.53E-08-	6.11E-11-	2.53E-01+	1.92E+02+	**0.00E+00-**	3.15E-01+	1.94E-04
	Std Dev (Rank)	**4.92E-14(1)**	1.43E+01(12)	2.54E-11(5)	6.68E-03(9)	6.98E-14(3)	2.84E-14(4)	1.27E-07(7)	2.85E-10(6)	4.62E-01(10)	2.17E+02(13)	**6.36E-14(1)**	6.96E-01(11)	4.51E-04(8)
f_{29}	Mean Error	2.99E+01+	3.02E+01+	3.72E+00-	8.64E-14-	5.14E+00-	5.46E-14-	1.93E-01-	3.07E+01+	1.66E-13-	7.17E+01+	**0.00E+00-**	9.75E+01+	1.62E+01
	Std Dev (Rank)	3.07E+01(9)	4.15E+01(10)	2.25E+00(6)	5.22E-14(3)	1.78E+01(7)	3.47E-14(2)	4.48E-01(5)	2.91E+01(11)	1.26E-13(4)	1.68E+01(12)	**7.43E-14(1)**	1.34E+01(13)	2.34E+01(8)
f_{30}	Mean Error	2.04E+01+	2.05E+01+	2.04E+01+	2.03E+01≈	2.01E+01-	2.02E+01-	2.04E+01+	2.06E+01+	2.06E+01+	2.04E+01+	**2.00E+01-**	2.10E+01+	2.03E+01
	Std Dev (Rank)	4.44E-02(6)	5.52E-02(10)	6.39E-02(6)	3.35E-02(4)	9.33E-02(2)	3.71E-02(3)	4.92E-02(6)	4.04E-02(11)	3.30E-02(11)	6.04E-02(6)	**2.66E-06(1)**	5.37E-02(13)	3.53E-02(4)
f_{31}	Mean Error	1.47E+01+	5.54E+00+	1.85E+01+	9.07E+00+	4.17E+00+	9.66E+00+	1.54E+01+	**3.73E-02-**	4.53E+00+	1.32E+01+	4.79E+01+	5.81E+00+	3.45E+00
	Std Dev (Rank)	1.55E+00(10)	1.84E+00(5)	2.27E+00(12)	3.15E+00(7)	2.12E+00(3)	3.56E+00(8)	9.41E-01(11)	**1.80E-01(1)**	2.06E+00(4)	1.01E+00(9)	7.85E+00(13)	4.11E+00(6)	2.48E+00(2)
f_{32}	Mean Error	**0.00E+00≈**	6.79E-03+	2.07E-03+	**0.00E+00≈**	4.93E-04+	3.55E-03+	5.32E-11+	**0.00E+00≈**	6.82E-14+	1.09E-05+	1.77E-03+	9.00E-12+	**0.00E+00**
	Std Dev (Rank)	**1.01E-13(1)**	1.10E-02(13)	4.31E-03(11)	**0.00E+00(1)**	2.46E-03(9)	6.28E-03(12)	1.19E-10(7)	**0.00E+00(1)**	5.68E-14(5)	3.96E-05(8)	3.73E-03(10)	1.66E-11(6)	**2.32E-14(1)**
f_{33}	Mean Error	**0.00E+00≈**	**0.00E+00≈**	3.98E-02+	**0.00E+00≈**	**0.00E+00≈**	5.00E-14+	8.61E+00+	2.05E-01+	**0.00E+00≈**	**0.00E+00≈**	4.08E+02+	2.24E+01+	**0.00E+00**
	Std Dev (Rank)	**0.00E+00(1)**	**0.00E+00(1)**	1.99E-01(9)	**0.00E+00(1)**	**0.00E+00(1)**	5.76E-14(8)	9.02E-01(11)	5.47E-01(10)	**0.00E+00(1)**	**0.00E+00(1)**	8.19E+01(13)	5.89E+00(10)	**3.28E-14(1)**
f_{34}	Mean Error	5.75E+01+	3.87E+01+	4.41E+01+	2.56E+01-	3.80E+01+	2.59E+01-	5.54E+01+	3.10E+01-	**2.07E+01-**	4.93E+01+	5.75E+02+	5.75E+01+	3.32E+01
	Std Dev (Rank)	6.87E+00(11)	8.25E+00(7)	7.79E+00(8)	3.52E+00(2)	8.84E+00(6)	8.67E+00(3)	7.09E+00(10)	7.62E+00(4)	**7.21E+00(1)**	8.35E+00(9)	1.39E+02(13)	5.92E+01(11)	5.33E+00(5)
f_{35}	Mean Error	**3.80E-02-**	2.24E-01+	1.45E+01+	7.49E-02-	6.09E-01+	1.08E-02-	2.02E+02+	7.81E+01+	1.07E+02+	3.20E+00+	5.12E+03+	7.25E+02+	1.07E-01
	Std Dev (Rank)	**5.92E-02(1)**	3.69E-01(6)	8.62E-02(5)	1.33E-02(2)	5.81E-01(3)	1.49E-02(7)	2.76E+01(11)	2.42E+01(9)	3.03E+01(10)	1.16E+00(8)	8.01E+02(13)	3.16E+02(12)	5.15E-02(4)
f_{36}	Mean Error	2.88E+03+	3.41E+03+	3.50E+03+	1.64E+03+	1.62E+03+	1.61E+03+	3.32E+03+	1.94E+03+	2.45E+03+	2.18E+03+	5.17E+03+	5.92E+03+	**1.58E+03**
	Std Dev (Rank)	2.49E+02(8)	4.42E+02(10)	4.04E+02(11)	2.51E+02(4)	4.34E+02(3)	2.45E+02(2)	2.42E+02(9)	5.52E+02(5)	4.88E+02(7)	2.74E+02(6)	9.79E+02(12)	1.51E+03(13)	**3.52E+02(1)**
f_{37}	Mean Error	5.27E-01+	8.05E-01+	5.19E-01+	2.62E-01-	**6.23E-02-**	2.37E-01-	6.36E-01+	9.98E-01+	8.74E-01+	3.84E-01+	4.25E-01+	2.44E+00+	3.44E-01
	Std Dev (Rank)	7.45E-02(8)	7.37E-02(10)	7.51E-02(7)	3.18E-02(3)	**3.60E-02(1)**	3.41E-02(2)	9.11E-02(9)	1.01E-01(12)	1.44E-01(11)	7.28E-02(5)	6.40E-01(6)	3.67E-01(13)	7.29E-02(4)
f_{38}	Mean Error	3.27E-01+	2.50E-01+	2.50E-01+	2.15E-01+	2.37E-01+	2.21E-01+	2.24E-01+	2.40E-01+	**9.24E-02-**	3.25E-01+	2.63E-01+	2.63E-01+	2.12E-01
	Std Dev (Rank)	4.58E-02(13)	3.49E-02(8)	4.76E-02(8)	3.40E-02(3)	4.57E-02(6)	3.91E-02(4)	2.56E-02(5)	3.41E-02(7)	**2.35E-02(1)**	4.79E-02(12)	8.79E-02(10)	3.76E-02(10)	3.86E-02(2)
f_{39}	Mean Error	2.96E-01+	2.38E-01+	2.85E-01+	2.41E-01+	2.30E-01+	2.58E-01+	2.18E-01+	2.40E-01+	2.91E-01+	2.52E-01+	3.66E-01+	2.59E-01+	**2.12E-01**
	Std Dev (Rank)	3.23E-02(12)	3.32E-02(4)	7.60E-02(10)	2.81E-02(6)	3.82E-02(3)	5.62E-02(8)	2.08E-02(2)	2.80E-02(5)	2.76E-02(11)	2.56E-02(7)	7.55E-02(13)	3.35E-02(9)	**2.60E-02(1)**
f_{40}	Mean Error	6.63E+00+	4.48E+00-	5.38E+00+	3.07E+00-	2.87E+00-	**2.74E+00-**	6.21E+00+	3.99E+00-	4.38E+00-	7.37E+00+	3.83E+00-	1.30E+01+	4.98E+00
	Std Dev (Rank)	8.45E-01(11)	1.87E+00(7)	6.73E-01(9)	3.37E-01(3)	7.18E-01(2)	**4.65E-01(1)**	7.58E-01(10)	8.95E-01(5)	9.80E-01(6)	9.44E-01(12)	1.07E+00(4)	4.77E+00(13)	1.31E+00(8)
f_{41}	Mean Error	1.03E+01+	1.10E+01+	1.12E+01+	9.43E+00+	9.26E+00+	9.52E+00+	1.06E+01+	1.08E+01+	9.45E+00+	1.03E+01+	1.43E+01+	1.18E+01+	**9.18E+00**
	Std Dev (Rank)	2.69E-01(6)	3.01E-01(10)	5.43E-01(11)	4.09E-01(3)	8.05E-01(2)	3.56E-01(5)	2.27E-01(8)	4.43E-01(9)	7.90E-01(4)	3.59E-01(7)	4.16E-01(13)	2.90E-01(12)	**5.15E-01(1)**
f_{42}	Mean Error	2.62E+03+	1.41E+04+	3.49E+04+	3.18E+04+	1.45E+03+	8.93E+02-	**1.77E+02-**	9.28E+04+	1.51E+04+	7.71E+05+	1.84E+03+	1.99E+05+	1.00E+03
	Std Dev (Rank)	2.02E+03(6)	1.51E+04(7)	4.64E+04(10)	1.53E+05(9)	1.24E+03(4)	3.73E+02(2)	**1.20E+02(1)**	6.91E+04(11)	6.94E+04(8)	2.74E+05(13)	4.63E+02(5)	1.06E+05(12)	7.24E+02(3)
f_{43}	Mean Error	1.83E+01-	3.27E+02+	7.66E+02+	2.77E+02+	1.51E+01-	5.27E+01-	**9.14E+00-**	4.82E+02+	9.74E+01-	1.11E+02-	1.56E+02-	2.37E+02-	2.76E+02
	Std Dev (Rank)	7.42E+00(3)	4.72E+02(11)	2.21E+03(13)	7.24E+02(10)	6.93E+00(2)	2.27E+01(4)	**3.55E+00(1)**	6.17E+02(12)	3.17E+01(5)	5.12E+01(6)	3.78E+01(7)	3.21E+02(8)	3.20E+02(9)
f_{44}	Mean Error	5.14E+00+	6.67E+00+	1.32E+01+	4.32E+00+	3.96E+00+	4.68E+00+	3.57E+00≈	**3.41E+00-**	4.52E+00+	7.47E+00+	1.02E+01+	4.71E+00+	3.57E+00
	Std Dev (Rank)	5.99E-01(9)	1.17E+01(10)	1.10E+00(13)	9.11E-01(5)	1.09E+00(4)	7.63E-01(7)	7.83E-01(2)	**6.96E-01(1)**	5.95E-01(6)	4.82E-01(11)	1.53E+00(12)	6.23E-01(8)	7.03E-01(2)

(*continued on next page*)

Table 9 *(continued)*

Func	Statistic	jDE	SaDE	EPSDE	JADE	CoDE	SHADE	MPEDE	SinDE	CIPDE	CLPSO	CMA-ES	GL-25	ISDE
f_{45}	Mean Error	1.38E+01+	8.76E+01+	1.07E+02+	3.45E+03+	1.40E+01+	1.83E+01+	1.14E+01+	1.09E+01+	8.74E+02+	3.23E+03+	2.84E+02+	1.94E+02+	**1.07E+01**
	Std Dev (Rank)	3.93E+00(4)	5.30E+01(7)	3.76E+02(8)	3.18E+03(13)	1.01E+01(5)	9.42E+00(6)	3.34E+00(3)	2.88E+00(2)	1.26E+03(11)	1.62E+03(12)	1.22E+02(10)	1.23E+02(9)	**3.53E+00(1)**
f_{46}	Mean Error	4.52E+02+	5.28E+03+	7.41E+03+	3.29E+03+	2.49E+02+	2.72E+02+	2.79E+02+	3.84E+03+	7.91E+03+	8.47E+04+	9.82E+02+	5.59E+04+	**2.47E+02**
	Std Dev (Rank)	2.13E+02(5)	7.94E+03(9)	1.09E+04(10)	1.51E+04(7)	1.41E+02(2)	9.71E+01(3)	9.36E+01(4)	4.61E+03(8)	2.76E+04(11)	5.47E+04(13)	3.16E+02(6)	2.45E+04(12)	**1.89E+02(1)**
f_{47}	Mean Error	1.86E+02+	1.72E+02+	2.42E+02+	1.61E+02+	1.55E+02+	9.37E+01+	1.45E+02+	8.47E+01+	2.04E+02+	2.02E+02+	2.27E+02+	1.52E+02+	**8.17E+01**
	Std Dev (Rank)	6.88E+01(9)	5.98E+01(8)	1.07E+02(13)	7.09E+01(7)	7.12E+01(6)	6.42E+01(3)	5.71E+01(4)	4.98E+01(2)	1.01E+02(11)	7.25E+01(10)	1.40E+02(10)	6.09E+01(5)	**6.04E+01(1)**
f_{48}	Mean Error	3.15E+02+	3.15E+02+	**3.14E+02≈**	3.15E+02+	3.15E+02+	3.15E+02+	3.15E+02+	3.15E+02+	3.15E+02+	3.15E+02+	3.15E+02+	3.15E+02+	**3.14E+02**
	Std Dev (Rank)	9.28E-13(3)	1.25E-12(3)	**1.21E-12(1)**	0.00E+00(3)	1.28E-13(3)	0.00E+00(3)	1.33E-10(3)	5.78E-14(3)	0.00E+00(3)	3.72E-06(3)	3.93E-12(3)	1.33E-09(3)	**1.11E-02(1)**
f_{49}	Mean Error	2.24E+02+	2.26E+02+	2.30E+02+	2.25E+02+	2.25E+02+	2.30E+02+	2.24E+02+	**2.22E+02-**	2.25E+02+	2.23E+02≈	2.41E+02+	**2.22E+02-**	2.23E+02
	Std Dev (Rank)	1.21E+00(5)	3.14E+00(10)	7.49E+00(11)	2.83E+00(7)	2.79E+00(7)	6.11E+00(11)	4.59E-01(5)	**1.28E+00(1)**	2.33E+00(7)	4.72E+00(3)	4.69E+01(13)	**5.99E-01(1)**	9.54E-01(3)
f_{50}	Mean Error	2.04E+02+	2.09E+02+	**2.00E+02-**	2.03E+02+	2.04E+02+	2.03E+02+	2.03E+02+	2.04E+02+	2.08E+02+	2.08E+02+	2.04E+02+	2.07E+02+	2.02E+02
	Std Dev (Rank)	6.74E-01(6)	1.76E+00(13)	**3.19E-02(1)**	6.92E-01(3)	1.48E+00(6)	4.94E-01(3)	1.45E-01(3)	4.82E-01(6)	3.17E+00(11)	8.30E-01(11)	2.44E+00(6)	1.95E+00(10)	5.04E-01(2)
f_{51}	Mean Error	**1.00E+02≈**	1.04E+02+	**1.00E+02≈**	**1.00E+02≈**	**1.00E+02≈**	**1.00E+02≈**	**1.00E+02≈**	**1.00E+02≈**	**1.00E+02≈**	**1.00E+02≈**	1.13E+02+	**1.00E+02≈**	**1.00E+02**
	Std Dev (Rank)	**4.54E-02(1)**	2.00E+01(12)	**4.13E-02(1)**	**3.56E-02(1)**	**4.57E-02(1)**	**5.09E-02(1)**	**2.67E-02(1)**	**2.98E-02(1)**	**1.78E-02(1)**	**9.15E-02(1)**	4.60E+01(13)	**4.64E-02(1)**	**3.63E-02(1)**
f_{52}	Mean Error	3.81E+02+	4.17E+02+	8.41E+02+	3.65E+02+	3.85E+02+	3.61E+02+	3.97E+02+	3.04E+02-	3.21E+02-	4.13E+02+	4.07E+02+	**3.02E+02-**	3.56E+02
	Std Dev (Rank)	4.12E+01(7)	3.46E+01(12)	8.24E+01(13)	4.96E+01(6)	3.49E+01(8)	3.34E+01(5)	1.79E+01(9)	1.35E+01(2)	3.80E+01(3)	5.77E+00(11)	1.41E+02(10)	**8.78E-01(1)**	4.68E+01(4)
f_{53}	Mean Error	7.99E+02+	8.92E+02+	**3.96E+02-**	7.90E+02+	8.26E+02+	8.28E+02+	8.60E+02+	7.91E+02+	7.96E+02+	9.20E+02+	3.69E+03+	8.80E+02+	4.12E+02
	Std Dev (Rank)	2.48E+01(6)	3.48E+01(11)	**1.58E+01(1)**	4.57E+01(3)	3.08E+01(7)	2.80E+01(8)	2.53E+01(9)	2.34E+01(4)	2.96E+01(5)	5.66E+01(12)	2.68E+03(13)	3.04E+01(10)	1.21E+01(2)
f_{54}	Mean Error	8.92E+02+	1.05E+03+	**2.14E+02-**	7.26E+02+	7.87E+02+	7.13E+02+	4.00E+02-	1.48E+03+	7.61E+02+	1.01E+03+	8.14E+02+	1.01E+03+	5.14E+02
	Std Dev (Rank)	1.32E+02(9)	1.63E+02(12)	**1.03E+00(1)**	9.93E+00(5)	1.50E+02(7)	6.68E+01(4)	2.85E+02(2)	2.72E+02(13)	7.01E+01(6)	9.84E+01(10)	9.54E+01(8)	1.03E+02(10)	3.26E+02(3)
f_{55}	Mean Error	1.56E+03+	1.76E+03+	6.04E+02+	1.64E+03+	8.46E+02+	1.92E+03+	5.54E+02+	1.34E+03+	1.48E+03+	3.69E+03+	2.29E+03+	1.33E+03+	**5.43E+02**
	Std Dev (Rank)	6.83E+02(8)	4.17E+02(10)	1.03E+02(3)	6.24E+02(9)	3.30E+02(4)	1.17E+03(11)	1.14E+02(2)	5.02E+02(6)	4.35E+02(7)	9.60E+02(13)	6.05E+02(12)	2.74E+02(5)	**1.48E+02(1)**
Sum Rank		189	201	261	151	125	150	175	171	216	260	253	261	91
Aver Rank		6.3	6.7	8.7	5.03	4.17	5	5.83	5.7	7.2	8.67	8.43	8.7	3.03
+		23	27	22	20	21	18	22	20	21	26	23	26	
–		3	1	5	6	6	10	6	7	7	1	6	3	
≈		4	2	3	4	3	2	2	3	2	3	1	1	

From Table 3, we see that the performance of ISDE is closely related to β. According to the statistical results of Table 3, ISDE with $\beta = 0.1, 0.3, 0.5, 0.7$ and 0.9 get 41, 35, 26, 38 and 47 in term of total rank and 2.93, 2.5, 1.86, 2.71 and 3.36 in term of average rank on all problems, respectively. These might be due to the fact that a too large value for β could lead to a large amount of invalid searches during the earlier evolutionary process, while a too small value could reduce the global search ability of ISDE. Thus β should not be a too large or small value, and 0.5 is a suitable choice of β.

From Table 4, we see that the performance of ISDE is sensitive to γ. Specially, according to the statistical results of Table 4, ISDE with $\gamma = 0.1, 0.3, 0.5, 0.7$ and 0.9 have 37, 40, 23, 35 and 45 in term of total rank and 2.64, 2.86, 1.64, 2.5 and 3.21 in term of average rank on all problems, respectively. These might be attributed to the fact that a too large value for γ could not make good use of the information of the current individual during the earlier evolutionary process, which could reduce the convergence speed, while a too small value would decrease the chance to exchange the information of individuals. Therefore, γ should not be set to a too large or small value, and 0.5 is suitable for γ.

In summary, 0.6, 0.5 and 0.5 are reasonable for α, β and γ in ISDE respectively, and they are helpful to balance the exploration and exploitation and improve the performance of ISDE.

4.2. The validity of the proposed strategies

In this subsection, the performance of the cosine perturbation and the information intercrossing and sharing mechanism shall be illustrated.

4.2.1. The validity of the cosine perturbation

To show the validity of the cosine perturbation in ISDE, the probability parameter ξ_1 in (9) is compared with $\xi_{1-2} = 1 - g/G$, $\xi_{1-3} = (1 - g/G)^2$ [11] and $\xi_{1-4} = (1 - g/G)^{0.5}$ on the basic and expanded multimodal functions f_6-f_{14} with $D = 30$ and $NP = 50$, respectively. In the experiments, the parameters in ISDE are consistent with Section 3 except for ξ_1.

Fig. 2 depicts the curves of four probability parameters above. From Fig. 2, we see that as the iteration increases, ξ_1 decreases gradually with oscillation, while ξ_{1-2}-ξ_{1-4} decrease monotonously. Thus, the proposed probability parameter could select the mutation strategy that has more explorative stability with a larger probability to avoid the premature convergence during the later evolutionary process.

Table 5 reports the experimental results and rank of ISDE with various probability rules, where ISDE-ξ_{1-2}, ISDE-ξ_{1-3} and ISDE-ξ_{1-4} represent the corresponding variants of ISDE by replacing ξ_1 with ξ_{1-2}, ξ_{1-3} and ξ_{1-4} respectively, the best results are marked by bold on each function, Sum Rank and Aver Rank are same as Table 1.

From Table 5, we see that the probability parameter has great influence on the performance of ISDE, and the cosine perturbation could effectively improve its performance. In particular, ISDE-ξ_1 has the best results on f_6, f_8-f_9 and f_{11}-f_{14}, while ISDE-ξ_{1-2} obtains the best results on f_7 and f_{10}. Moreover, according to the statistical results of Table 5, ISDE-ξ_1, ISDE-ξ_{1-2}, ISDE-ξ_{1-3} and ISDE-ξ_{1-4} get 12, 22, 25 and 30 in term of total rank and 1.33, 2.44, 2.78 and 3.33 in term of average rank on all problems, respectively. This is because the cosine perturbation could enhance the randomness of choosing mutation strategy during the evolutionary process, such that the mutation strategy with more exploration or exploitation would have more possibility to be chosen at the later or earlier evolutionary stage. Thus the cosine perturbation could effectively alleviate the difficulty to jump out of the local optimum and improve the performance of ISDE for complicated problems.

4.2.2. The validity of the information intercrossing and sharing mechanism

To evaluate the performance of the information intercrossing and sharing mechanism (IS), ISDE is compared with ISDE without the IS mechanism (denoted by ISDE-1) on f_1-f_{14} with $D = 30$. In this experiment, $NP = 50$, $k = 100$, and other parameters are consistent with Section 3.

Firstly, to clearly show the effects of the IS mechanism on the exploration capability, ISDE and ISDE-1 are tested on four typical functions f_1, f_4, f_{10} and f_{14}, and the following standard deviations of individuals (DI) in Ref. [19] is used to measure the diversity of population:

$$DI = \frac{1}{NP}\sqrt{\sum_{i=1}^{NP} \|\vec{x}_i - \frac{1}{NP}\sum_{j=1}^{NP}(\vec{x}_j)\|^2}. \tag{22}$$

Fig. 3 depicts the DI curves of ISDE and ISDE-1 on four typical functions during the evolutionary process.

From Fig. 3, we see that as the iteration increases, the DI curves of ISDE decrease with a concussion gradually, while the DI curves of ISDE-1 decrease with a very small concussion or none. The reason for this phenomenon is that the IS mechanism is executed for every k iterations, where each individual from inferior subpopulation exchanges its information with the one randomly generated from search space by probability. Then, the IS mechanism could enhance the exploration ability of algorithm, and is helpful to avoid the premature convergence.

Next, to illustrate the effects of the IS mechanism on the exploitation ability, it is conducted on f_1 with $D = 2$, $NP = 10$, $FES_{\max} = 2000$ and $k = 20$. Fig. 4 depicts the distribution of population before and after the IS mechanism is executed, where the red and blue points represent the better and worse individuals, respectively.

From Fig. 4, we see that at the beginning of evolution, the initialization population evenly distributes in the search space, and most individuals gather and exploit the promising region after several iterations, while the better and worse individuals are more crowding and decentralized when the IS mechanism is executed. The reason for this is that in the IS mechanism, the better individuals share their information with each other by the opposite operation, and the individuals from inferior subpopulation exchange their information with the best individual by probability. Then, the IS mechanism could enhance the exploitation ability of algorithm.

Finally, a comparison of ISDE with ISDE-1 is conducted on f_1-f_{14} with $D = 30$ to show the validity of the IS mechanism. Table 6 reports their numerical results, and the best results are marked by bold on each function.

From Table 6, we see that for unimodal functions f_1-f_5, ISDE-1 has the best performance on all functions except for f_4. The reason for this is that the improved diversity of population could reduce the convergence speed. For multimodal functions f_6-f_{14}, ISDE obtains the best results on all functions. This is because the proposed mechanism employs the opposite operation and binomial crossover operation to share the information of superior individuals with each other and exchange the information of inferior individuals with the best individual or the one randomly generated from search space respectively, such that the information of different individuals could be made good use of during the evolutionary process. Therefore, the IS mechanism could effectively balance the exploration and exploitation, and improve the performance of algorithm.

4.3. Comparisons and discussions

To evaluate the benefits of ISDE, we shall make a comparison of ISDE with twelve well-known optimization algorithms, including nine variants of DE algorithms (jDE [40], SaDE [23], EPSDE [24], JADE [14], CoDE [22], SHADE [21], MPEDE [32], SinDE [17] and CIPDE [18]) and three non-DE algorithms (CLPSO [7], CMA-ES [44] and GL-25 [8]), on 55 benchmark functions f_1-f_{55} from both CEC 2005 [42] and CEC 2014 [43]. In particular, jDE [40], SaDE [23], EPSDE [24] and JADE [14] adjust their control parameters adaptively, SHADE [21] is a recently improved version of JADE by using the success history information to adaptively set its parameters, CoDE [22] is an effective

Table 10

Experimental results of ISDE and nine typical algorithms on CEC 2014 contest test instances with $D = 50$.

Func	Statistic	jDE	SaDE	EPSDE	JADE	CoDE	SHADE	MPEDE	SinDE	CIPDE	CLPSO	CMA-ES	GL-25	ISDE
f_{26}	Mean Error	1.30E+06+	9.24E+05+	1.74E+06+	1.55E+04-	2.23E+05-	1.74E+04-	1.08E+05-	2.89E+06+	1.73E+04-	1.60E+07+	**4.43E-14-**	2.11E+06+	4.50E+05
	Std Dev (Rank)	4.60E+05(9)	3.47E+05(8)	5.51E+06(10)	8.53E+03(2)	9.39E+04(6)	1.68E+04(4)	9.12E+04(5)	1.17E+06(12)	8.76E+03(3)	3.36E+06(13)	**1.55E-14(1)**	9.69E+05(11)	2.44E+05(7)
f_{27}	Mean Error	3.82E+02-	3.78E+03+	1.50E-08-	9.21E-14-	3.25E+01-	**8.75E-14-**	1.43E+01-	3.17E+03+	7.57E-12-	6.85E+01-	**8.75E-14-**	2.60E+03+	2.02E+03
	Std Dev (Rank)	5.96E+02(9)	3.53E+03(13)	3.46E-08(5)	4.29E-14(3)	5.80E+01(7)	**4.01E-14(1)**	3.01E+01(6)	3.20E+03(12)	3.51E-11(4)	1.38E+02(8)	**2.00E-14(1)**	1.30E+03(11)	2.58E+03(10)
f_{28}	Mean Error	2.15E-01-	3.83E+03+	3.79E-04-	3.77E+03+	2.86E+01-	1.98E+02-	4.71E-04-	4.13E+02-	1.82E+03+	2.19E+03+	**1.75E-13-**	3.02E+02-	8.40E+02
	Std Dev (Rank)	4.67E-01(4)	2.53E+03(13)	1.83E-03(2)	2.31E+03(12)	4.42E+01(5)	9.88E+02(6)	1.05E-03(3)	3.44E+02(8)	1.58E+03(10)	7.74E+02(11)	**7.32E-14(1)**	3.20E+02(7)	3.67E+02(9)
f_{29}	Mean Error	9.60E+01+	8.69E+01-	4.16E+01-	2.35E+01-	2.99E+01-	1.98E+01-	6.61E+01-	9.54E+01+	**1.35E+01-**	9.52E+01+	1.57E+01-	9.56E+01+	9.26E+01
	Std Dev (Rank)	2.76E+00(13)	4.17E+01(8)	2.30E+01(6)	4.28E+01(4)	3.47E+01(5)	4.00E+01(3)	3.38E+01(7)	4.03E+00(10)	**2.90E+01(1)**	1.11E+01(11)	3.68E+01(2)	1.35E+00(12)	5.41E+00(9)
f_{30}	Mean Error	2.05E+01+	2.07E+01+	2.06E+01+	2.04E+01≈	2.04E+01≈	2.04E+01≈	2.06E+01+	2.08E+01+	2.08E+01+	2.05E+01+	**2.00E+01-**	2.11E+01+	2.04E+01
	Std Dev (Rank)	3.05E-02(6)	4.73E-02(10)	3.24E-02(8)	3.25E-02(2)	8.45E-02(2)	2.99E-02(2)	3.48E-02(8)	4.97E-02(11)	8.94E-02(11)	4.75E-02(6)	**1.75E-06(1)**	3.20E-02(13)	2.41E-02(2)
f_{31}	Mean Error	3.48E+01+	1.84E+01+	4.56E+01+	1.70E+01+	8.67E+00-	2.29E+01+	3.02E+01+	**1.95E-01-**	6.39E+00-	2.89E+01+	7.50E+01+	2.95E+00-	1.04E+01
	Std Dev (Rank)	2.30E+00(11)	3.48E+00(8)	2.63E+00(12)	5.77E+00(6)	3.51E+00(4)	5.27E+00(7)	2.14E+00(10)	**4.16E-01(1)**	2.80E+00(3)	2.69E+00(9)	9.95E+00(13)	1.99E+00(2)	6.56E+00(5)
f_{32}	Mean Error	3.47E-10+	1.16E-02+	5.12E-03+	9.86E-04+	8.88E-04+	4.14E-03+	4.77E-03+	**4.93E-14-**	3.65E-03+	1.45E-04+	6.90E-04+	1.14E-09+	1.82E-13
	Std Dev (Rank)	6.15E-10(3)	1.50E-02(13)	7.58E-03(12)	2.76E-03(8)	2.45E-03(7)	5.36E-03(10)	4.62E-03(11)	**5.73E-14(1)**	5.42E-03(9)	1.69E-04(5)	2.41E-03(6)	1.67E-09(4)	1.79E-13(2)
f_{33}	Mean Error	3.00E-12+	1.07E+00+	1.95E-02+	9.09E-15-	6.37E-01+	1.36E-13+	1.94E+01+	7.50E+00+	**0.00E+00-**	1.46E-13+	6.96E+02+	5.13E+01+	1.14E-13
	Std Dev (Rank)	3.04E-12(6)	1.11E+00(9)	1.39E-01(7)	3.15E-14(2)	6.96E-01(8)	4.64E-14(4)	1.34E+00(11)	3.60E+00(10)	**0.00E+00(1)**	5.21E-14(5)	1.14E+02(13)	1.56E+01(12)	0.00E+00(3)
f_{34}	Mean Error	1.59E+02+	9.17E+01+	1.46E+02+	8.17E+01+	7.74E+01+	6.84E+01-	1.16E+02+	**6.50E+01-**	8.36E+01+	1.23E+02+	1.22E+03+	1.26E+02+	7.64E+01
	Std Dev (Rank)	1.00E+01(12)	1.44E+01(7)	1.63E+01(11)	6.15E+00(5)	1.52E+01(4)	1.24E+01(2)	9.93E+00(8)	**8.13E+00(1)**	1.15E+01(6)	1.63E+01(9)	1.97E+02(13)	1.14E+02(10)	1.37E+01(3)
f_{35}	Mean Error	2.35E+01+	1.41E+00+	5.53E+02+	7.20E-02+	5.17E+00+	5.50E-02+	4.67E+02+	1.51E+02+	3.88E+02+	7.06E+00+	8.26E+03+	2.58E+03+	**4.75E-02+**
	Std Dev (Rank)	5.48E+00(7)	9.96E-01(4)	6.07E+02(11)	1.51E-02(3)	2.39E+00(5)	7.97E-03(2)	5.23E+01(10)	8.23E+01(8)	8.13E+01(9)	2.50E+00(6)	7.11E+02(13)	9.78E+02(12)	**2.48E-02(1)**
f_{36}	Mean Error	6.94E+03+	7.16E+03+	8.95E+03+	4.04E+03≈	4.63E+03+	**3.73E+03-**	6.74E+03+	4.32E+03+	5.73E+03+	4.96E+03+	8.42E+03+	1.27E+04+	4.04E+03
	Std Dev (Rank)	3.38E+02(9)	9.42E+02(10)	5.35E+02(12)	2.35E+02(2)	8.88E+02(5)	**3.33E+02(1)**	3.12E+02(8)	7.90E+02(4)	5.23E+02(7)	4.16E+02(6)	1.12E+03(11)	3.05E+02(13)	7.36E+02(2)
f_{37}	Mean Error	7.37E-01+	1.08E+00+	8.55E-01+	2.59E-01-	**8.13E-02-**	2.30E-01-	7.42E-01+	1.35E+00+	1.15E+00+	4.13E-01+	2.21E-01-	3.39E+00+	3.65E-01
	Std Dev (Rank)	7.65E-02(7)	1.27E-01(10)	6.97E-02(9)	2.83E-02(4)	**4.07E-02(1)**	3.32E-02(3)	7.99E-02(12)	1.40E-01(11)	1.12E-01(6)	7.59E-02(6)	1.52E-01(2)	2.81E-01(13)	6.45E-02(5)
f_{38}	Mean Error	4.50E-01+	4.28E-01+	3.63E-01+	3.30E-01+	3.24E-01+	3.29E-01+	3.10E-01+	3.43E-01+	3.87E-01+	4.17E-01+	3.69E-01+	4.21E-01+	**3.07E-01**
	Std Dev (Rank)	5.70E-02(13)	5.15E-02(12)	5.92E-02(7)	4.62E-02(5)	5.15E-02(3)	5.26E-02(4)	2.94E-02(2)	3.59E-02(6)	4.07E-02(9)	3.42E-02(10)	7.60E-02(8)	5.16E-02(11)	**6.66E-02(1)**
f_{39}	Mean Error	4.13E-01+	3.07E-01+	3.52E-01+	3.11E-01+	2.67E-01+	3.15E-01+	2.80E-01+	2.81E-01+	3.56E-01+	2.93E-01+	5.06E-01+	3.06E-01+	**2.44E-01**
	Std Dev (Rank)	1.66E-01(12)	2.42E-02(7)	9.01E-02(10)	7.89E-02(8)	3.27E-02(2)	8.47E-02(9)	1.85E-02(3)	9.84E-02(4)	3.03E-02(11)	3.39E-02(5)	3.06E-01(13)	3.82E-02(6)	**2.05E-02(1)**
f_{40}	Mean Error	1.64E+01+	1.59E+01+	1.86E+01+	1.41E+01+	6.46E+00-	8.12E+00-	1.33E+01+	7.99E+00-	9.07E+00-	1.73E+01+	**6.02E+00-**	2.09E+01+	1.29E+01
	Std Dev (Rank)	1.49E+00(10)	3.24E+00(9)	2.63E+00(12)	7.48E+00(8)	1.44E+00(2)	1.35E+00(4)	3.95E+00(7)	1.46E+00(3)	2.85E+00(5)	1.63E+00(11)	**1.48E+00(1)**	1.10E+01(13)	3.46E+00(6)
f_{41}	Mean Error	1.94E+01+	2.01E+01+	2.07E+01+	**1.78E+01≈**	1.85E+01+	1.81E+01+	1.92E+01+	2.00E+01+	**1.78E+01≈**	1.90E+01+	2.37E+01+	2.15E+01+	**1.78E+01**
	Std Dev (Rank)	3.03E-01(8)	3.93E-01(10)	4.16E-01(11)	**3.88E-01(1)**	7.62E-01(5)	4.95E-01(4)	4.42E-01(7)	4.14E-01(9)	**1.16E+00(1)**	4.24E-01(6)	5.48E-01(13)	2.72E-01(12)	**3.62E-01(1)**
f_{42}	Mean Error	5.73E+04+	5.30E+04+	2.22E+05+	2.22E+03-	2.38E+04+	2.21E+03-	**9.45E+02-**	3.59E+05+	2.68E+03-	2.67E+06+	2.74E+03-	5.14E+05+	2.06E+04
	Std Dev (Rank)	3.47E+04(9)	3.60E+04(8)	1.49E+05(10)	7.03E+02(3)	8.19E+03(7)	4.11E+02(2)	**3.32E+02(1)**	1.98E+05(11)	1.03E+03(4)	9.10E+05(13)	6.51E+02(5)	2.31E+05(12)	9.23E+03(6)
f_{43}	Mean Error	4.27E+02-	6.45E+02+	3.69E+03+	5.85E+02+	3.95E+02-	3.72E+02-	**4.33E+01-**	3.10E+02-	1.43E+02-	1.92E+02-	2.49E+02-	6.10E+02+	5.14E+02
	Std Dev (Rank)	4.16E+02(8)	5.83E+02(12)	8.15E+03(13)	4.98E+01(10)	2.81E+02(7)	4.87E+01(6)	**1.33E+01(1)**	3.63E+02(5)	3.04E+01(2)	6.99E+01(3)	5.86E+01(4)	3.72E+02(11)	8.29E+02(9)
f_{44}	Mean Error	2.27E+01+	1.77E+01+	2.46E+01+	1.72E+01+	**6.47E+00-**	1.81E+01+	1.01E+01-	9.33E+00-	1.67E+01+	1.70E+01+	1.92E+01+	3.49E+01+	1.61E+01
	Std Dev (Rank)	1.13E+01(11)	8.45E+00(8)	1.69E+00(12)	2.44E+01(7)	**9.00E-01(1)**	3.17E+00(9)	1.26E+00(3)	7.75E-01(2)	7.57E+00(5)	2.17E+00(6)	2.78E+00(10)	1.28E+00(13)	6.49E+00(4)
f_{45}	Mean Error	7.44E+01+	7.81E+02+	4.51E+02+	6.94E+03+	3.04E+02+	1.82E+02+	4.18E+01+	2.14E+02+	3.49E+03+	7.06E+03+	4.61E+02+	3.97E+02+	**4.10E+01**
	Std Dev (Rank)	2.15E+01(3)	3.77E+02(13)	5.63E+02(8)	6.31E+03(11)	2.44E+02(6)	1.07E+02(4)	1.23E+01(2)	1.41E+02(5)	4.20E+03(10)	2.74E+03(12)	1.08E+02(9)	1.68E+02(7)	**1.28E+01(1)**

(*continued on next page*)

Table 10 *(continued)*

Func	Statistic	jDE	SaDE	EPSDE	JADE	CoDE	SHADE	MPEDE	SinDE	CIPDE	CLPSO	CMA-ES	GL-25	ISDE
f_{46}	Mean Error	2.69E+04+	5.87E+04+	7.44E+04+	1.25E+03-	2.55E+04+	1.24E+03-	**5.88E+02-**	2.25E+05+	1.51E+03-	1.55E+06+	1.71E+03-	3.19E+05+	1.14E+04
	Std Dev (Rank)	2.83E+04(8)	3.51E+04(9)	8.53E+04(10)	3.88E+02(3)	6.84E+04(7)	3.69E+02(2)	**2.09E+02(1)**	1.18E+05(11)	4.28E+02(4)	7.42E+05(13)	3.79E+02(5)	1.08E+05(12)	9.01E+03(6)
f_{47}	Mean Error	8.15E+02+	4.43E+02+	8.04E+02+	4.73E+02+	6.69E+02+	4.32E+02+	5.44E+02+	4.49E+02+	6.33E+02+	6.43E+02+	4.51E+02+	5.44E+02+	**4.27E+02**
	Std Dev (Rank)	1.47E+02(13)	1.80E+02(3)	2.05E+02(12)	1.72E+02(6)	2.02E+02(11)	1.74E+02(2)	1.29E+02(7)	1.25E+02(4)	2.45E+02(9)	1.02E+02(10)	2.83E+02(5)	3.39E+02(7)	**1.94E+02(1)**
f_{48}	Mean Error	3.44E+02+	3.44E+02+	**3.37E+02≈**	3.44E+02+	3.44E+02+	3.44E+02+	3.44E+02+	3.44E+02+	3.44E+02+	3.44E+02+	3.44E+02+	3.44E+02+	**3.37E+02**
	Std Dev (Rank)	4.55E-13(3)	0.00E+00(3)	**3.45E-12(1)**	1.71E-13(3)	2.32E-13(3)	3.01E-13(3)	4.29E-11(3)	2.89E-13(3)	5.80E-14(3)	1.61E-06(3)	2.77E-05(3)	2.90E-09(3)	**1.31E-01(1)**
f_{49}	Mean Error	2.69E+02+	2.76E+02+	2.73E+02+	2.74E+02+	2.71E+02+	2.79E+02+	2.71E+02+	2.64E+02+	2.71E+02+	2.58E+02+	3.20E+02+	2.58E+02+	**2.55E+02**
	Std Dev (Rank)	2.40E+00(5)	3.67E+00(11)	5.79E+00(9)	2.27E+00(10)	1.95E+00(6)	2.98E+00(12)	1.54E+00(6)	3.97E+00(4)	1.46E+01(6)	2.50E+00(2)	2.04E+02(13)	4.18E+00(2)	**1.04E+00(1)**
f_{50}	Mean Error	2.09E+02+	2.18E+02+	**2.01E+02-**	2.17E+02+	2.10E+02+	2.09E+02+	2.06E+02+	2.08E+02+	2.21E+02+	2.16E+02+	2.05E+02≈	2.19E+02+	2.05E+02
	Std Dev (Rank)	1.69E+00(6)	7.37E+00(11)	**2.02E+00(1)**	6.63E+00(10)	5.44E+00(8)	5.87E+00(6)	9.67E-01(4)	1.21E+00(5)	8.23E+00(13)	9.75E-01(9)	1.78E+00(2)	3.41E+00(12)	1.62E+00(2)
f_{51}	Mean Error	1.04E+02+	1.80E+02+	**1.00E+02≈**	**1.00E+02≈**	1.08E+02+	**1.00E+02≈**	**1.00E+02≈**	1.04E+02+	1.14E+02+	1.01E+02+	**1.00E+02≈**	1.12E+02+	**1.00E+02**
	Std Dev (Rank)	1.98E+01(8)	4.07E+01(13)	**4.75E-02(1)**	**1.30E-01(1)**	2.76E+01(10)	**8.48E-02(1)**	**2.61E-02(1)**	1.82E+01(8)	3.34E+01(12)	7.37E-02(7)	**8.76E-02(1)**	3.28E+01(11)	**4.30E-02(1)**
f_{52}	Mean Error	4.93E+02+	7.87E+02+	1.56E+03+	4.55E+02-	5.22E+02+	7.13E+02+	3.47E+02-	3.34E+02-	4.51E+02-	7.76E+02+	4.92E+02+	**3.28E+02-**	4.88E+02
	Std Dev (Rank)	1.35E+02(8)	7.79E+01(12)	5.89E+01(13)	5.09E+01(5)	5.89E+01(9)	1.42E+02(10)	3.87E+01(3)	2.24E+01(2)	5.05E+01(4)	3.03E+02(11)	5.85E+01(7)	**2.09E+01(1)**	1.14E+02(6)
f_{53}	Mean Error	1.17E+03+	1.41E+03+	**3.87E+02-**	1.15E+03+	1.17E+03+	1.19E+03+	1.27E+03+	1.06E+03+	1.14E+03+	1.47E+03+	5.63E+03+	1.27E+03+	4.48E+02
	Std Dev (Rank)	5.49E+01(6)	9.15E+01(11)	**1.16E+01(1)**	1.26E+02(5)	4.30E+01(6)	5.96E+01(8)	5.23E+01(9)	6.01E+01(3)	5.86E+01(4)	9.75E+01(12)	4.27E+03(13)	5.86E+01(9)	9.21E+00(2)
f_{54}	Mean Error	1.49E+03+	1.47E+03+	**2.26E+02-**	8.77E+02+	9.16E+02+	8.74E+02+	6.59E+02+	1.99E+03+	9.30E+02+	1.71E+03+	8.62E+02+	1.40E+03+	3.69E+02
	Std Dev (Rank)	1.96E+02(11)	4.10E+02(10)	**1.13E+01(1)**	6.56E+01(6)	1.24E+02(7)	6.04E+01(5)	1.41E+02(3)	3.49E+02(13)	5.51E+01(8)	2.46E+02(12)	7.27E+01(4)	1.24E+02(9)	2.79E+02(2)
f_{55}	Mean Error	8.86E+03+	1.22E+04+	**1.06E+03-**	9.75E+03+	9.19E+03+	1.03E+04+	9.31E+03+	8.20E+03+	1.03E+04+	1.04E+04+	8.95E+03+	9.84E+03+	1.32E+03
	Std Dev (Rank)	5.16E+02(4)	2.55E+03(13)	**2.09E+02(1)**	7.51E+02(8)	4.29E+02(6)	1.05E+03(10)	7.39E+02(7)	2.99E+02(3)	7.74E+02(10)	1.09E+03(11)	6.01E+02(5)	2.46E+02(9)	8.26E+01(2)
Sum Rank		242	288	238	163	165	146	189	190	190	261	198	280	111
Aver Rank		8.07	9.6	7.93	5.43	5.5	4.87	6.3	6.33	6.33	8.7	6.6	9.33	3.7
+		27	29	21	18	20	17	20	22	19	28	18	27	
–		3	1	7	8	9	11	9	8	10	2	10	3	
≈		0	0	2	4	1	2	1	0	1	0	2	0	

and efficient DE variant by implementing three mutant strategies with different characteristics simultaneously, and MPEDE [32] is a recent variant of DE algorithm based on the concept of work specialization. In SinDE [17], the sinusoidal formulas is employed to adjust automatically the values of main parameters in DE, and CIPDE [18] employs the collective information of the best candidates to form a part of the difference vector in mutation and execute crossover. In CLPSO [7], the personal historical best information of all particles is employed to update the velocity of population, CMA-ES [44] is a very efficient and famous evolution strategy (ES), and GL-25 [8] is a hybrid real-coded genetic algorithm by combining the global and local searches. These algorithms are the better comparative or recently published in the literature, and thus chosen as the compared ones.

In these experiments, the average error (Mean Error) and standard deviation (Std Dev) of the results obtained by 25 runs independently are recorded for measuring their performances. The parameter settings of ISDE and all compared algorithms are listed in Table 7, where the compared algorithms take the same control parameter settings as their original papers, and parameters for ISDE are consistent with Section 3. Moreover, to have statistically sound conclusions, Wilcoxons rank sum test at 0.05 significance level and the rank of each algorithm are conducted on the experimental results.

4.3.1. Comparison with ten algorithms on CEC 2005

First, we make a comparison of ISDE with ten compared algorithms above (jDE [40], SaDE [23], EPSDE [24], JADE [14], CoDE [22], SHADE [21], MPEDE [32], CLPSO [7], CMA-ES [44] and GL-25 [8]) on 25 benchmark functions f_1-f_{25} from CEC 2005 [42] with $D = 30$. Table 8 reports the experimental results, Wilcoxons rank sum test results and rank, where the best results are marked by bold on each function, Sum Rank and Aver Rank are same as Table 1, “+”, “-” and “≈” denote that the performance of IDEI is better than, worse than, and similar to that of the corresponding algorithm respectively and the last three rows summarize them.

From Table 8, we see that for.

1) unimodal functions f_1-f_5, SHADE obtains the best results on f_2 and f_4, while CMA-ES gets the best results on f_3 and f_5. In particular, ISDE gets the similar best results to jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE and CLPSO on f_1. According to Wilcoxons rank sum test, ISDE is much better than jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE, CLPSO, CMA-ES and GL-25 on 3, 4, 4, 0, 3, 0, 1, 4, 3 and 5 test functions respectively, and slightly worse on 1, 0, 0, 4, 1, 4, 3, 0, 2 and 0 test functions, respectively. Then ISDE outperforms others except for JADE, SHADE and MPEDE. The reason for it is that the information exchanging and sharing mechanism enhances the diversity of population such that the convergence speed of algorithm is reduced for simple problems.
2) basic multimodal functions f_6-f_{12}, ISDE gets the best results on all functions except that CoDE, MPEDE and CMA-ES have the best results on f_8, f_{10} and f_{11}, respectively. According to Wilcoxons rank sum test, ISDE is much better than jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE, CLPSO, CMA-ES and GL-25 on 6, 7, 7, 5, 4, 4, 5, 6, 5 and 7 test functions respectively, and slightly worse on 0, 0, 0, 1, 2, 2, 1, 0, 2 and 0 test functions, respectively. Then ISDE has a more promising performance than other algorithms for these functions. This is because the information exchanging and sharing mechanism could effectively balance the exploitation and exploration by sharing the information of superior individuals with each other and exchanging the information of inferior individuals with the best individual or the randomly generated individual.
3) expanded multimodal functions f_{13}-f_{14}, ISDE obtains the best mean fitness value and standard deviation on all test functions.
4) hybrid composition functions f_{15}-f_{25}, the results of Wilcoxons rank sum test show that ISDE is much better than jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE, CLPSO, CMA-ES and GL-25 on 5, 6, 6, 6, 3, 7, 2, 6, 6 and 7 test functions respectively, and slightly worse on 0, 0, 5, 0, 1, 1, 2, 2, 2 and 1 test functions, respectively. Furthermore, one can see that jDE, SaDE and JADE have worse results than ISDE on all these functions, which is due to the fact that the better diversity enhanced by the information exchanging and sharing mechanism during evolutionary process is helpful to release individuals from local minima, and the proposed stochastic mixed mutation strategy improves the convergence speed during the later evolutionary process.

In summary, according to Wilcoxons rank sum test, the statistical results in Table 8 indicate that ISDE performs better than jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE, CLPSO, CMA-ES and GL-25 on 16, 19, 19, 13, 12, 13, 10, 18, 16 and 21 test functions respectively, slightly worse on 1, 0, 5, 5, 5, 7, 6, 2, 6 and 1 test functions respectively, and similar to 8, 6, 1, 7, 8, 5, 9, 5, 3 and 3 test functions, respectively. Meanwhile, ISDE and others get 63, 129, 152, 166, 97, 105, 97, 80, 176, 140 and 202 in term of total rank, and 2.52, 5.16, 6.08, 6.64, 3.88, 4.2, 3.88, 3.2, 7.04, 5.6 and 8.08 in term of average rank on all problems, respectively.

To see clearly, Fig. 5 draws the bar chart of the statistical results for ISDE and other compared algorithms on all functions from CEC 2005 with $D = 30$, where the blue and red bars represent the average rank and the number of function obtained the best results, respectively. From Fig. 5, we see that ISDE has the best average rank and the most best results for all functions. Furthermore, Figs. 6 and 7 depict the evolutionary curves of ISDE, jDE, SaDE, EPSDE, JADE, CoDE, SHADE, CLPSO and GL-25 on f_1-f_{14}. From Figs. 6 and 7, we see that for unimodal functions f_1-f_5, ISDE has a slower convergence speed than SHADE on f_1-f_5, JADE on f_2-f_5, EPSDE on f_1-f_2, SaDE on f_1, and jDE and CoDE on f_5, which is because the enhanced diversity by the information exchanging and sharing mechanism reduces its exploitation capability, while for multimodal functions f_6-f_{14}, ISDE has a faster convergence than others on six functions f_6-f_7, f_9 and f_{12}-f_{14} except for CoDE and SHADE on f_8, JADE and SHADE on f_{10} and CoDE on f_{11}, which might be because the proposed information exchanging and sharing mechanism and stochastic mixed mutation strategy could effectively balance exploitation and exploration during evolutionary process. Therefore, ISDE has better performance than them on CEC 2005 test functions.

4.3.2. Comparison with twelve algorithms on CEC 2014

Now, we shall make a comparison of ISDE with jDE [40], SaDE [23], EPSDE [24], JADE [14], CoDE [22], SHADE [21], MPEDE [32], SinDE [17], CIPDE [18], CLPSO [7], CMA-ES [44] and GL-25 [8] on 30 benchmark functions f_{26}-f_{55} from CEC 2014 [43] with $D = 30$ and 50. Tables 10 and 11 report their experimental results, Wilcoxons rank sum test results and rank on all problems with $D = 30$ and 50 respectively, where the best results are marked by bold on each function, Sum Rank and Aver Rank are same as Table 1, “+”, “-” and “≈ ”are same as Table 8 and their last three rows summarize them.

When $D = 30$, from Table 9, we see that for.

1) unimodal functions f_{26}-f_{28}, CMA-ES obtains the best results on all functions, and ISDE and jDE get the same results as CMA-ES on f_{27} and f_{28}, respectively. The reason for it is that the evolution path added in CMA-ES is helpful to improve the quality of evaluation.
2) simple multimodal functions f_{29}-f_{41}, ISDE attains the best performances on five functions f_{32}, f_{33}, f_{36}, f_{39} and f_{41}. In particular, CMA-ES gets the best results on f_{29}-f_{30}, SinDE obtains the results on f_{31}, jDE, JADE and SinDE gets the same best performance as ISDE on f_{32}, jDE, SaDE, JADE, CoDE, CIPDE and CLPSO obtains the same best results as ISDE on f_{33}, CIPDE obtains the best results on f_{34} and f_{38}, jDE has the best results on f_{35}, and CoDE and SHADE get the best performances on f_{37} and f_{40}, respectively. According to Wilcoxons rank sum test, ISDE is much better than jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE, SinDE, CIPDE, CLPSO, CMA-

Table 11
CPU time of ISDE, DE, EPSDE and SaDE.

Function	unimodal		multimodal		
	f_4	f_{26}	f_{11}	f_{31}	f_{34}
DE	18.17 s	19.00 s	77.44 s	54.64 s	16.50 s
EPSDE	24.45 s	24.39 s	85.06 s	60.31 s	23.10 s
SaDE	52.64 s	56.00 s	112.37 s	88.69 s	54.80 s
ISDE	26.09 s	24.71 s	84.90 s	61.45 s	25.49 s

ES and GL-25 on 10, 11, 12, 5, 8, 7, 12, 9, 8, 12, 10 and 13 test functions respectively, and slightly worse on 1, 1, 1, 5, 4,6, 1, 3, 4, 0, 3 and 0 test functions, respectively. Then ISDE has a more competitive performance among them, which is because the exploration and exploitation of ISDE are balanced effectively by the proposed information exchanging and sharing mechanism and the stochastic mixed mutation strategy during evolutionary process.

3) hybrid multimodal functions f_{42}-f_{47}, ISDE gets the best performances on all functions except that MPEDE obtains the best performances on f_{42} and f_{43}, CIPDE has the best results on f_{44}.
4) composition multimodal functions f_{48}-f_{55}, ISDE obtains the best results on f_{48}, f_{51} and f_{55}, and ISDE and jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE, SinDE, CIPDE, CLPSO, CMA-ES and GL-25 get 17, 45, 83, 32, 37, 43, 46, 34, 36, 43, 64, 68 and 41 in term of total rank, respectively. Moreover, the results of Wilcoxons rank sum test show that ISDE is much better than jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE, SinDE, CIPDE, CLPSO, CMA-ES and GL-25 on 7, 8, 3, 7, 7, 7, 6, 5, 6,6, 8 and 5 test functions respectively, and slightly worse on 0, 0, 3, 0, 0, 0, 1,2, 1, 0, 0 and 2 test functions, respectively. Then ISDE has a more promising performance than other on these functions. This is because the proposed information exchanging and sharing mechanism and stochastic mixed mutation strategy could effectively balance the exploration and exploitation.

According to Wilcoxons rank sum test, the statistical results in Table 9 show that ISDE performs better than jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE, SinDE, CIPDE, CLPSO, CMA-ES and GL-25 on 23, 27, 22, 20, 21, 18, 22, 20, 21, 26, 23 and 26 test functions respectively, similar to 4, 2, 3, 4, 3, 2, 2, 3, 2, 3, 1 and 1 test functions respectively, and slightly worse on 3, 1, 5, 6, 6, 10, 6, 7, 7, 1, 6 and 3 test functions, respectively. Meanwhile, ISDE and others get 91, 189, 201, 261, 151, 125, 150, 175, 171, 216, 260, 253 and 261 in term of total rank and 3.03, 6.3, 6.7, 8.7, 5.03, 4.17, 5, 5.83, 5.7, 7.2, 8.67, 8.43 and 8.7 in term of average rank, respectively. To see clearly, Fig. 8 draws the bar chart of the statistical results for ISDE and other compared algorithms on all functions from CEC 2014, where the blue and red bars represent the average rank and the number of function obtained the best results, respectively. From Fig. 8 (a), we see that ISDE has the best average rank and the most best results on these functions. Therefore, ISDE has better performance than jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE, SinDE, CIPDE, CLPSO, CMA-ES and GL-25 on CEC 2014 contest test instances.

When $D = 50$, from Table 10, we see that for unimodal functions f_{26}-f_{28}, CMA-ES obtains the best results on all functions; for simple multimodal functions f_{29}-f_{41}, ISDE attains the best performances on four functions f_{35}, f_{38}, f_{39} and f_{41}, SinDE obtains the best results on f_{31}, f_{32} and f_{33}, CIPDE gets the best performances on f_{29}, f_{33} and f_{41}, CMA-ES gets the best results on f_{30} and f_{40}, and CoDE and SHADE obtains the best performances on f_{37} and f_{36} respectively; for hybrid multimodal functions f_{42}-f_{47}, ISDE gets the best performances on f_{45} and f_{47}, CoDE gets the best performance on f_{44}, and MPEDE obtains the best results on f_{42}, f_{43} and f_{46}; and for composition multimodal functions f_{48}-f_{55}, ISDE obtains the best results on f_{48}, f_{49} and f_{51}, and has the similar performance to EPSDE on f_{48} and f_{51}, while EPSDE gets the best results on f_{48}, f_{50}, f_{51} and f_{53}-f_{55}, and GL-25 obtains the best performance on f_{52}.

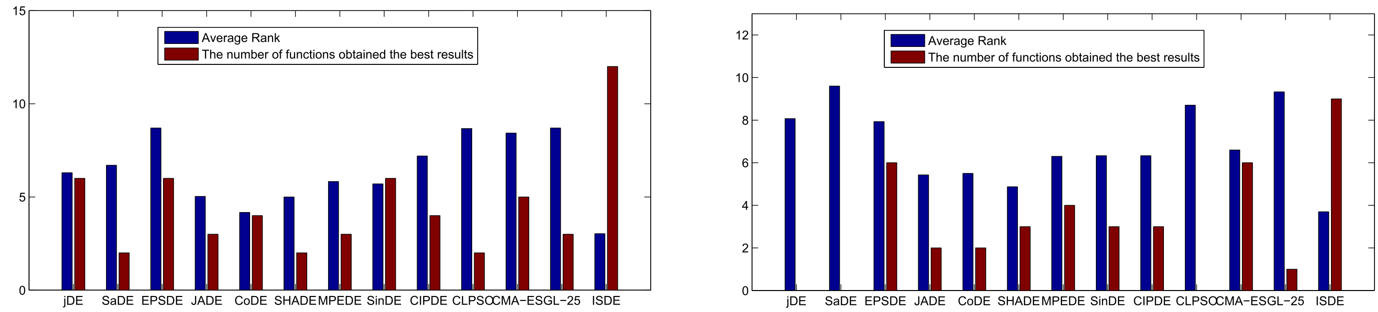
According to Wilcoxons rank sum test, the statistical results in Table 10 show that ISDE performs better than jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE, SinDE, CIPDE, CLPSO, CMA-ES and GL-25 on 27, 29, 21, 18, 20, 17, 20, 22, 19, 28, 18 and 27 test functions respectively, similar to 0, 0, 2, 4, 1, 2, 1, 0, 1, 0, 2 and 0 test functions respectively, and slightly worse on 3, 1, 7, 8, 9, 11, 9, 8, 10, 2, 10 and 3 test functions, respectively. Meanwhile, ISDE and others get 111, 242, 288, 238, 163, 165, 146, 189, 190, 190, 261, 198 and 280 in term of total rank and 3.7, 8.07, 9.6, 7.93, 5.43, 5.5, 4.87, 6.3, 6.33, 6.33, 8.7, 6.6 and 9.33 in term of average rank, respectively. Furthermore, Fig. 8 (b) shows that ISDE has the best average rank and the most best results on these functions. The reason for these might be that the proposed information exchanging and sharing mechanism could enhance the population diversity to release from local minima and the proposed stochastic mixed mutation strategy could improve the convergence speed. Therefore, ISDE has better performance than jDE, SaDE, EPSDE, JADE, CoDE, SHADE, MPEDE, SinDE, CIPDE, CLPSO, CMA-ES and GL-25 on these instances.

It should be mentioned that the reason for ISDE performing better than other algorithms is mainly attributed to the fact that the incorporation of the cosine perturbation and the information intercrossing and sharing mechanism could effectively enhance the ability of ISDE to jump out of local optimum and make good use of the information of different individuals. But the information intercrossing and sharing mechanism might result in the slow convergence on some unimodal functions at the same time. Thus, ISDE has better performance than others on most complicated functions, and worse results on some unimodal functions. Overall, ISDE is a more promising algorithm among them.

4.4. Algorithm efficiency

In order to show the efficiency of ISDE, a comparison of ISDE with the classical DE, EPSDE and SaDE is conducted on five typical functions with $D = 30$, including unimodal functions f_4 and f_{26}, and basic multimodal functions f_{11}, f_{31} and f_{34}. In particular, DE/rand/1 and binomial crossover are used in the classical DE, and both the scaling factor and crossover rate are set to 0.5. Moreover, in this experiment, the maximum number of function evaluation $FES_{\max}$ is set to $10000D$, and the average CPU time with 25 independent runs is recorded to evaluate their performances. Table 11 reports the numerical results obtained by them.

From Table 11, we see that ISDE is faster than SaDE, slower than DE and similar to EPSDE. Unlike the classical DE, the proposed algorithm requires to sort the population in both the stochastic mixed mutation strategy and the information intercrossing and sharing mechanism. Then, ISDE takes a slightly longer time than the classical DE. Overall, numerical results show that ISDE is a promising algorithm.



(a) $D = 30$. (b) $D = 50$.

Fig. 8. Statistical results of ISDE and its counterparts on CEC 2014. (a) $D = 30$, (b) $D = 50$.

Table 12
Numerical results of ISDE on PEFM.

Function	Best(Result)	Worst(Results)	Average value	Standard deviation
EPSDE	3.76E+00	1.29E+01	1.00E+01	2.50E+00
jDE	3.06E+00	1.25E+01	7.37E+00	3.01E+00
SaDE	**0.00E+00**	6.61E+00	9.12E-01	2.11E+00
CLPSO	1.36E+01	2.65E+01	2.15E+01	4.55E+00
GL-25	**0.00E+00**	1.71E+01	8.83E+00	6.69E+00
ISDE	**0.00E+00**	**4.78E-01**	2.28E-02	**9.64E-02**

4.5. Application

As an application, we consider Parameter Estimation for Frequency-Modulated Sound Waves (PEFM) in Ref. [48]. This problem has an important role in several modern music systems and aims to generate a sound similar to target sound, and can be modeled as a six dimensional optimization problem:

$$\min\ f(\vec{X}) = \sum_{t=0}^{100} (y(t) - y_0(t))^2, \tag{23}$$

where $\vec{X} = (a_1, \omega_1, a_2, \omega_2, a_3, \omega_3)$,

$y\,(t) = a_1 \cdot \sin\,(\omega_1 \cdot t \cdot \theta + a_2 \cdot \sin\,(\omega_2 \cdot t \cdot \theta + a_3 \cdot \sin\,(\omega_3 \cdot t \cdot \theta))),$

and

$y_0\,(t) = \sin\,(5 \cdot t \cdot \theta + 1.5 \cdot \sin\,(4.8 \cdot t \cdot \theta + 2 \cdot \sin\,(4.9 \cdot t \cdot \theta))).$

Obviously, it is a highly complex and multimodal problem, and has minimum value $f(\vec{X}_{sol}) = 0$.

We use ISDE to solve this problem. Meanwhile, five existing algorithms EPSDE, jDE, SaDE, CLPSO and GL-25 are also employed to make a comparison. Table 12 reports their numerical results over 25 runs independently with $FES_{\max} = 60000$, where the best results are marked by bold. From Table 12, we see that ISDE gets the best performance among them. In particular, ISDE obtains the same result as SaDE and GL-25 in term of the Best value. Thus, ISDE can solve effectively this problem.

5. Conclusion

This paper presents a novel differential evolution algorithm by developing a stochastic mixed mutation strategy and an information intercrossing and sharing mechanism to avoid the premature convergence and enhance the information dissemination between subpopulations. To effectively balance the exploitation and exploration, a stochastic mixed mutation strategy is first proposed by incorporating the cosine perturbation into the probability parameter setting and combining two mutation strategies with this probability. In the proposed mutation strategy, the cosine perturbation enhances the randomness of choosing mutation strategy during the evolutionary process and the possibility of using the mutation strategy with better exploration during the later evolutionary process, and the probability setting is only related to the number of iteration. Next, an information intercrossing and sharing mechanism is developed to make full use of the information of individuals. In the proposed mechanism, the population is first divided into the superior and inferior subpopulations according to the fitness value, and then the superior individuals share their information with each other by the opposite operation, while the inferior individuals exchange their information with the best individual or the one randomly generated from search space by the binomial crossover operation according to their fitness values. Meanwhile, a simple and efficient approach is applied to adjust control parameters. Compared with the existing DE variants, the proposed stochastic mixed mutation strategy could adapt different evolutionary stages and complex problems without requiring extra store spaces or computational costs, and the proposed information intercrossing and sharing mechanism not only keeps the concept of work specialization, but also promotes the information dissemination between different subpopulations. Then the proposed algorithm could effectively balance the exploration and exploitation, and makes good use of the information of different individuals. Finally, a comparison of the proposed algorithm with twelve typical algorithms is conducted to evaluate its performance on 55 test functions from both CEC2005 and CEC2014. The experimental results show that the proposed algorithm is very competitive. Furthermore, the proposed algorithm is applied to Parameter Estimation for Frequency-Modulated Sound Waves.

Further research can be focused on designing an adaptive manner to set the extra parameters in the proposed algorithm, developing a more suitable adaptation method for population size to further enhance its performance, and applying it to other complex optimization scenarios and real-world optimization problems.

Acknowledgment

The authors would like to thank the Handling Editor and three anonymous referees for their valuable comments and suggestions on earlier versions of this paper.

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.swevo.2017.12.010.

References

[1] R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[2] T. Rogalsky, R.W. Derksen, S. Kocabiyik, Differential evolution in aerodynamic optimization, in: Proc. 46th Annu. Conf. Can. Aeronautics Space Inst, 1999, pp. 29–36.
[3] S. Das, A. Konar, Design of two dimensional IIR filters with modern search heuristics: a comparative study, Int. J. Comput. Intell. Appl. 6 (3) (2006) 329–355.
[4] F.S. Wang, H.J. Jang, Parameter estimation of bio-reaction model by hybrid differential evolution, in: Proc. 2000th Congr. Evol. Comput., IEEE Press, Piscataway, NJ, 2000, pp. 410–417.
[5] M. Omran, A.P. Engelbrecht, A. Salman, Differential evolution methods for unsupervised image classification, in: Proc. 7th Congr. Evol. Comput., IEEE Press, Piscataway, NJ, 2005, pp. 966–973.
[6] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.
[7] J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.
[8] C. Garcia-Martinez, M. Lozano, F. Herrera, D. Molina, A.M. Sanchez, Global and local real-coded genetic algorithm based on parent-centric crossover operators, Eur. J. Oper. Res. 185 (3) (2008) 1088–1113.
[9] A.P. Piotrowski, Review of differential evolution population size, Swarm and Evol. Comput (2017) 1–24.
[10] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution-An updated survey, Swarm and Evol. Comput (2016) 1–30.
[11] A.W. Mohamed, An improved differential evolution algorithm with triangular mutation for global numerical optimization, Comput. Ind. Eng. (2015) 359–375.
[12] M. Yang, C.H. Li, Z.H. Cai, J. Guan, Differential evolution with auto-enhanced population diversity, IEEE Trans. Comput. 45 (2) (2015) 302–315.
[13] Y.L. Li, Z.H. Zhan, Y.J. Gong, W.N. Chen, J. Zhang, Y. Li, Differential evolution with an evolution path: a DEEP evolutionary algorithm, IEEE Trans. Comput. 45 (9) (2015) 1798–1810.
[14] J.Q. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.
[15] W.C. Yi, Y.Z. Zhou, L. Gao, X.Y. Li, J.H. Mou, An improved adaptive differential evolution algorithm for continuous optimization, Expert Syst. Appl. 44 (2016) 1–12.
[16] P. Rakshit, A. Konar, Differential evolution for noisy multiobjective optimization, Artif. Intell. (2015) 165–189.
[17] A. Draa, S. Bouzoubia, I. Boukhalfa, A sinusoidal differential evolution algorithm for numerical optimization, Appl. Soft Comput. (2015) 99–126.
[18] L.M. Zheng, S.X. Zhang, K.S. Tang, S.Y. Zheng, Differential evolution powered by collective information, Inf. Sci. (2017) 13–29.
[19] L.X. Tang, Y. Dong, J.Y. Liu, Differential evolution with an individual-dependent mechanism, IEEE Trans. Evol. Comput. 19 (4) (2015) 560–574.
[20] M.N. Tian, X.B. Gao, C. Dai, Differential evolution with improved individual-based parameter setting and selection strategy, Appl. Soft Comput. (2017) 286–297.
[21] R. Tanable, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in: Proc. IEEE Congr. Evol. Comput., 2013, pp. 71–78.
[22] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (1) (2011) 55–66.
[23] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.
[24] R. Mallipeddi, P.N. Suganthan, Q.K. Pan, M.F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (2) (2011) 1679–1696.
[25] W.Y. Gong, A.M. Zhou, Z.H. Cai, A multioperator search strategy based on cheap surrogate models for evolutionary optimization, IEEE Trans. Evol. Comput. 19 (5) (2015) 746–758.
[26] Y. Xue, Y. Zhuang, J.J. Gu, X.M. Chang, Z. Wang, Strategy selecting problem of self-adaptive discrete differential evolution algorithm, J. Software 25 (5) (2014) 984–996.
[27] W.Y. Gong, Z.H. Cai, C.X. Ling, H. Li, Enhanced differential evolution with adaptive strategies for numerical optimization, IEEE Trans. Syst. Man Cyber. Part B 41 (2) (2011) 397–413.
[28] W.L. Xiang, X.L. Meng, M.Q. An, Y.Z. Li, M.X. Gao, An enhanced differential evolution algorithm based on multiple mutation strategies, Comput. Intell. Neurosci. 2015 (2015) 1–15 285730.
[29] M.Z. Ali, N.H. Awad, P.N. Suganthan, R.G. Reynolds, An adaptive multipopulation differential evolution with dynamic population reduction, IEEE Trans. Comput. 99 (2016) 1–12.
[30] L.Z. Cui, G.H. Li, Q.Z. Lin, J.Y. Chen, N. Lu, Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations, Comput. Oper. Res. (2016) 155–173.
[31] M.F. Han, S.H. Liao, J. Chang, C.T. Lin, Dynamic group-based differential evolution using a self-adaptive strategy for global optimization problems, Appl. Intell. (2013) 41–56.
[32] G.H. Wu, R. Mallipeddi, P.N. Suganthan, R. Wang, H.K. Chen, Differential evolution with multi-population based ensemble of mutation strategies, Inf. Sci. (2016) 329–345.
[33] X.C. Zhao, W.Q. Lin, C.C. Yu, J. Chen, S.G. Wang, A new hybrid differential evolution with simulated annealing and self-adaptive immune operation, Comput. Math. Appl. (2013) 1948–1960.
[34] D.H. Tran, M.Y. Cheng, M.T. Cao, Hybrid multiple objective artificial bee colony with differential evolution for the time-cost-quality tradeoff problem, Knowl. Base Syst. 74 (2015) 176–186.
[35] Y.J. Zheng, X.L. Xu, H.F. Ling, S.Y. Chen, A hybrid fireworks optimization method with differential evolution operators, Neurocomputing (2015) 75–82.
[36] N.H. Awad, M.Z. Ali, P.N. Suganthan, CADE: a hybridization of cultural algorithm and differential evolution for numerical optimization, Inf. Sci. 378 (2017) 215–241.
[37] A. Qamar, A. Jamil, J. Hajira, A novel tournament selection based differential evolution variant for continuous optimization problem, Math. Probl Eng. (2015) 1–21.
[38] S. Rahnamayan, H.R. Tizhoosh, M.M. Salama, Opposition-based differential evolution (ODE) with variable jumping rate, Proc. IEEE Symp. Fundat. Comput. Intell. 118 (2) (2011) 401–408.
[39] W.H. Wei, J.L. Zhou, M. Tao, H.Q. Yuan, Constrained differential evolution using opposition-based learning, Acta Electron. Sin. 44 (2) (2016) 426–436.
[40] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657.
[41] S. Das, A. Abraham, U.K. Chakraborty, A. Konar, Differential evolution using a neighborhood based mutation operator, IEEE Trans. Evol. Comput. 13 (3) (2009) 526–553.
[42] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger, S. Tiwari, Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-parameter Optimization, Nanyang Technol. Univ., Singapore, IIT Kanpur, India, May 2005. Tech. Rep. KanGAL, 2005005.
[43] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-parameter Numerical Optimization, Technical report, Nanyang Technol. Univ., China: Zhenzhou Univ., Singapore, 2013.
[44] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195.
[45] C. Qian, K. Tang, Z.H. Zhou, Selection hyper-heuristics can provably be helpful in evolutionary multi-objective optimization, in: Proceedings of the 14th International Conference on Parallel Problem Solving from Nature (PPSN'16), Edinburgh, Scotland, 2016, pp. 835–846.
[46] M.Z. Ali, N.H. Awad, P.N. Suganthan, Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization, Appl. Soft Comput. (2015) 304–327.
[47] S. Das, A. Konar, U.K. Chakraborty, Two improved differential evolution schemes for faster global search, Proc. IEEE Congr. Evol. Comput (2005) 991–998.
[48] S. Das, P.N. Suganthan, Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Roblems, Jadavpur Univ., India and Nanyang Technological Univ, Singapore, 2010 (Technical report).