Applied Soft Computing 56 (2017) 286–297

Contents lists available at ScienceDirect

Applied Soft Computing

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

# Differential evolution with improved individual-based parameter setting and selection strategy☆

Mengnan Tian[a], Xingbao Gao[a,*], Cai Dai[b]

[a] School of Mathematics and Information Science, Shaanxi Normal University, Xi'an, Shaanxi 710119, China
[b] School of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi 710119, China

ARTICLE INFO

*Article history:*
Received 12 March 2016
Received in revised form 6 October 2016
Accepted 10 March 2017
Available online 18 March 2017

*Keywords:*
Differential evolution
Global optimization
Combined mutation strategy
Parameters setting
Selection strategy

ABSTRACT

In this paper, a novel differential evolution (DE) algorithm is proposed to improve the search efficiency of DE by employing the information of individuals to adaptively set the parameters of DE and update population. Firstly, a combined mutation strategy is developed by using two mixed mutation strategies with a prescribed probability. Secondly, the fitness values of original and guiding individuals are used to guide the parameter setting. Finally, a diversity-based selection strategy is designed by assembling greedy selection strategy and defining a new weighted fitness value based on the fitness values and positions of target and trial individuals. The proposed algorithm compares with eight existing algorithms on CEC 2005 and 2014 contest test instances, and is applied to solve the Spread Spectrum Radar Polly Code Design. Experimental results show that the proposed algorithm is very competitive.



## 1. Introduction

Over the last two decades, global optimization problems have attracted a great interest of researchers and many nature-inspired intelligent algorithms for them have been developed, such as genetic algorithm [1], differential evolution [2], particle swarm optimization [3], artificial bee colony algorithm [4], fireworks algorithm [5] and tabu search algorithm [6]. Among them, DE algorithm has been shown to be an accurate, reasonably fast and robust optimizer for optimization problems [7] and has wide application in many fields such as mechanical engineering design [8], signal processing [9], chemical engineering [10], pattern recognition [11] and so on. However, similar to other stochastic search methods, DE does not guarantee the convergence to global optimum, especially for complicated problems [12,13].

As pointed out in [14–16], the performance of DE mainly depends on trial vector generation strategies including mutation and crossover operators, and control parameters such as population size *NP*, scale factor *F*, crossover rate *CR* and so on. Then, many novel mutation operators [19–21,24–27] and parameter settings [7,17,18] were developed to improve the performance of DE. In particular, the most popular approaches are to set parameters to constant [7], random [17] or adaptive methods [18], and to balance its exploration and exploitation by the single mutation operator strategies with composite search features [19–21] and the multi-mutation operators strategies with different search features [24–27].

To improve the exploration and exploitation of single mutation strategy, Epitropakis et al. [19] employed a line combination of an explorative operator and an exploitive one, and Das et al. [20] used a weighted sum of a global mutant individual and a local one. Even though each of them has a simple structure, and the combined or weighted coefficient is self-adaptive, mutation operators in them are problem dependent and difficult to determine. Another way is to control or choose difference vector of single mutation operator [21–23]. However, a large amount of extra store spaces [21,22] or computational costs [23] are required during the mutation process. Meanwhile, many multi-mutation operators [24–28] are developed to improve the performance of DE. By using three mutation strategies in parallel to generate offspring, Wang et al. [24] proposed a DE with composite trial vector generation strategies and control parameters (CoDE). A self-adaptive DE algorithm (SaDE) [25] is designed by using a self-adaptive probability to choose one trial vector generation strategy to generate offspring from four given strategies, and is enhanced in [26] by a teaching and learning mechanism. Mallipeddi et al. [27] proposed a DE with ensemble of parameters and mutation strategies

☆ The work is supported by the National Natural Science Foundations of China (Nos. 61273311 and 61502290).

* Corresponding author.
*E-mail addresses:* mengnan_tian@snnu.edu.cn (M. Tian), xinbaog@snnu.edu.cn (X. Gao), cdai0320@snnu.edu.cn (C. Dai).

http://dx.doi.org/10.1016/j.asoc.2017.03.010
1568-4946/© 2017 Published by Elsevier B.V.

(EPSDE), which employs a set of mutation strategies along with a set of parameter values that compete to generate offspring, and is further improved in [28] by incorporating a SaDE type learning scheme. Even though these multi-mutation operators have made great progress on improving the performance of DE, they are at the expense of increasing store spaces or computational costs. More importantly, they might not obtain a promising result for complicated problems since mutation operators are prescribed and the difference of individuals are not considered during the mutation process.

Differing from the approaches above, the guiding-based idea [29] is also employed to improve the search ability of DE since it can make full use of the information of individuals during the evolutionary process. Recently, DE with an individual-dependent mechanism (IDE) [36] employs four guiding-based (mixed) mutation operators to generate offspring, but it is hard to determine the generation index threshold to divide the evolution process into prior or later stages, and the method to separate the superior and inferior subpopulations is more complicated. In summary, one can see that there are still the following challenges: (1) it is often hard and time-consuming to set extra parameters except for *F*, *NP* and *CR*; (2) it is difficult to prescribe the mutation strategies in single and multi-operators approaches; and (3) for the multi-operators approach, the more mutation strategies are combined, the better performance may be obtained, while extra cost would be paid. Therefore it is necessary to develop a new mutation strategy.

For parameter settings, the constant [7], random [17] and adaptive methods [18] are widely applied. Obviously, a fixed value cannot fit all problems and requires more time to tune, the random methods could not adapt different evolutionary process even if they can avoid to tune the parameters and improve the robustness of DE. Meanwhile, the adaptive methods in [18,32–35] can dynamically adjust parameters, yet they require a large amount of extra store spaces and expensive cost to compute. It should be noted that the fitness value of individual is rarely used to set parameters in the above methods. Tang et al. [36] set control parameters by means of the differences of their fitness values for mixed mutation strategy. But it cannot accurately control the size of difference vector since the information of base (origin) individual is only considered. Therefore, it is vital to design a new parameter setting method by using the information of individuals.

Furthermore, it is well known that selection strategy plays an important role in evolutionary process, and greedy selection [2] and tournament selection [37] are widely applied in DE. In contrast to greedy selection strategy, tournament selection strategy is more helpful to enhance population diversity during selection process. But the evaluation criterions for each one only consider the fitness value of individual such that the search will become inefficient since a lot of exploration information is wasted before a better solution is found. So it is also necessary to design a new selection strategy to improve the search by using a lot of exploration information.

Based on above discussions, this paper presents a differential evolution with improved individual-based parameter setting and selection strategy (IDEI). In particular, we propose (a) a combined mutation (CM) strategy based on two mixed mutation strategies with a prescribed constant probability; (b) a guiding individual-based parameter (GIP) setting for mixed mutation strategy via the fitness values of original and guiding individuals simultaneously; and (c) a diversity-based selection (DS) strategy by defining a weighted fitness value. Compared with the strategy in [36], CM strategy has a simple structure, does not require to set the generation index threshold to divide the evolution stage into earlier and later stages and a complicated method to divide population into the superior and inferior subpopulations. Meanwhile, unlike [36], GIP setting not only makes the search in a suitable area, but also employs information from parent and mutant individuals to generate offspring. Furthermore, differing from greedy and tournament selection strategies, DS strategy not only enhances the population diversity, but also alleviates a waste of a large amount of information before a better solution is found. Then the proposed algorithm can effectively balance the exploration and exploitation during the evolution process. Finally, numerical experiments are carried out to evaluate the performance of IDEI by comparing with four DE algorithms and four non-DE algorithms on 55 benchmark functions from both CEC 2005 [40,41] and CEC 2014 [48]. Moreover, IDEI is applied to Spread Spectrum Radar Polly phase Code Design (SSRPPCD) [47].

The rest of this paper is organized as follows. Section 2 presents the proposed algorithm. Experimental results are reported in Section 3. Finally, some concluding remarks are drawn in Section 4.

## 2. Proposed DE

Similar to other evolutionary algorithms (EAs), the performance of DE is influenced heavily by parameter settings and evolutionary operators: mutation, crossover and selection. Then we shall propose a DE by designing a combined mutation strategy, a guiding individual-based parameter setting method and a diversity-based selection strategy in this section.

### 2.1. Combined mutation strategy

In general, mutation operations of DE aim at controlling the search direction. Then the quality of mutation operation is very important for DE. As pointed out in [36], four mutation strategies, DE/current/1, DE/best/1, DE/rand/1 and DE/current-to-best/1, are widely applied in literatures. The single base vector mutation strategy DE/current/1 has good diversity and global search ability, but it is aimless and may take a long time to obtain a promising solution since the information of good solutions are not sufficiently used during last evolution. DE/best/1 is implemented as a local search around the best individual and has good convergence, yet it easily traps in local minima and leads to premature convergence. Because the base vector is randomly chosen from current population, DE/rand/1 is considered as a balance between DE/current/1 and DE/best/1. Then the possibility of a promising individual being selected is very low and it couldn't cope with stagnation in most cases for multimodal problems. DE/current-to-best/1 is thought as a mixed strategy due to the composite base vector generated by using a current individual and a best individual as origin individual and guiding individual, respectively. In contrast to three single base vector mutation strategies above, DE/current-to-best/1 uses only the best individual as an indicator to guide the individual to move along a promising direction, and thus it may reduce diversity and lead to premature convergence. By using guiding-based idea for each individual, mixed strategy with composite base vector, named DE/origin-to-guiding/differ, is proposed in [36]. This strategy not only provides a promising moving direction, but also focuses on the diversity of population. Furthermore, four mixed mutation strategies are defined to generate mutant individuals according to different evolution stages divided by generation index threshold and different original individuals from superior subpopulation or inferior subpopulation in [36]. Even though they are superior over others, it is difficult to determine generation index threshold, and the method dividing the superior and inferior subpopulations is more complicated.

To improve these shortcomings above, we propose the following combined mutation strategy by using two mixed mutation strategies with a predefined constant probability:

$$\vec{v}_{i,G} = \begin{cases} \vec{x}_{rand,G} + F_1(\vec{x}_{g,G} - \vec{x}_{rand,G}) + F_2(\vec{x}_{r1,G} - \vec{d}_{r2,G}), & \text{if } \ rand < \xi_1 \\ \vec{x}_{cur,G} + F_1(\vec{x}_{g,G} - \vec{x}_{cur,G}) + F_2(\vec{x}_{r1,G} - \vec{d}_{r2,G}), & \text{otherwise} \end{cases} \quad (1)$$

where $F_1$ and $F_2$ are scale factors, $\vec{x}_{r1,G}$ is a random individual with $r1 \neq i$, $\xi_1$ is a prescribed factor to determine which mixed mutation strategy is employed, $\vec{x}_{g,G}$ is guiding individual, origin individuals $\vec{x}_{cur,G}$ and $\vec{x}_{rand,G}$ are current and a randomly selected individuals from the population at $G$ generation, respectively. The aim of the first strategy DE/rand-to-guiding/1 in (1) is to improve convergence, while the second strategy aims at enhancing the search ability. Then a big $\xi_1$ may reduce the diversity of population, while a small $\xi_1$ cannot improve effectively the convergence. In particular, $\xi_1$ should be bigger for complicated problems since stagnation often occur and the potential individuals should be exploited to improve DE performance [30,31,42]. Thus, let $\xi_1 = 0.05$ and $\xi_1 = 0.2$ for simple and complicated problems respectively in this paper, and they are shown to be suitable choices by experiments in Section 3.1

According to [36], $\vec{d}_{r2,G}$ in (1) represents a hybrid individual determined by

$$\vec{d}^{j}_{r2,G} = \begin{cases} L^j + rand(0,1) \times (U^j - L^j), & \text{if } \ rand < \xi_2 \\ \vec{x}^{j}_{r2,G}, & \text{otherwise} \end{cases}$$

where $\vec{x}_{r2,G}$ is a random individual with $r2 \neq r1$ and $r2 \neq i$, $U^j$ and $L^j$ are lower and upper bounds of the $j$th element respectively, $\xi_2$ is the probability of generating components in the whole search space given by

$$\xi_2 = \frac{1}{100}(1 + 9 \times 10^{5(t-1)})$$

where $t = G/G_{\max}$ with $G$ and $G_{\max}$ being the current generation and maximum generation numbers, respectively. Obviously, $\vec{d}_{r2,G}$ can enhance the global search during evolution process.

To state the selection rule of guiding individual $\vec{x}_{g,G}$, let $P(t) = 1 - t^3$, $Pop_s$ and $Pop_g$ be the superior individual set of top 10% best individuals and general individual set of top $P(t) \times 100\%$ best individuals respectively, and their size are rounded down to integral number. The success ratio [42] below is used to express the evolutionary situation at each generation $G$:

$$SR_G = \frac{NS_G}{NP} \quad (2)$$

where $NS_G$ is the number of offspring individuals that can successfully enter the next generation.

It is well know that premature convergence and stagnation have significant influence on the performance of DE, and they occur whenever no or little good solutions will be generated. Then success ratio $SR_G$ will tend or close to zero whenever evolution process traps in premature convergence or stagnation situation [31,38,42]. To avoid this situation, let

$$\vec{x}_{g,G} \text{ be randomly selected from } \begin{cases} Pop_s, & \text{if } \ SR_{G-1} < \xi_3 \\ Pop_g, & \text{otherwise,} \end{cases}$$

where $\xi_3$ is a prescribed threshold. Obviously, $\xi_3$ is key to the performance of (1), a big $\xi_3$ will reduce the exploration of algorithm since more guiding individuals are selected from $Pop_s$, while a small $\xi_3$ cannot effectively avoid the premature convergence and stagnation situations. Thus, let $\xi_3 = 0.05$, which is shown to be a suitable choice by experiments in Section 3.1.

Because $SR_G$ is big at earlier stage and reduces gradually during evolution process, $\vec{x}_{g,G}$ is selected from large search space at earlier stage and better individuals at later stage. Then CM strategy has good exploration ability at earlier stage and exploitation ability at later stage. In contrast to the mutant strategy in [36], CM strategy employs only a simple combination of DE/rand-to-guiding/1 and DE/current-to-guiding/1 to balance exploration and exploitation, does not require the generation index threshold to divide the evolution stage into earlier and later stage and a complicated method to separate the superior and inferior subpopulations.

### 2.2. Guiding individual-based parameter setting

As control parameters of DE aim at controlling the size of the search area, suitable parameters will enhance its performance. In the evolution operation of classical DE, scale factor $F$ controls the size of the search area around base individual and $CR$ is the probability of inheriting components from the mutant individual when trial individual is constructed. However, for the mixed evolution operation DE/origin-to-guiding/differ, $F$ controls the length of vector from origin individual to guiding individual and the size of difference perturbation, and $CR$ is the probability of inheriting components from the origin individual. In particular, the fitness value of origin individual is only used to set $F$ and $CR$ in [36]. Obviously, the fitness value of guiding individual might be also helpful to set the control parameters. Then, we shall use the fitness values of origin and guiding individuals simultaneously to set parameters $F_1$ and $CR$.

Differing from [36], to distinguish the importance of the vector from origin individual to guiding individual with that of the difference perturbation in DE/origin-to-guiding/1, scale factors $F_1$ and $F_2$ are used to control the length of vector from origin individual to guiding individual, and size of difference perturbation in CM strategy, respectively. According to [7], we set $F_2 = 0.5$ to maintain the exploration ability of CM strategy.

To set $F_1$ suitably, we consider the following two cases.

*Case 1.* The guiding individual is better than origin individual. Then the direction from origin individual to guiding individual can be considered as a promising direction, and $F_1$ should be big. Thus we set

$$F_1 = \frac{1}{2}\left(1 + \frac{f_{\max} - f_g}{f_{\max} - f_{\min}}\right) \quad (3)$$

where $f_{\max}$ and $f_{\min}$ represent the largest and smallest fitness values respectively, and $f_g$ represents the fitness value of the guiding individual. Clearly, the better the guiding individual is, the larger $F_1$ will be obtained. Then, it can enhance the exploitation around good guiding individual.

*Case 2.* The guiding individual is not better than origin individual. Then the negative direction from origin individual to guiding individual is considered as a promising direction. Thus, let

$$F_1 = -randn(0.5, 0.2) \quad (4)$$

where $randn(mean, std)$ represents a normal distribution. Noticing that the diversity of population will reduce as $F_1$ tends to 0, while the convergence cannot improve as $F_1$ tends to $-1$, we further let

$$F_1 = \begin{cases} -0.05, & \text{if } \ F_1 > -0.05 \\ -0.95, & \text{if } \ F_1 < -0.95. \end{cases} \quad (5)$$

From DE/origin-to-guiding/differ and the rule for $F_1$, the better guiding individual is, the larger probability that mutant vector preserves good information is. Then, a big $CR$ is helpful to generate a good trial individual when guiding individual is good. Thus, let

$$CR = 1 - \frac{R_g}{NP} \quad (6)$$

where $R_g$ is the rank index of guiding individual with ascending order corresponding to fitness value of each individual. Moreover, to void too more or less components inheriting from mutant individual, i.e., to avoid that CR tends to 1 or 0, we let CR lie in interval [0.05, 0.95] by

$$CR = \begin{cases} 0.95, & \text{if } CR > 0.95 \\ 0.05, & \text{if } CR < 0.05. \end{cases} \quad (7)$$

In contrast to [19–21,24,25,27] and [36], the proposed parameter (GIP) setting could make the search lie in a more suitable area since it uses the fitness values of original and guiding individuals simultaneously.

It should be noted that parameters $F_1$ and $F_2$ are used to control the sizes of different difference vectors in single base vector mutation strategies DE/best/2, DE/rand/2 and so on, yet the same values are taken for them in most applications. Here, different values for $F_1$ and $F_2$ are set to distinguish the effects of two difference vectors in (1).

### 2.3. Diversity-based selection strategy

It is well known that selection criterion is key to selection strategy. But the fitness value of individual is only considered to set the selection criterion in popular strategies, such as greedy selection and tournament selection strategies. Then a lot of exploration information is wasted before a better solution is found such that the search becomes inefficient. Obviously, the position of individual may be helpful to improve the exploration of population during selection process. Thus, by using fitness value and position of individual, a weighted fitness value can be defined as follows:

$$f_w(x_{i,G}) = \alpha \frac{f(x_{i,G}) - f_{\min}}{f_{\max} - f_{\min}} + (1-\alpha) \frac{Dis_{\max} - Dis(x_{i,G}, x_{best,G})}{Dis_{\max} + Dis(x_{i,G}, x_{best,G})} \quad (8)$$

where $\alpha \in [0, 1]$ is weighted factor, $f(x_{i,G})$ is the fitness value at $x_{i,G}$, $f_{\max}$ and $f_{\min}$ are maximum and minimum fitness values in current and offspring population respectively, $Dis_{\max}$ is the maximum Euclidean distance between the best individual $x_{best,G}$ and other individuals in current and offspring population, and $Dis(x_{i,G}, x_{best,G})$ is the Euclidean distance between $x_{i,G}$ and $x_{best,G}$. Obviously, the weighted fitness value is nonnegative and lying in [0, 1].

From (8), the Euclidean distance to the best individual will play a main role for small $\alpha$. According to the survival of the fittest and to guarantee the dominant role of fitness value, let

$$\alpha = randn(0.9, 0.05)$$

and

$$\alpha = \begin{cases} 1, & \text{if } \alpha > 1 \\ 0.8, & \text{if } \alpha < 0.8. \end{cases}$$

When the weighted fitness value defined in (8) is used only as selection criterion and $\alpha \neq 1$, we see that if $f(x_{i,G}) \neq f(u_{i,G})$ and $Dis(x_{i,G}, x_{best,G}) \neq Dis(u_{i,G}, x_{best,G})$, then $\vec{x}_{i,G}$ or $\vec{u}_{best,G}$ could enter the next generation even when $f(\vec{u}_{i,G}) < f(\vec{x}_{i,G})$ or $f(\vec{x}_{best,G}) < f(\vec{u}_{best,G})$. Thus, we propose the following diversity-based selection strategy by combining greedy selection with the weighted fitness value:

$$\vec{x}_{i,G+1} = \begin{cases} \vec{u}_{i,G}, & \text{if } f(\vec{u}_{i,G}) < f(\vec{x}_{i,G}) \\ \vec{u}_{i,G}, & \text{if } f_w(\vec{u}_{i,G}) \le f_w(\vec{x}_{i,G}) \text{ and } \vec{x}_{i,G} \neq \vec{x}_{best,G} \\ \vec{x}_{i,G}, & \text{otherwise} \end{cases} \quad (9)$$

where $f_w(\vec{u}_{i,G})$ and $f_w(\vec{x}_{i,G})$ are defined in (8). Obviously, if $u_{i,G}$ has better fitness value or weighted fitness value, then it survives in next generation excepting for $\vec{u}_{best,G}$; otherwise, $x_{i,G}$ survives in next generation. In contrast to greedy selection strategy [2] and tournament selection [37], the proposed strategy (9) could improve population exploration since it uses the fitness values and positions of individuals simultaneously.

### 2.4. Algorithm framework

In summary, the framework of IDEI can be described in Algorithm 1 by means of CM strategy, GIP setting and DS strategy in Section 2.1–2.3, and the binomial crossover operation [39]:

$$u_{i,G}^j = \begin{cases} v_{i,G}^j, & \text{if } rand \le Cr \text{ or } j = randn(i) \\ x_{i,G}^j, & \text{otherwise.} \end{cases} \quad (10)$$

where $i = 1, 2, \ldots, NP$, $j = 1, 2, \ldots, D$, $v_{i,G}^j$, $u_{i,G}^j$ and $x_{i,G}^j$ are the $j$th components of $\vec{v}_{i,G}$, $\vec{u}_{i,G}$ and $\vec{x}_{i,G}$ respectively, and $randn(i) \in \{1, 2, \ldots, D\}$ is a random chosen index to ensure that $\vec{u}_{i,G}$ has at least one element from $\vec{v}_{i,G}$.

**Algorithm 1** *((IDEI)).*

**Step 1. Initialization**
Set up the maximum generation number $G_{\max}$, generation index $G = 0$, success ratio $SR_0 = 1$, the scale factor of difference perturbation $F_2 = 0.5$, randomly create the initial population of $NP$ individuals within the range $[L, U]$.
**Step 2. Evolution Iteration**
WHILE the termination criterion is not satisfied
DO
Compute the success ratio $SR_G$ by (2)
**Step 2.1. Mutation**
Compute $Pop_s$ and $Pop_g$ based on Section 2.1, compute $F_1$ by (3) or (4) and (5), and generate mutant individual by (1).
**Step 2.2. Crossover Operation**
Compute $CR$ for each target individual by (6) and (7), and generate trial individual by (10).
**Step 2.3. Selection Operation**
DS strategy (9) is used to update population.
$G = G + 1$
END WHILE

Obviously, the main differences between IDEI and the classic DE are the CM and DS strategies. Then the complexity of IDEI is about the sum of that for the classic DE, CM and DS strategies. According to [20], the complexity for the classic DE algorithm is $O(NP \cdot D \cdot G_{max})$.

For the CM strategy, the guiding sets $Pop_s$ and $Pop_g$ are required to determine at each generation by sorting the population based on their fitness values. Then, its complexity is $O(NP \cdot \log_2 NP \cdot G_{max})$ by simple calculation.

For the DS strategy, it must compute the distances between each individual and the best individual at each generation. Then its complexity is $O(NP \cdot D \cdot G_{max})$. Therefore the total complexity of IDEI is $O(NP \cdot (\log_2 NP + 2D) \cdot G_{max})$, which is about two times of that for the classical DE.

## 3. Experiments and discussion

In this section, the performance of IDEI is evaluated on 55 well-known benchmark functions $f_1 - f_{25}$ from CEC 2005 [40,41] and $f_{26} - f_{55}$ from CEC 2014 [48]. Meanwhile, the suitability of parameters in IDEI and the validity of CM and DS strategies are illustrated by four typical functions: unimodal function $f_4$, basic multimodal functions $f_6$ and $f_{11}$ and expanded multimodal function $f_{13}$. Finally, IDEI is compared with four existing state-of-the-art DE variants and four non-DE algorithms, and its efficiency is discussed.

In all experiments except for those in Section 3.2, the maximum number of function evaluations $FES_{max}$ is set to 300,000, and all

**Table 1**
Experimental results of IDEI with different values for $\xi_1$ and $\xi_3$.

| Functions | | $f_4$ | $f_6$ | $f_{11}$ | $f_{13}$ |
|---|---|---|---|---|---|
| $\xi_1$ | $\xi_3$ | Mean error ± Std. dev. | Mean error ± Std. dev. | Mean error ± Std. dev. | Mean error ± Std. dev. |
| 0.025 | 0.025 | 2.18E−05 ± 6.87E−05 | 3.05E−04 ± 1.37E−03 | 4.92E+00 ± 2.15E+00 | 7.99E+00 ± 5.83E+00 |
| | 0.05 | 2.49E−06 ± 6.70E−06 | 1.49E−04 ± 5.30E−04 | 4.13E+00 ± 2.04E+00 | 4.86E+00 ± 4.49E+00 |
| | 0.075 | 1.68E−05 ± 6.68E−05 | 2.13E−04 ± 6.46E−04 | 5.11E+00 ± 2.78E+00 | 3.32E+00 ± 3.55E+00 |
| 0.05 | 0.025 | 6.95E−06 ± 1.95E−05 | **1.37E−04** ± **2.53E−04** | 4.96E+00 ± 3.40E+00 | 3.63E+00 ± 4.79E+00 |
| | 0.05 | **2.46E−06** ± **6.05E−06** | 2.17E−03 ± 9.38E−03 | **3.94E+00** ± **1.93E+00** | 5.25E+00 ± 5.50E+00 |
| | 0.075 | 2.57E−06 ± 4.99E−06 | 1.80E−04 ± 2.92E−04 | 4.99E+00 ± 2.39E+00 | 3.58E+00 ± 4.12E+00 |
| 0.1 | 0.025 | 2.13E−05 ± 7.74E−05 | 7.66E−01 ± 2.04E+00 | 4.57E+00 ± 2.40E+00 | 1.83E+00 ± 2.86E+00 |
| | 0.05 | 6.39E−06 ± 1.31E−05 | 1.55E−01 ± 7.52E−01 | 3.99E+00 ± 2.45E+00 | 3.55E+00 ± 3.90E+00 |
| | 0.075 | 1.90E−04 ± 7.80E−04 | 1.55E−01 ± 6.16E−01 | 5.43E+00 ± 2.52E+00 | 1.89E+00 ± 2.62E+00 |
| 0.2 | 0.025 | 1.96E−05 ± 6.01E−05 | 1.49E+01 ± 4.60E+00 | 6.00E+00 ± 2.11E+00 | 1.07E+00 ± 2.02E−01 |
| | 0.05 | 3.61E−06 ± 5.16E−06 | 1.44E+01 ± 4.86E+00 | 4.59E+00 ± 1.69E+00 | **1.06E+00** ± **1.74E−01** |
| | 0.075 | 3.22E−05 ± 8.19E−05 | 1.38E+01 ± 5.37E+00 | 5.71E+00 ± 2.34E+00 | 2.56E+00 ± 3.70E+00 |
| 0.3 | 0.025 | 6.52E−05 ± 1.46E−04 | 1.85E+01 ± 2.55E+00 | 8.43E+00 ± 3.04E+00 | 1.11E+00 ± 2.12E−01 |
| | 0.05 | 2.93E−05 ± 1.01E−04 | 2.55E+01 ± 1.66E+01 | 7.59E+00 ± 2.99E+00 | 1.51E+00 ± 2.46E+00 |
| | 0.075 | 1.92E−04 ± 8.93E−04 | 2.27E+01 ± 1.40E+01 | 7.01E+00 ± 2.64E+00 | 1.76E+00 ± 2.30E+00 |

problems with $D = 30$ are run 25 times independently. In IDEI, the size of population $NP$ is set to 100.

### 3.1. Reasonability of prescribed values for $\xi_1$ and $\xi_3$

To demonstrate the reasonability of prescribed values for $\xi_1$ and $\xi_3$, IDEI is run on $f_4$, $f_6$, $f_{11}$ and $f_{13}$ with various values of $\xi_1$ and $\xi_3$. Table 1 reports the experimental results, where Mean Error and Std Dev represent the average and standard deviation of error function value $f(\vec{x}) - f(\vec{x}^*)$ recorded for measuring their performances respectively, and the best results are marked by bold on each function.

From Table 1, we see that unimodal function $f_4$ and basic multimodal function $f_{11}$ attain the best results when $\xi_1 = 0.05$ and $\xi_3 = 0.05$, $f_6$ attains the best results when $\xi_1 = 0.05$ and $\xi_3 = 0.025$, and expanded multimodal function $f_{13}$ gets the best results when $\xi_1 = 0.2$ and $\xi_3 = 0.05$. Then, prescribed value $\xi_3 = 0.05$ is reasonable for most functions, prescribed value $\xi_1 = 0.05$ and 0.2 are reasonable for unimodal and basic multimodal functions and expanded multimodal functions. In fact, in contrast to unimodal and basic multimodal functions, the expanded multimodal function needs to exploit more potential individuals to avoid stagnation [42]. Thus it is more reasonable that $\xi_1$ for expanded multimodal functions is bigger than for simple functions.

### 3.2. Validity of the CM and DS strategies

Same as Section 3.1, the performance of CM and DS strategies shall be illustrated on four typical functions $f_4, f_6, f_{11}$ and $f_{13}$. In particular, to show clearly the performances of CM and DS strategies, we set $D = 10$, $NP = 20$ and $FES_{max} = 30{,}000$ in this subsection, and the average of error function values are recorded to evaluate their performances.

#### 3.2.1. Validity of the CM strategy

To evaluate the performance of CM strategy, we employ greedy selection strategy [2] in IDEI and make a comparison of CM strategy with DE/best/1 (DE/B/1), DE/current/1 (DE/C/1) and DE/current-to-best/1 (DE/C-to-B/1).

In these experiments, $F = CR = 0.5$ in three strategies above, and are same as in Section 2.2 for CM strategy. Fig. 1(a)–(d) depict the evolution curves of different strategies on four functions above, respectively.

From Fig. 1, we see that CM strategy has more promising performance than others. In particular, CM strategy has better exploitation ability from Fig. 1(a), and can enhance the exploration ability to search the decision space from Fig. 1(b)–(d). Then, CM strategy can effectively balance the exploration and exploitation during evolution process and is very helpful to improve the performance of DE.

#### 3.2.2. Validity of the DS strategy

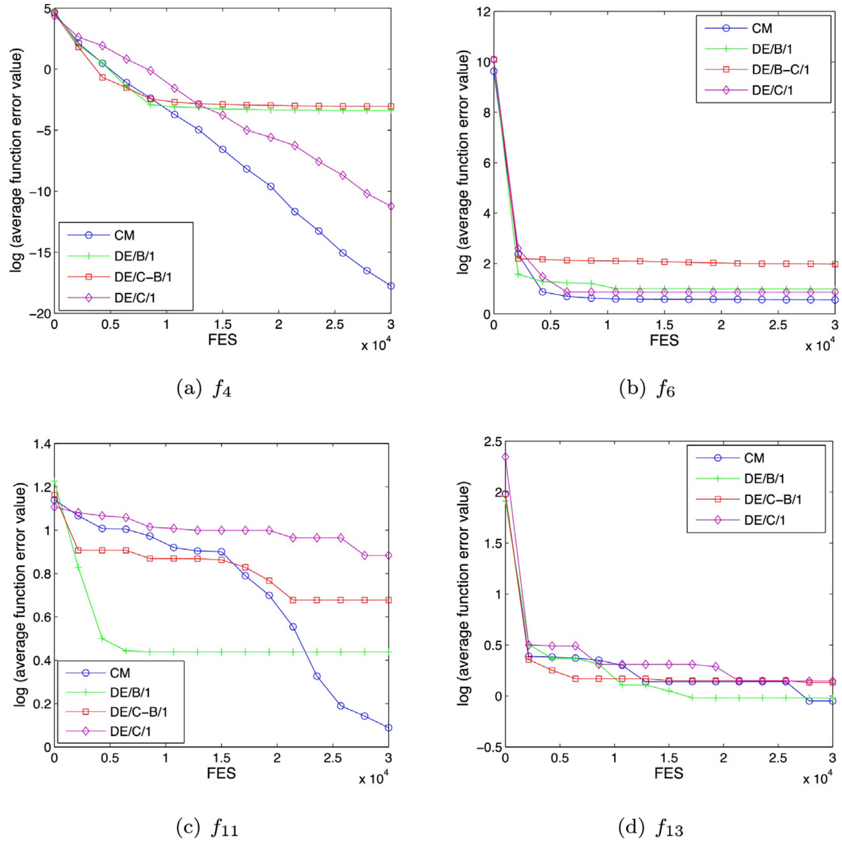
To evaluate the performance of DS strategy, we make a comparison of DS strategy with greedy selection strategy [2] in DE/best/1 (DE/B/1), DE/current/1 (DE/C/1) and DE/rand/1 (DE/R/1), where $F = CR = 0.5$. Figs. 2–5 depict their evolution curves on four functions above, respectively.

From Figs. 2 and 4, we see that DS strategy outperforms greedy selection strategy in all cases except that they have similar performance on $f_4$ with DE/current/1. For $f_6$, Fig. 3 shows that DS strategy has better performance with DE/best/1, and has similar performance as greedy selection strategy with DE/current/1 and DE/rand/1. Moreover, Fig. 5 shows that DS strategy outperforms on $f_{13}$ with DE/current/1 and DE/best/1, and has similar performance as greedy selection strategy with DE/rand/1. Then DS strategy has better performance in most cases. In fact, unlike greedy selection strategy, DS strategy uses not only the information of better individual but also that of worse individual in some cases. Thus, DS strategy is a promising selection strategy. So, the proposed CM strategy and DS strategy strategies are competitive.

### 3.3. Comparisons and discussions

To evaluate the benefits of IDEI, we shall make a comparison of IDEI with four existing DE variants: jDE [33], SaDE [25], EPSDE [27] and CoDE [24] and four non-DE algorithms: CLPSO [13], CMA-ES [43], GL-25 [44] and HSOGA [45] on 55 benchmark functions $f_1 - f_{55}$. It should be mentioned that hybrid composition functions $f_{15} - f_{25}$ and $f_{48} - f_{55}$ are much harder than $f_1 - f_{14}$ and $f_{26} - f_{47}$ since each of them is composed of several sub-functions, and no method can reduce the average error function values $f(\vec{x}) - f(\vec{x}^*)$ of $f_{15} - f_{25}$ under 10, where $\vec{x}$ is the best solution found by the algorithm in a run and $\vec{x}^*$ is the global optimum of test function.

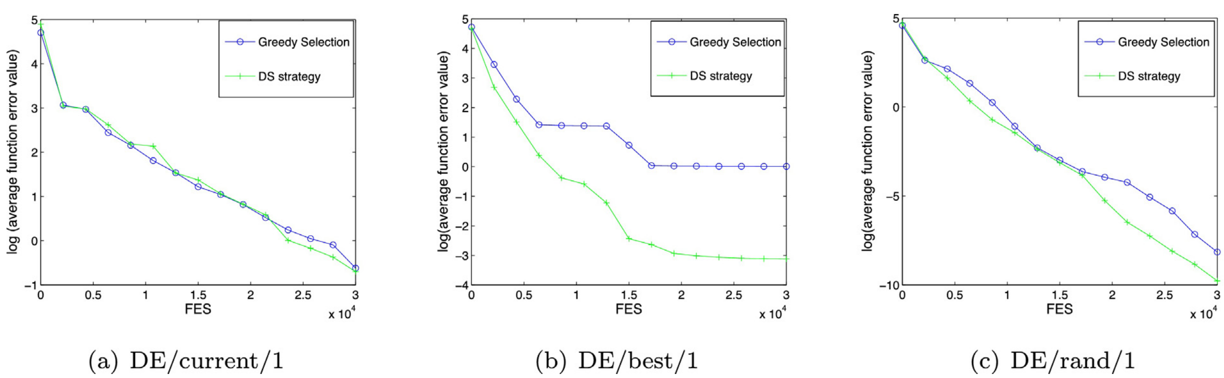
In all experiments of this subsection, the average error (Mean Error) and standard deviation (Std Dev) of $f(\vec{x}) - f(\vec{x}^*)$ are recorded for measuring their performances, and Wilcoxon's rank sum test at a 0.05 significance level and average rank are conducted on the experimental results to have statistically sound conclusions. According to Section 3.1, $\xi_3 = 0.05$ for all functions, $\xi_1$ is set to 0.05 for unimodal and basic multimodal functions $f_1 - f_{12}$ and $f_{26} - f_{41}$, and 0.2 for expanded multimodal and hybrid composition functions $f_{13} - f_{25}$ and $f_{42} - f_{55}$.




(a) $f_4$ (b) $f_6$

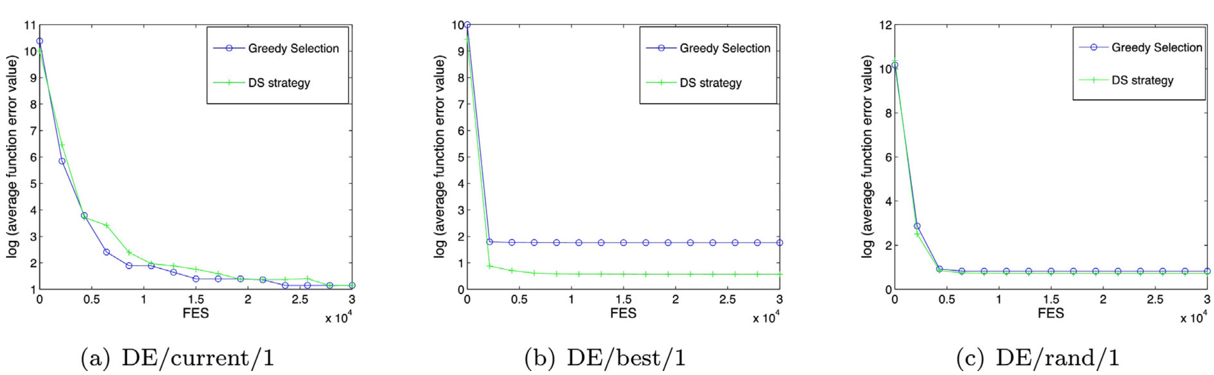
(c) $f_{11}$ (d) $f_{13}$

**Fig. 1.** The evolution curves of CM strategy and three mutation operators. (a) $f_4$, (b) $f_6$, (c) $f_{11}$ and (d) $f_{13}$.



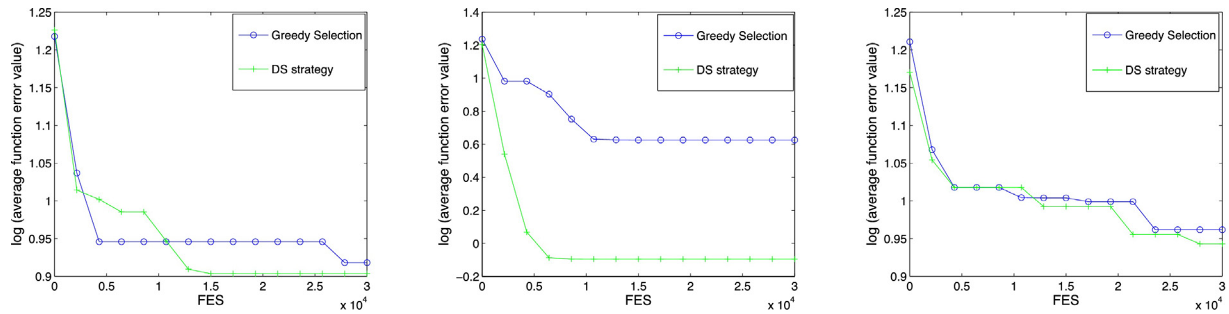

(a) DE/current/1 (b) DE/best/1 (c) DE/rand/1

**Fig. 2.** The evolution curves of DS strategy and greedy selection strategy on $f_4$. (a) DE/C/1, (b) DE/B/1 and (c) DE/R/1.



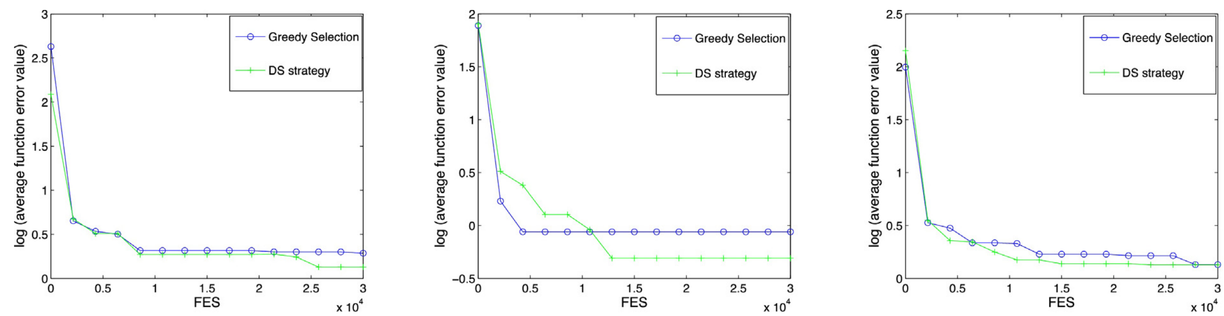

(a) DE/current/1 (b) DE/best/1 (c) DE/rand/1

**Fig. 3.** The evolution curves of DS strategy and greedy selection strategy on $f_6$. (a) DE/C/1, (b) DE/B/1 and (c) DE/R/1.




(a) DE/current/1 (b) DE/best/1 (c) DE/rand/1

**Fig. 4.** The evolution curves of DS strategy and greedy selection strategy on $f_{11}$. (a) DE/C/1, (b) DE/B/1 and (c) DE/R/1.




(a) DE/current/1 (b) DE/best/1 (c) DE/rand/1

**Fig. 5.** The evolution curves of DS strategy and greedy selection strategy on $f_{13}$. (a) DE/C/1, (b) DE/B/1 and (c) DE/R/1.

*3.3.1. Comparison with four DE algorithms*

First, we make a comparison of IDEI with four existing DE variants jDE [33], SaDE [25], EPSDE [27] and CoDE [24], whose control parameters are self-adaptive.

In the experiments, we take the same control parameter settings for four existing DE variants as their original papers. Tables 2 and 3 list the experimental results on 55 test functions from CEC 2005 and 2014 respectively, and the last three rows summarize them according to Wilcoxon's rank sum test, where "Mean Error" and "Std Dev" are same as Table 1. From Tables 2 and 3, we see that for

(1) unimodal functions $f_1-f_5$ and $f_{26}-f_{28}$, IDEI obtains the best mean fitness value and standard deviation except that EPSDE and CoDE achieve the better results on $f_2$. According to the Wilcoxon's rank sum test, IDEI outperforms jDE, SaDE, EPSDE and CoDE on 5, 6, 5 and 4 test functions respectively, and slightly worse than them on 0, 0, 1 and 1 benchmark functions, respectively.
(2) basic multimodal functions $f_6-f_{12}$ and $f_{29}-f_{41}$, IDEI obtains the best results except that CoDE achieves the best results on $f_8, f_{30}$ and $f_{37}$, and jDE on $f_{33}$ and $f_{35}$. From the Wilcoxon's rank sum test, IDEI outperforms jDE, SaDE, EPSDE and CoDE on 11, 14, 15 and 10 benchmark functions respectively, and slightly worse than them on 6, 5, 4 and 8 benchmark functions, respectively.
(3) expanded multimodal functions $f_{13}-f_{14}$ and $f_{42}-f_{47}$, IDEI obtains the best mean fitness value and standard deviation except that CoDE has the best results on $f_{43}$. In particular, IDEI is much better than jDE, SaDE, EPSDE and CoDE on 8, 8, 8 and 7 benchmark functions, respectively.
(4) hybrid composition functions $f_{15}-f_{25}$ and $f_{48}-f_{55}$, IDEI performs significantly better than jDE, SaDE, EPSDE and CoDE on 7, 10, 8 and 5 test functions respectively, and slightly worse than them on 1, 0, 10 and 1 test functions, respectively. Then IDEI is significantly better than jDE, SaDE and CoDE, and worse than EPSDE on these composition functions. Differing from others, EPSDE survives the mutation strategy that produces better offspring and replaces it with another one randomly chosen from the strategy pool when this strategy fail to produce better offspring.

In summary, according to the results of Wilcoxon's rank sum test reported in Tables 2 and 3, IDEI is better than jDE, SaDE, EPSDE and CoDE on 31, 40, 36 and 27 test functions respectively, similar as jDE, SaDE, EPSDE and CoDE on 17, 10, 4 and 17 test functions respectively, and slightly worse than jDE, SaDE, EPSDE and CoDE on 7, 5, 15 and 11 test functions, respectively. Moreover, jDE, SaDE, EPSDE, CoDE and IDEI obtain 2.72, 2.88, 3.48, 1.88 and 1.40 in the term of the average rank on the 25 test functions from CEC 2005, and 2.77, 3.67, 3.43, 1.97 and 1.93 on the 30 test functions from CEC 2014. Therefore, IDEI is a more promising one among them.

*3.3.2. Comparison with four non-DE algorithms*

Meanwhile, a comparison of IDEI with four non-DE approaches: CLPSO [13], CMA-ES [43], GL-25 [44] and HSOGA [45] is also made. In CLPSO, a particle uses the personal historical best information of all particles to update its velocity [13]. CMA-ES [43] is a very efficient and famous evolution strategy (ES). GL-25 [44] is a hybrid real-coded genetic algorithm which combines the global and local search. HSOGA [45] is a hybrid self-adaptive orthogonal genetic algorithm by designing a self-adaptive orthogonal crossover operator.

In the experiments, we take the same control parameters for CLPSO, CMA-ES, GL-25 and HSOGA as their original papers. Tables 4 and 5 list the experimental results on 55 test functions from CEC 2005 and 2014 respectively, and the last three rows summarize them according to Wilcoxon's rank sum test. From Tables 4 and 5, we see that for

**Table 2**
Experimental results of IDEI and four state-of-the-art DEs on *CEC*2005 functions.

| Functions | jDE Mean error ± Std. dev. /Rank | SaDE Mean error ± Std. dev. /Rank | EPSDE Mean error ± Std. dev. /Rank | CoDE Mean error ± Std. dev /Rank | IDEI 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 |
| $f_2$ | 1.11E−06 ± 1.96E−06+ /4 | 8.26E−06 ± 1.65E−05+ /5 | 4.23E−26 ± 4.07E−26− /1 | 1.69E−15 ± 3.95E−15− /2 | 6.22E−12 ± 1.61E−11 /3 |
| $f_3$ | 1.98E+05 ± 1.10E+05+ /3 | 4.27E+05 ± 2.08E+05+ /4 | 8.74E+05 ± 3.28E+06+ /5 | 1.05E+05 ± 6.25E+04+ /2 | 6.04E+04 ± 3.09E+04 /1 |
| $f_4$ | 4.40E−02 ± 1.26E−01+ /3 | 1.77E+02 ± 2.67E+02+ /4 | 3.49E+02 ± 2.23E+03+ /5 | 5.81E−03 ± 1.38E−02+ /2 | 2.46E−06 ± 6.05E−06 /1 |
| $f_5$ | 5.11E+02 ± 4.40E+02+ /3 | 3.25E+03 ± 5.90E+02+ /5 | 1.40E+03 ± 7.12E+02+ /4 | 3.31E+02 ± 3.44E+02+ /2 | 5.41E+00 ± 1.55E+01 /1 |
| $f_6$ | 2.35E+01 ± 2.50E+01+ /4 | 5.31E+01 ± 3.25E+01+ /5 | 6.38E−01 ± 1.49E+00+ /3 | 1.60E−01 ± 7.85E−01+ /2 | 2.17E−03 ± 9.38E−03 /1 |
| $f_7$ | 1.18E−02 ± 7.78E−03+ /3 | 1.57E−02 ± 1.38E−02+ /4 | 1.77E−02 ± 1.34E−02+ /5 | 7.46E−03 ± 8.55E−03+ /2 | 2.76E−03 ± 4.73E−03 /1 |
| $f_8$ | 2.09E+01 ± 4.86E−02≈ /2 | 2.09E+01 ± 4.95E−02≈ /2 | 2.09E+01 ± 5.81E−02≈ /2 | 2.01E+01 ± 1.41E−01− /1 | 2.09E+01 ± 4.34E−02 /2 |
| $f_9$ | 0.00E+00 ± 0.00E+00≈ /1 | 2.39E−01 ± 4.33E−01+ /5 | 3.98E−02 ± 1.99E−01+ /4 | 0.00E+00 ± 0.00E+00≈ /1 | 0.00E+00 ± 0.00E+00 /1 |
| $f_{10}$ | 5.54E+01 ± 8.46E+00+ /5 | 4.72E+01 ± 1.01E+01+ /3 | 5.36E+01 ± 3.03E+01+ /4 | 4.15E+01 ± 1.16E+01+ /2 | 2.37E+01 ± 6.49E+00 /1 |
| $f_{11}$ | 2.79E+01 ± 1.61E+00+ /4 | 1.65E+01 ± 2.42E+00+ /3 | 3.56E+01 ± 3.88E+00+ /5 | 1.18E+01 ± 3.40E+00+ /2 | 3.94E+00 ± 1.93E+00 /1 |
| $f_{12}$ | 8.63E+03 ± 8.31E+03+ /4 | 3.02E+03 ± 2.33E+03+ /2 | 3.58E+04 ± 7.05E+03+ /5 | 3.05E+03 ± 3.80E+03+ /3 | 1.60E+03 ± 1.91E+03 /1 |
| $f_{13}$ | 1.66E+00 ± 1.35E−01+ /3 | 3.94E+00 ± 2.81E−01+ /5 | 1.94E+00 ± 1.46E−01+ /4 | 1.57E+00 ± 3.27E+01+ /2 | 1.06E+00 ± 1.74E−01 /1 |
| $f_{14}$ | 1.30E+01 ± 2.00E−01+ /4 | 1.26E+01 ± 2.83E−01+ /3 | 1.35E+01 ± 2.09E−01+ /5 | 1.23E+01 ± 4.81E−01+ /2 | 1.18E+01 ± 4.38E−01 /1 |
| $f_{15}$ | 3.77E+02 ± 8.02E+01+ /4 | 3.76E+02 ± 7.83E+01≈ /2 | 2.12E+02 ± 1.98E+01− /1 | 3.88E+02 ± 6.85E+01+ /5 | 3.76E+02 ± 7.79E+01 /2 |
| $f_{16}$ | 7.94E+01 ± 2.96E+01+ /3 | 8.57E+01 ± 6.94E+01+ /4 | 1.22E+02 ± 9.19E+01+ /5 | 7.37E+01 ± 5.13E+01+ /2 | 5.37E+01 ± 2.40E+01 /1 |
| $f_{17}$ | 1.37E+02 ± 3.80E+01+ /4 | 7.83E+01 ± 3.76E+01+ /3 | 1.69E+02 ± 1.02E+02+ /5 | 6.67E+01 ± 2.12E+01+ /2 | 6.58E+01 ± 3.70E+01 /1 |
| $f_{18}$ | 9.04E+02 ± 1.08E+01≈ /2 | 8.68E+02 ± 6.23E+01≈ /2 | 8.20E+02 ± 3.35E+00− /1 | 9.04E+02 ± 1.04E+00≈ /2 | 9.04E+02 ± 1.25E+00 /2 |
| $f_{19}$ | 9.04E+02 ± 1.11E+00≈ /2 | 8.74E+02 ± 6.22E+01≈ /2 | 8.21E+02 ± 3.35E+00− /1 | 9.04E+02 ± 9.42E−01≈ /2 | 9.04E+02 ± 1.32E+00 /2 |
| $f_{20}$ | 9.04E+02 ± 1.10E+00≈ /2 | 8.78E+02 ± 6.03E+01≈ /2 | 8.22E+02 ± 4.17E+00− /1 | 9.04E+02 ± 9.01E−01≈ /2 | 9.04E+02 ± 1.07E+00 /2 |
| $f_{21}$ | 5.00E+02 ± 4.80E−13≈ /1 | 5.52E+02 ± 1.82E+02+ /4 | 8.33E+02 ± 1.00E+02+ /5 | 5.00E+02 ± 4.88E−13≈ /1 | 5.00E+02 ± 9.91E−14 /1 |
| $f_{22}$ | 8.75E+02 ± 1.91E+01− /3 | 9.36E+02 ± 1.83E+01+ /5 | 5.07E+02 ± 7.26E+00− /1 | 8.63E+02 ± 2.43E+01− /2 | 8.85E+02 ± 1.44E+01 /4 |
| $f_{23}$ | 5.34E+02 ± 2.77E−04≈ /1 | 5.34E+02 ± 3.57E−03≈ /1 | 8.58E+02 ± 6.82E+01+ /5 | 5.34E+02 ± 4.12E−04≈ /1 | 5.34E+02 ± 4.20E−04 /1 |
| $f_{24}$ | 2.00E+02 ± 2.85E−14≈ /1 | 2.00E+02 ± 6.20E−13≈ /1 | 2.13E+02 ± 1.52E+00+ /5 | 2.00E+02 ± 2.85E−14≈ /1 | 2.00E+02 ± 2.90E−14 /1 |
| $f_{25}$ | 2.11E+02 ± 7.32E−01≈ /1 | 2.14E+02 ± 2.00E+00+ /5 | 2.13E+02 ± 2.55E+00+ /4 | 2.11E+02 ± 9.02E−01≈ /1 | 2.11E+02 ± 8.78E−01 /1 |
| Average rank | 2.72 | 2.88 | 3.48 | 1.88 | 1.40 |
| + | 14 | 17 | 17 | 13 | |
| − | 1 | 0 | 6 | 3 | |
| ≈ | 10 | 8 | 2 | 9 | |

Wilcoxon's rank sum test at a 0.05 significance level is performed between IDEI and each of jDE, SaDE, EPSDE and CoDE.
"+", "−" and "≈" denote that the performance of IDEI is better than, worse than, and similar to that of the corresponding algorithm, respectively.

(1) unimodal functions $f_1$-$f_5$ and $f_{26}$-$f_{28}$, IDEI obtains the best mean fitness value and standard deviation on $f_1$, $f_4$, $f_{27}$ and $f_{28}$. In particular, IDEI has similar performances as CLPSO on $f_1$, and CMA-ES on $f_{27}$ and $f_{28}$. For other four functions, IDEI outperforms CLPSO, GL-25 and HSOGA, and CMA-ES achieves the best results. According to Wilcoxon's rank sum test, IDEI is much better than CLPSO, CMA-ES, GL-25 and HSOGA on 7, 2, 8 and 8 test functions respectively, and slightly worse on 0, 4, 0 and 0 test functions, respectively. Then IDEI outperforms CLPSO, GL-25 and HSOGA, and worse than CMA-ES. In fact, CMA-ES adds the evolution path such that it has better results.

(2) basic multimodal functions $f_6 - f_{12}$ and $f_{29} - f_{41}$, IDEI obtains the best results on 9 functions $f_6$, $f_9 - f_{12}$, $f_{31}$, $f_{32}$, $f_{38}$ and $f_{40}$. In particular, CMA-ES obtains the best results on $f_7, f_8, f_{29}$ and $f_{30}$, HSOGA has the best results on $f_{34}$, and CLPSO has the best performances on $f_{33}, f_{35} - f_{37}, f_{39}$ and $f_{41}$. According to the statistical results of the Wilcoxon's rank sum test, IDEI is much better than CLPSO, CMA-ES, GL-25 and HSOGA on 11, 15, 18 and 16 benchmark functions respectively, and slightly worse on 7, 5, 1 and 3 test functions, respectively.

(3) expanded multimodal functions $f_{13} - f_{14}$ and $f_{42} - f_{47}$, IDEI obtains the best mean fitness value and standard deviation. In particular, IDEI is much better than CLPSO, CMA-ES, GL-25 and HSOGA on 8, 8, 8 and 8 benchmark functions, respectively. Thus IDEI outperforms four competitors.

(4) hybrid composition functions $f_{15} - f_{25}$ and $f_{48} - f_{55}$, IDEI obtains the best results on 11 functions $f_{16} - f_{21}$, $f_{23}$, $f_{24}$, $f_{48}$, $f_{51}$ and $f_{55}$. In particular, GL-25 obtains the best performances on $f_{15}$ and $f_{52}$, CMA-ES and CLPSO have the best results on $f_{22}$ and $f_{25}$ respectively, and HSOGA obtains the best results on $f_{49}$, $f_{50}$, $f_{53}$ and $f_{54}$. From the Wilcoxon's rank sum test, IDEI outperforms CLPSO, CMA-ES, GL-25 and HSOGA on 12, 13, 11 and 10 benchmark functions respectively, and slightly worse on 1, 2, 3 and 5 test functions, respectively.

In summary, the statistical results of the Wilcoxon's rank sum test in Tables 4 and 5 indicate that IDEI performs better than CLPSO, CMA-ES, GL-25 and HSOGA on 38, 38, 45 and 42 test functions respectively, similar as them on 8, 6, 6 and 5 test functions respectively, and worse than them on 9, 11, 4 and 8 test functions, respectively. Furthermore, according to the average rank, CLPSO, CMA-ES, GL-25, HSOGA and IDEI obtain 3.08, 2.76, 3.04, 3.52 and 1.40 on the 25 test functions from CEC 2005, and 3.03, 3.27, 3.07, 3.20 and 1.73 on the 30 test functions from CEC 2014. Then, IDEI significantly outperforms CLPSO, CMA-ES, GL-25 and HSOGA.

Moreover, it should be pointed out that the reason for IDEI performs better than eight algorithms above is that the CM, GIP and DS strategies could avoid effectively premature convergence or stagnation for the complicated functions. But the DS strategy might result in the slow convergence on some unimodal functions. Therefore, IDEI has better performance than others on most complicated functions, and worse results on some unimodal functions.

### 3.4. Algorithm efficiency

In order to show the efficiency of IDEI, a comparison of IDEI with jDE, SaDE, EPSDE and CoDE is conducted on five typical functions, unimodal functions $f_4$ and $f_{26}$, and basic multimodal functions $f_{11}$, $f_{31}$ and $f_{34}$. The experiments are conducted in MATLAB 2010a on a personal computer (Intel(R) Core(TM)2 Quad CPU at 2.83GHz and 4GB memory), and the prescribed function error values (PFEV) for $f_4, f_{26}, f_{11}, f_{31}$ and $f_{34}$ are set to 0.001, 50,000, 10, 1 and 40 respectively according to Tables 2 and 3. Table 6 reports their numerical

**Table 3**
Experimental results of IDEI and four state-of-the-art DEs on *CEC*2014 functions.

| Functions | jDE<br>Mean error ± Std. dev. /Rank | SaDE<br>Mean error ± Std. dev. /Rank | EPSDE<br>Mean error ± Std. dev. /Rank | CoDE<br>Mean error ± Std. dev. /Rank | IDEI<br>Mean error ± Std. dev. /Rank |
|---|---|---|---|---|---|
| $f_{26}$ | 1.82E+05 ± 1.87E+05+ /4 | 3.44E+05 ± 2.31E+05+ /5 | 2.80E+04 ± 1.13E+05+ /3 | 2.56E+04 ± 2.27E+04+ /2 | 9.57E+03 ± 5.80E+03 /1 |
| $f_{27}$ | 0.00E+00 ± 2.32E−14≈ /1 | 0.00E+00 ± 5.16E−14≈ /1 | 0.00E+00 ± 7.67E−14≈ /1 | 0.00E+00 ± 0.00E+00≈ /1 | 0.00E+00 ± 1.83E−14 /1 |
| $f_{28}$ | 0.00E+00 ± 4.92E−14≈ /1 | 5.55E+00 ± 1.43E+01+ /5 | 9.00E−12 ± 2.54E−11+ /4 | 0.00E+00 ± 1.83E−14≈ /1 | 0.00E+00 ± 3.85E−14 /1 |
| $f_{29}$ | 2.99E+01 ± 3.07E+01+ /3 | 3.02E+01 ± 4.15E+01+ /4 | 3.72E+00 ± 2.25E+00+ /5 | 2.11E+00 ± 1.16E+01+ /2 | 4.47E−01 ± 4.31E−01 /1 |
| $f_{30}$ | 2.04E+01 ± 4.44E−02− /2 | 2.05E+01 ± 5.52E−02− /4 | 2.04E+01 ± 6.39E−02− /2 | 2.01E+01 ± 8.60E−02− /1 | 2.09E+01 ± 4.83E−02 /5 |
| $f_{31}$ | 1.47E+01 ± 1.55E+00+ /4 | 5.54E+00 ± 1.84E+00+ /3 | 1.85E+01 ± 2.27E+00+ /5 | 2.77E+00 ± 1.81E+00+ /2 | 5.65E−01 ± 8.70E−01 /1 |
| $f_{32}$ | 0.00E+00 ± 1.01E−13≈ /1 | 6.79E−03 ± 1.10E−02+ /5 | 2.07E−03 ± 4.31E−03+ /4 | 0.00E+00 ± 0.00E+00≈ /1 | 0.00E+00 ± 8.99E−14 /1 |
| $f_{33}$ | 0.00E+00 ± 0.00E+00− /1 | 0.00E+00 ± 0.00E+00− /1 | 3.98E−02 ± 1.99E−01− /4 | 3.32E−02 ± 1.82E−01− /3 | 3.18E−01 ± 4.74E−01 /5 |
| $f_{34}$ | 5.75E+01 ± 6.87E+00+ /5 | 3.87E+01 ± 8.25E+00+ /3 | 4.41E+01 ± 7.79E+00+ /4 | 3.75E+01 ± 9.45E+00+ /2 | 2.15E+01 ± 6.02E+00 /1 |
| $f_{35}$ | 3.80E−02 ± 5.92E−02− /1 | 2.24E−01 ± 3.69E−01− /3 | 1.45E−01 ± 8.62E−02− /2 | 6.13E−01 ± 8.03E−01− /4 | 1.57E+02 ± 1.25E+02 /5 |
| $f_{36}$ | 2.88E+03 ± 2.49E+02− /2 | 3.41E+03 ± 4.42E+02+ /4 | 3.50E+03 ± 4.04E+02+ /5 | 1.83E+03 ± 4.90E+02− /1 | 3.06E+03 ± 2.36E+03 /3 |
| $f_{37}$ | 5.27E−01 ± 7.45E−02− /3 | 8.05E−01 ± 7.37E−02− /4 | 5.19E−01 ± 7.51E−02− /2 | 5.96E−02 ± 3.61E−02− /1 | 2.42E+00 ± 3.97E−01 /5 |
| $f_{38}$ | 3.27E−01 ± 4.58E−02+ /5 | 2.50E−01 ± 3.49E−02+ /3 | 2.50E−01 ± 4.76E−02+ /3 | 2.43E−01 ± 5.33E−02+ /2 | 2.06E−01 ± 4.62E−02 /1 |
| $f_{39}$ | 2.96E−01 ± 3.23E−02+ /5 | 2.38E−01 ± 3.32E−02− /1 | 2.85E−01 ± 7.60E−02+ /4 | 2.40E−01 ± 4.01E−02− /2 | 2.70E−01 ± 3.67E−02 /3 |
| $f_{40}$ | 6.63E+00 ± 8.45E−01+ /5 | 4.48E+00 ± 1.87E+00+ /3 | 5.38E+00 ± 6.73E−01+ /4 | 3.21E+00 ± 7.84E−01≈ /1 | 3.21E+00 ± 1.57E+00 /1 |
| $f_{41}$ | 1.03E+01 ± 2.69E−01− /2 | 1.10E+01 ± 3.01E−01+ /4 | 1.12E+01 ± 5.43E−01+ /5 | 9.49E+00 ± 7.00E−01− /1 | 1.07E+01 ± 5.19E−01 /3 |
| $f_{42}$ | 2.62E+03 ± 2.02E+03+ /3 | 1.41E+04 ± 1.51E+04+ /4 | 3.49E+04 ± 4.64E+04+ /5 | 1.05E+03 ± 1.10E+03+ /2 | 9.13E+02 ± 6.82E+02 /1 |
| $f_{43}$ | 1.83E+01 ± 7.42E+00+ /3 | 3.27E+02 ± 4.72E+02+ /4 | 7.66E+02 ± 2.21E+03+ /5 | 1.52E+01 ± 6.36E+00− /1 | 1.77E+01 ± 8.34E+00 /2 |
| $f_{44}$ | 5.14E+00 ± 5.99E−01+ /3 | 6.67E+00 ± 1.17E+01+ /4 | 1.32E+01 ± 1.10E+00+ /5 | 2.74E+00 ± 4.99E−01+ /2 | 2.47E+00 ± 7.55E−01 /1 |
| $f_{45}$ | 1.38E+01 ± 3.93E+00+ /3 | 8.76E+01 ± 5.30E+01+ /4 | 1.07E+02 ± 3.76E+02+ /5 | 1.32E+01 ± 6.63E+00+ /2 | 7.91E+00 ± 3.22E+00 /1 |
| $f_{46}$ | 4.52E+02 ± 2.13E+02+ /3 | 5.28E+03 ± 7.94E+03+ /4 | 7.41E+03 ± 1.09E+04+ /5 | 2.09E+02 ± 1.35E+02+ /2 | 1.79E+02 ± 1.15E+02 /1 |
| $f_{47}$ | 1.86E+02 ± 6.88E+01+ /4 | 1.72E+02 ± 5.98E+01+ /3 | 2.42E+02 ± 1.07E+02+ /5 | 1.61E+02 ± 1.10E+02+ /2 | 1.23E+02 ± 8.83E+01 /1 |
| $f_{48}$ | 3.15E+02 ± 9.28E−13≈ /2 | 3.15E+02 ± 1.25E−12≈ /2 | 3.14E+02 ± 1.21E−12− /1 | 3.15E+02 ± 1.36E−12≈ /2 | 3.15E+02 ± 9.28E−13 /2 |
| $f_{49}$ | 2.24E+02 ± 1.21E+00≈ /1 | 2.26E+02 ± 3.14E+00+ /4 | 2.30E+02 ± 7.49E+00+ /5 | 2.24E+02 ± 1.22E+00≈ /1 | 2.24E+02 ± 1.20E+00 /1 |
| $f_{50}$ | 2.04E+02 ± 6.74E−01+ /4 | 2.09E+02 ± 1.76E+00+ /5 | 2.00E+02 ± 3.19E−02− /1 | 2.03E+02 ± 5.27E−01≈ /2 | 2.03E+02 ± 3.80E−01 /2 |
| $f_{51}$ | 1.00E+02 ± 4.54E−02≈ /1 | 1.04E+02 ± 2.00E+01+ /5 | 1.00E+02 ± 4.13E−02≈ /1 | 1.00E+02 ± 5.08E−02≈ /1 | 1.00E+02 ± 3.28E−02 /1 |
| $f_{52}$ | 3.81E+02 ± 4.12E+01+ /2 | 4.17E+02 ± 3.46E+01+ /3 | 8.41E+02 ± 8.24E+01+ /5 | 4.89E+02 ± 3.13E+01+ /4 | 3.66E+02 ± 4.78E+01 /1 |
| $f_{53}$ | 7.99E+02 ± 2.48E+01≈ /2 | 8.92E+02 ± 3.48E+01+ /5 | 3.96E+02 ± 1.58E+01− /1 | 8.35E+02 ± 2.97E+01+ /4 | 7.97E+02 ± 4.11E+01 /2 |
| $f_{54}$ | 8.92E+02 ± 1.32E+02+ /3 | 1.05E+03 ± 1.63E+02+ /5 | 2.14E+02 ± 1.03E+00− /1 | 9.44E+02 ± 1.29E+02+ /4 | 7.15E+02 ± 1.06E+02 /2 |
| $f_{55}$ | 1.56E+03 ± 6.83E+02+ /4 | 1.76E+03 ± 4.17E+02+ /5 | 6.04E+02 ± 1.03E+02− /1 | 1.04E+03 ± 5.81E+02+ /3 | 8.96E+02 ± 3.98E+02 /2 |
| Average Rank | 2.77 | 3.67 | 3.43 | 1.97 | 1.93 |
| + | 17 | 23 | 19 | 14 | |
| − | 6 | 5 | 9 | 8 | |
| ≈ | 7 | 2 | 2 | 8 | |

Wilcoxon's rank sum test at a 0.05 significance level is performed between IDEI and each of jDE, SaDE, EPSDE and CoDE.
"+", "−" and "≈" denote that the performance of IDEI is better than, worse than, and similar to that of the corresponding algorithm, respectively.

**Table 4**
Experimental results of IDEI and four non-DE algorithms on *CEC*2005 functions.

| Functions | CLPSO<br>Mean error ± Std. dev. /Rank | CMA-ES<br>Mean error ± Std. dev. /Rank | GL-25<br>Mean rrror ± Std. dev. /Rank | HSOGA<br>Mean error ± Std. dev. /Rank | IDEI<br>Mean error ± Std. dev. /Rank |
|---|---|---|---|---|---|
| $f_1$ | 0.00E+00 ± 0.00E+00≈ /1 | 1.58E−25 ± 3.35E−26+ /4 | 5.60E−27 ± 1.76E−26+ /3 | 5.60E−03 ± 1.75E−03+ /5 | 0.00E+00 ± 0.00E+00 /1 |
| $f_2$ | 8.40E+02 ± 1.90E+02+ /5 | 1.12E−24 ± 2.93E−25− /1 | 4.04E+01 ± 6.28E+01+ /3 | 3.00E+02 ± 8.64E+01+ /4 | 6.22E−12 ± 1.61E−11 /2 |
| $f_3$ | 1.42E+07 ± 4.19E+06+ /5 | 5.54E−21 ± 1.69E−21− /1 | 2.19E+06 ± 1.08E+06+ /3 | 1.29E+07 ± 3.18E+06+ /4 | 6.04E+04 ± 3.09E+04 /2 |
| $f_4$ | 6.99E+03 ± 1.73E+03+ /4 | 9.15E+05 ± 2.16E+06+ /5 | 9.07E+02 ± 4.25E+02+ /3 | 7.08E+02 ± 2.58E+02+ /2 | 2.46E−06 ± 6.05E−06 /1 |
| $f_5$ | 3.86E+03 ± 4.35E+02+ /5 | 2.77E−10 ± 5.04E−11− /1 | 2.51E+03 ± 1.96E+02+ /4 | 1.81E+03 ± 2.54E+02+ /3 | 5.41E+00 ± 1.55E+01 /2 |
| $f_6$ | 4.16E+00 ± 3.48E+00+ /3 | 4.78E−01 ± 1.32E+00+ /2 | 2.15E+01 ± 1.17E+00+ /4 | 6.43E+02 ± 3.32E+02+ /5 | 2.17E−03 ± 9.38E−03 /1 |
| $f_7$ | 4.51E−01 ± 8.47E−02+ /4 | 1.82E−03 ± 4.33E−03− /1 | 2.78E−02 ± 3.62E−02+ /3 | 1.82E+00 ± 3.25E−01+ /5 | 2.76E−03 ± 4.73E−03 /2 |
| $f_8$ | 2.09E+01 ± 4.41E−02≈ /2 | 2.03E+01 ± 5.72E−01− /1 | 2.09E+01 ± 5.94E−02≈ /2 | 2.09E+01 ± 4.12E−02≈ /2 | 2.09E+01 ± 4.34E−02 /2 |
| $f_9$ | 0.00E+00 ± 0.00E+00≈ /1 | 4.45E+02 ± 7.12E+01+ /5 | 2.45E+01 ± 7.35E+00+ /4 | 6.06E+00 ± 2.17E+00+ /3 | 0.00E+00 ± 0.00E+00 /1 |
| $f_{10}$ | 1.04E+02 ± 1.53E+01+ /3 | 4.63E+01 ± 1.16E+01+ /2 | 1.42E+02 ± 6.45E+01+ /4 | 1.81E+02 ± 1.33E+01+ /5 | 2.37E+01 ± 6.49E+00 /1 |
| $f_{11}$ | 2.60E+01 ± 1.63E+00+ /4 | 7.11E+00 ± 2.14E+00+ /2 | 3.27E+01 ± 7.79E+00+ /5 | 2.23E+01 ± 1.97E+00+ /3 | 3.94E+00 ± 1.93E+00 /1 |
| $f_{12}$ | 1.79E+04 ± 5.24E+03+ /4 | 1.26E+04 ± 1.74E+04+ /2 | 6.53E+04 ± 4.69E+04+ /5 | 1.56E+04 ± 5.37E+03+ /3 | 1.60E+03 ± 1.91E+03 /1 |
| $f_{13}$ | 2.06E+00 ± 2.15E−01+ /2 | 3.43E+00 ± 7.60E−01+ /3 | 6.23E+00 ± 4.88E+00+ /4 | 1.11E+01 ± 1.62E+00+ /5 | 1.06E+00 ± 1.74E−01 /1 |
| $f_{14}$ | 1.28E+01 ± 2.48E−01+ /2 | 1.47E+01 ± 3.31E−01+ /5 | 1.31E+01 ± 1.84E−01+ /4 | 1.28E+01 ± 2.80E−01+ /2 | 1.18E+01 ± 4.38E−01 /1 |
| $f_{15}$ | 5.77E+01 ± 2.76E+01+ /5 | 5.55E+02 ± 3.32E+02+ /4 | 3.04E+02 ± 1.99E+01− /1 | 3.29E+02 ± 2.80E+01− /2 | 3.76E+02 ± 7.79E+01 /3 |
| $f_{16}$ | 1.74E+02 ± 2.82E+01+ /3 | 2.98E+02 ± 2.08E+02+ /5 | 1.32E+02 ± 7.60E+01+ /2 | 1.92E+02 ± 1.29E+01+ /4 | 5.37E+01 ± 2.40E+01 /1 |
| $f_{17}$ | 2.46E+02 ± 4.81E+01+ /4 | 4.43E+02 ± 3.34E+02+ /5 | 1.61E+02 ± 6.80E+01+ /2 | 2.27E+02 ± 1.29E+01+ /3 | 6.58E+01 ± 3.70E+01 /1 |
| $f_{18}$ | 9.13E+02 ± 1.42E+00+ /4 | 9.04E+02 ± 3.01E−01≈ /1 | 9.07E+02 ± 1.48E+00+ /3 | 9.25E+02 ± 3.79E+00+ /5 | 9.04E+02 ± 1.25E+00 /1 |
| $f_{19}$ | 9.14E+02 ± 1.45E+00+ /3 | 9.16E+02 ± 6.03E+01+ /4 | 9.06E+02 ± 1.24E+00+ /2 | 9.24E+02 ± 3.33E+00+ /5 | 9.04E+02 ± 1.32E+00 /1 |
| $f_{20}$ | 9.14E+02 ± 3.62E+00+ /4 | 9.04E+02 ± 2.71E−01≈ /1 | 9.07E+02 ± 1.35E+00+ /3 | 9.25E+02 ± 3.01E+00+ /5 | 9.04E+02 ± 1.07E+00 /1 |
| $f_{21}$ | 5.00E+02 ± 3.39E−13≈ /1 | 5.00E+02 ± 2.68E−12≈ /1 | 5.00E+02 ± 4.83E−13≈ /1 | 5.00E+02 ± 3.19E−13≈ /1 | 5.00E+02 ± 9.91E−14 /1 |
| $f_{22}$ | 9.72E+02 ± 1.20E+01+ /5 | 8.26E+02 ± 1.46E+01− /1 | 9.28E+02 ± 7.04E+01+ /4 | 9.21E+02 ± 5.58E+00+ /3 | 8.85E+02 ± 1.44E+01 /2 |
| $f_{23}$ | 5.34E+02 ± 2.19E−04≈ /1 | 5.36E+02 ± 5.44E+00+ /5 | 5.34E+02 ± 4.66E−04≈ /1 | 5.34E+02 ± 2.74E−04≈ /1 | 5.34E+02 ± 4.20E−04 /1 |
| $f_{24}$ | 2.00E+02 ± 1.49E−12≈ /1 | 2.12E+02 ± 6.00E+01+ /5 | 2.00E+02 ± 5.52E−11≈ /1 | 2.01E+02 ± 1.60E−01+ /4 | 2.00E+02 ± 2.90E−14 /1 |
| $f_{25}$ | 2.00E+02 ± 1.96E+00− /1 | 2.07E+02 ± 6.07E+00− /2 | 2.17E+02 ± 1.36E−01+ /5 | 2.15E+02 ± 2.56E+01+ /4 | 2.11E+02 ± 8.78E−01 /3 |
| Average Rank | 3.08 | 2.76 | 3.04 | 3.52 | 1.40 |
| + | 18 | 15 | 20 | 21 | |
| − | 1 | 7 | 1 | 1 | |
| ≈ | 6 | 3 | 4 | 3 | |

Wilcoxon's rank sum test at a 0.05 significance level is performed between IDEI and each of CLPSO, CMA-ES, GL-25 and HSOGA.
"+", "−" and "≈" denote that the performance of IDEI is better than, worse than, and similar to that of the corresponding algorithm, respectively.

**Table 5**
Experimental results of IDEI and four non-DE algorithms on *CEC*2014 functions.

| Functions | CLPSO Mean error ± Std. dev. /Rank | CMA-ES Mean error ± Std. dev. /Rank | GL-25 Mean rrror ± Std. dev. /Rank | HSOGA Mean error ± Std. dev. /Rank | IDEI Mean error ± Std. dev. /Rank |
|---|---|---|---|---|---|
| $f_{26}$ | 9.15E+06 ± 2.15E+06+ /4 | 0.00E+00 ± 1.74E−14− /1 | 1.08E+06 ± 1.32E+06+ /3 | 1.71E+07 ± 5.69E+06+ /5 | 9.57E+03 ± 5.80E+03 /2 |
| $f_{27}$ | 1.31E+02 ± 3.14E+02+ /3 | 0.00E+00 ± 3.62E−14≈ /1 | 1.16E+03 ± 2.03E+03+ /4 | 1.49E+05 ± 1.23E+05+ /2 | 0.00E+00 ± 1.83E−14 /1 |
| $f_{28}$ | 1.92E+02 ± 2.17E+02+ /5 | 0.00E+00 ± 6.36E−14≈ /1 | 3.15E−01 ± 6.96E−01+ /3 | 4.44E+01 ± 1.72E+01+ /4 | 0.00E+00 ± 3.85E−14 /1 |
| $f_{29}$ | 7.17E+01 ± 1.68E+01+ /3 | 0.00E+00 ± 7.43E−14− /1 | 9.75E+01 ± 1.34E+01+ /5 | 7.38E+01 ± 1.40E+01+ /4 | 4.47E−01 ± 4.31E−01 /2 |
| $f_{30}$ | 2.04E+01 ± 6.04E−02− /2 | 2.00E+01 ± 2.66E−06− /1 | 2.10E+01 ± 5.37E−02+ /4 | 2.10E+01 ± 8.25E−02+ /4 | 2.09E+01 ± 4.83E−02 /3 |
| $f_{31}$ | 1.32E+01 ± 1.01E+00+ /4 | 4.79E+01 ± 7.85E+00+ /5 | 5.81E+00 ± 4.11E+00+ /3 | 2.69E+00 ± 1.11E+00+ /2 | 5.65E−01 ± 8.70E−01 /1 |
| $f_{32}$ | 1.09E−05 ± 3.96E−05+ /3 | 1.77E−03 ± 3.73E−03+ /4 | 9.00E−12 ± 1.66E−11+ /2 | 1.20E−01 ± 8.93E−02+ /5 | 0.00E+00 ± 8.99E−14 /1 |
| $f_{33}$ | 0.00E+00 ± 0.00E+00− /1 | 4.08E+02 ± 8.19E+01+ /5 | 2.24E+01 ± 5.89E+00+ /4 | 6.91E+00 ± 4.29E+00+ /3 | 3.18E−01 ± 4.74E−01 /2 |
| $f_{34}$ | 4.93E+01 ± 8.35E+00+ /3 | 5.75E+02 ± 1.39E+02+ /5 | 5.75E+01 ± 5.92E+01+ /4 | 1.73E+02 ± 1.14E+01− /1 | 2.15E+01 ± 6.02E+00 /2 |
| $f_{35}$ | 3.20E+00 ± 1.16E+00− /1 | 5.12E+03 ± 8.01E+02+ /5 | 7.25E+02 ± 3.16E+02+ /4 | 1.28E+02 ± 1.25E+02− /2 | 1.57E+02 ± 1.25E+02 /3 |
| $f_{36}$ | 2.18E+03 ± 2.74E+02− /1 | 5.17E+03 ± 9.79E+02+ /4 | 5.92E+03 ± 1.51E+03+ /5 | 4.41E+03 ± 1.26E+03+ /3 | 3.06E+03 ± 2.36E+03 /2 |
| $f_{37}$ | 3.84E−01 ± 7.28E−02− /1 | 4.25E−01 ± 6.40E−01− /2 | 2.44E+00 ± 3.67E−01+ /5 | 1.23E+00 ± 2.02E−01− /3 | 2.42E+00 ± 3.97E−01 /4 |
| $f_{38}$ | 3.25E−01 ± 4.79E−02+ /4 | 2.63E−01 ± 8.79E−02+ /2 | 2.63E−01 ± 3.76E−02+ /2 | 3.53E−01 ± 6.97E−02+ /5 | 2.06E−01 ± 4.62E−02 /1 |
| $f_{39}$ | 2.52E−01 ± 2.56E−02− /1 | 3.66E−01 ± 7.55E−02+ /4 | 2.59E−01 ± 3.35E−02− /2 | 4.77E−01 ± 3.83E−02+ /5 | 2.70E−01 ± 3.67E−02 /3 |
| $f_{40}$ | 7.37E+00 ± 9.44E−01+ /3 | 3.83E+00 ± 1.07E+00+ /2 | 1.30E+01 ± 4.77E+00+ /4 | 1.60E+01 ± 1.26E+00+ /5 | 3.21E+00 ± 1.57E+00 /1 |
| $f_{41}$ | 1.03E+01 ± 3.59E−01− /1 | 1.43E+01 ± 4.16E−01+ /5 | 1.18E+01 ± 2.90E−01+ /3 | 1.27E+01 ± 2.24E−01+ /4 | 1.07E+01 ± 5.19E−01 /2 |
| $f_{42}$ | 7.71E+05 ± 2.74E+05+ /5 | 1.84E+03 ± 4.63E+02+ /2 | 1.99E+05 ± 1.06E+05+ /3 | 4.02E+05 ± 2.56E+05+ /4 | 9.13E+02 ± 6.82E+02 /1 |
| $f_{43}$ | 1.11E+02 ± 5.12E+01+ /2 | 1.56E+02 ± 3.78E+01+ /3 | 2.37E+02 ± 3.21E+02+ /4 | 1.32E+04 ± 1.06E+04+ /5 | 1.77E+01 ± 8.34E+00 /1 |
| $f_{44}$ | 7.47E+00 ± 4.82E−01+ /3 | 1.02E+01 ± 1.53E+00+ /5 | 4.71E+00 ± 6.23E−01+ /2 | 9.08E+00 ± 1.09E+00+ /4 | 2.47E+00 ± 7.55E−01 /1 |
| $f_{45}$ | 3.23E+03 ± 1.62E+03+ /5 | 2.84E+02 ± 1.22E+02+ /3 | 1.94E+02 ± 1.23E+02+ /2 | 7.11E+02 ± 3.95E+02+ /4 | 7.91E+00 ± 3.22E+00 /1 |
| $f_{46}$ | 8.47E+04 ± 5.47E+04+ /4 | 9.82E+02 ± 3.16E+02+ /2 | 5.59E+04 ± 2.45E+04+ /3 | 2.04E+05 ± 1.20E+05+ /5 | 1.79E+02 ± 1.15E+02 /1 |
| $f_{47}$ | 2.02E+02 ± 7.25E+01+ /4 | 2.27E+02 ± 1.40E+02+ /5 | 1.52E+02 ± 6.09E+01+ /2 | 1.97E+02 ± 1.28E+02+ /3 | 1.23E+02 ± 8.83E+01 /1 |
| $f_{48}$ | 3.15E+02 ± 3.72E−06≈ /1 | 3.15E+02 ± 3.93E−12≈ /1 | 3.15E+02 ± 1.33E−09≈ /1 | 3.14E+02 ± 3.60E−02≈ /1 | 3.15E+02 ± 9.28E−13 /1 |
| $f_{49}$ | 2.23E+02 ± 4.72E+00− /3 | 2.41E+02 ± 4.69E+01+ /5 | 2.22E+02 ± 5.99E−01− /2 | 2.01E+02 ± 1.87E−01− /1 | 2.24E+02 ± 1.20E+00 /4 |
| $f_{50}$ | 2.08E+02 ± 8.30E−01+ /5 | 2.04E+02 ± 2.44E+00+ /3 | 2.07E+02 ± 1.95E+00+ /4 | 2.02E+02 ± 3.26E+00− /1 | 2.03E+02 ± 3.80E−01 /2 |
| $f_{51}$ | 1.00E+02 ± 9.15E−02≈ /1 | 1.13E+02 ± 4.60E+01+ /5 | 1.00E+02 ± 4.64E−02≈ /1 | 1.00E+02 ± 6.04E−02≈ /1 | 1.00E+02 ± 3.28E−02 /1 |
| $f_{52}$ | 4.13E+02 ± 5.77E+00+ /5 | 4.07E+02 ± 1.41E+02+ /4 | 3.02E+02 ± 8.78E−01− /1 | 4.05E+02 ± 3.40E+01+ /3 | 3.66E+02 ± 4.78E+01 /2 |
| $f_{53}$ | 9.20E+02 ± 5.66E+01+ /4 | 3.69E+03 ± 2.68E+03+ /5 | 8.80E+02 ± .04E+01+ /3 | 4.11E+02 ± 1.06E+01− /1 | 7.97E+02 ± 4.11E+01 /2 |
| $f_{54}$ | 1.01E+03 ± 9.84E+01+ /4 | 8.14E+02 ± 9.54E+01+ /3 | 1.01E+03 ± 1.03E+02+ /4 | 2.22E+02 ± 1.38E+00− /1 | 7.15E+02 ± 1.06E+02 /2 |
| $f_{55}$ | 3.69E+03 ± 9.60E+02+ /5 | 2.29E+03 ± 6.05E+02+ /4 | 1.33E+03 ± 2.74E+02+ /3 | 1.14E+03 ± 5.37E+02+ /2 | 8.96E+02 ± 3.98E+02 /1 |
| Average Rank | 3.03 | 3.27 | 3.07 | 3.20 | 1.73 |
| + | 20 | 23 | 25 | 21 | |
| − | 8 | 4 | 3 | 7 | |
| ≈ | 2 | 3 | 2 | 2 | |

Wilcoxon's rank sum test at a 0.05 significance level is performed between IDEI and each of CLPSO, CMA-ES, GL-25 and HSOGA.
"+", "−" and "≈" denote that the performance of IDEI is better than, worse than, and similar to that of the corresponding algorithm, respectively.

results, where the time represents the CUP time that PFEV or $FES_{max}$ is met, and the best results are marked by bold on each function.

From Table 6, we see that IDEI is faster than others except for EPSDE on $f_{26}$ and CoDE on $f_4$ and $f_{26}$. It should be mentioned that DS strategy could alleviates a waste of a large number of information before a better solution is found, yet it reduces the rate of convergence on some unimodal functions.

### 3.5. An application

As an application, we consider Spread Spectrum Radar Polly phase Code Design (SSRPPCD) [47]. This problem aims at selecting an appropriate waveform and can be modeled as min-max nonlinear non-convex optimization problem:

$$\min_{x \in X} f(x) = \max\{\phi_1(x), \ldots, \phi_{2m}(x)\},$$

where $X = \{(x_1, \ldots, x_n) \in R^n | 0 \le x_j \le 2\pi \text{ for } j = 1, \ldots, n\}$, $m = 2n - 1$, $\phi_{m+i}(x) = -\phi_i(x)$ for $i = 1, \ldots, m$ and

$$\begin{cases} \phi_{2i-1}(x) = \sum_{j=1}^{n} \cos\left( \sum_{k=|2i-j-1|+1}^{j} x_k \right), & i = 1, \ldots, n \\ \phi_{2i}(x) = 0.5 + \sum_{j=i+1}^{n} \cos\left( \sum_{k=|2i-j|+1}^{j} x_k \right), & i = 1, \ldots, n-1. \end{cases}$$

Fig. 6 shows its image with $n = 2$. Clearly, the objective function is piecewise smooth and has numerous local optima.

We use IDEI to solve this problem. Table 7 reports numerical results of IDEI over 25 runs with 50,000, 100,000 and 150,000 FES, respectively. To make a comparison, Table 7 also lists numerical results of SaDE [25] and JADE [21] obtained in [46], where the best algorithm among them is marked by bold at each case. From Table 7,

**Table 6**
Time of IDEI and jDE, SaDE, EPSDE and CoDE.

| Function | | IDEI | jDE | SaDE | EPSDE | CoDE |
|---|---|---|---|---|---|---|
| | | Time | Time | Time | Time | Time |
| Unimodal | $f_4$ | 16.564 s | 17.858 s | 50.151 s | 23.596 s | **14.093 s** |
| | $f_{26}$ | 16.415 s | 18.374 s | 53.313 s | **9.182 s** | 12.654 s |
| Multimodal | $f_{11}$ | **31.475 s** | 96.859 s | 113.444 s | 84.891 s | 73.776 s |
| | $f_{31}$ | **7.146 s** | 36.788 s | 88.973 s | 62.053 s | 13.432 s |
| | $f_{34}$ | **4.985 s** | 10.836 s | 20.724 s | 22.191 s | 14.074 s |

**Table 7**
Numerical results of IDEI on SSRPPCD.

| FES | Algorithms | Best (Result) | Worst (Result) | Average value | Standard deviation |
|---|---|---|---|---|---|
| 50,000 | SaDE | 1.25E+00 | 1.78E+00 | 1.47E+00 | 1.05E−01 |
| | JADE | 1.22E+00 | 1.49E+00 | 1.34E+00 | 8.40E−02 |
| | **IDEI** | 8.77E−01 | 2.56E+00 | 1.33E+00 | 3.06E−01 |
| 100,000 | SaDE | 8.35E−01 | 1.55E+00 | 1.32E+00 | 1.46E−01 |
| | JADE | 1.05E+00 | 1.34E+00 | 1.23E+00 | 8.26E−02 |
| | **IDEI** | 8.89E−01 | 1.67E+00 | 1.21E+00 | 2.03E−01 |
| 150,000 | **SaDE** | 5.41E−01 | 1.39E+00 | 1.10E+00 | 2.61E−01 |
| | JADE | 1.03E+00 | 1.34E+00 | 1.18E+00 | 7.46E−02 |
| | **IDEI** | 6.58E−01 | 1.24E+00 | 1.10E+00 | 1.74E−01 |

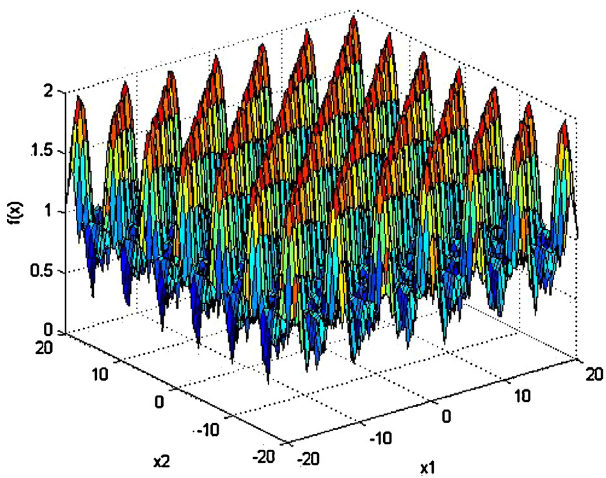



**Fig. 6.** Image of objective function $f$ with $n = 2$.

we see that IDEI obtains the best results at all cases. In particular, IDEI is better than SaDE and JADE for 50,000 and 100,000 FES, IDEI and SaDE have the same results for 150,000 FES, while IDEI has a smaller standard deviation than SaDE. Thus, IDEI is an efficient algorithm to this problem.

## 4. Conclusion

This paper proposes a differential evolution via incorporating fitness values of guiding and original individuals and positions of target and trail individuals into parameter setting and selection strategy respectively, which are rarely used and might be helpful to improve their performances. In IDEI, we propose a CM strategy, a GIP setting and a DS strategy by using two mixed mutation strategies with a prescribed constant probability, the fitness values of original and guiding individuals simultaneously, and fitness value and position of individual, respectively. Compared with the mutant strategy in [36], CM strategy and GIP setting not only do not require to set the generation index threshold and a complicated method to divide population into the superior and inferior subpopulation, but also can make the search in a suitable area and generate offspring by using better information from parent and mutant individuals. Moreover, differing from greedy selection and tournament selection strategies, DS strategy not only enhances the population diversity, but also alleviates a waste of a large number of information before a better solution is found. Then the proposed algorithm can effectively balance the exploration and exploitation during the evolution process. Finally, numerical experiments on 55 benchmark functions in the CEC 2005 and CEC 2014 special session and comparisons with four existing DE variants and four non-DE variants show that the proposed algorithm significantly outperforms other DE variants. Meanwhile, a practical application of IDEI to Spread Spectrum Radar Polly phase Code Design (SSRPPCD) [47] is given.

Further research can be focused on designing a new method to avoid the difficulty of setting the prescribed probability parameter in the CM strategy, improving and generalizing the DS strategies to other algorithms, and enhancing the performance of DE by employing the fitness value and position of individual simultaneously.

## Acknowledgements

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.

## References

[1] O.A. Abdul-Rahman, M. Munetomo, K. Akama, An adaptive parameter binary-real coded genetic algorithm for constraint optimization problems: performance analysis and estimation of optimal control parameters, Inform. Sci. (2013) 54–86.
[2] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces, Technical Report TR-95-012, ICSI.http://www.icsi.berkeley.edu/storn/litera.html.
[3] D.Y. Sha, C.Y. Hsu, A new particle swarm optimization for the open shop scheduling problem, Comput. Oper. Res. (2008) 3243–3261.
[4] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. (2007) 459–471.
[5] Y. Tan, Y. Zhu, Fireworks algorithm for optimization, in: Proceedings of the 2nd International Conference on Swarm Intelligence, Lecture Notes in Computer Science, 2010, pp. 355–364.
[6] Q.H. Wu, Y. Wang, Z.P. Lü, A tabu search based hybrid evolutionary algorithm for the max-cut problem, Appl. Soft Comput. 82 (2015) 7–837.
[7] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.
[8] T. Rogalsky, R.W. Derksen, S. Kocabiyik, Differential evolution in aerodynamic optimization, in: Proceedings of the 46th Annual Conference on Canadian Aeronautics and Space Institute, 1999, pp. 29–36.
[9] 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.
[10] F.-S. Wang, H.-J. Jang, Parameter estimation of bio-reaction model by hybrid differential evolution, in: Proc. IEEE Trans. Congr. Evol. Comput., vol. 1, IEEE Press, Piscataway, NJ, 2000, pp. 410–417.
[11] M. Omran, A.P. Engelbrecht, A. Salman, Differential evolution methods for unsupervised image classification, in: Proc. 7th Congr. Evol. Comput., vol. 2, IEEE Press, Piscataway, NJ, 2005, pp. 966–973.
[12] O. Hrstka, A. Kucerova, Improvement of real coded genetic algorithm based on differential operators preventing premature convergence, Adv. Eng. Softw. 35 (2004) 237–246.
[13] 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 Jun) 281–295.
[14] F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experiment analysis, Artif. Intell. Rev. 3 (1) (2010 Feb) 61–106.
[15] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011 Feb) 4–31.
[16] G.D. Tollo, F. Lardeux, J. Maturana, F. Saubion, An experimental study of adaptive control for evolutionary algorithms, Appl. Soft Comput. 35 (2015) 9–372.
[17] S. Das, A. Konar, U.K. Chakraborty, Two improved differential evolution schemes for faster global search, in: Proc. IEEE Congr. Evol. Comput. GECCO, 2005, pp. 991–998.

[18] J. Liu, J. Lampinen, A fuzzy adaptive differential evolution algorithm, Soft Comput. 9 (6) (2005) 448–462.
[19] M.G. Epitropakis, V.P. Plagianakos, M.N. Vrahatis, Balancing the exploration and exploitation capabilities of the differential evolution algorithm, in: Proc. IEEE Congr. Evol. Comput., vol. 1, Hong Kong, 2008, pp. 2686–2693.
[20] 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.
[21] J. Zhang, A.C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.
[22] L. Tang, Y. Zhao, J. Liu, An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production, IEEE Trans. Evol. Comput. 18 (2) (2014) 209–225.
[23] M.G. Epitropakis, D.K. Tasoulis, N.G. Pavlidis, V.P. Plagianakos, M.N. Vrahatis, Enhancing differential evolution utilizing proximity-based mutation operators, IEEE Trans. Evol. Comput. 15 (1) (2011) 99–119.
[24] 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.
[25] A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: Proc. IEEE Congr. Evol. Comput., vol. 2, 2005, pp. 1785–1791.
[26] S. Biswas, S. Kundu, S. Das, A.V. Vasilakos, Teaching and learning best differential evolution with self adaptation for real parameter optimization, in: Proc. IEEE Congr. Evol. Comput., Cancun, Mexico, 2013, pp. 1115–1122.
[27] 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.
[28] S.Z. Zhao, P.N. Suganthan, Comprehensive comparison of convergence performance of optimization algorithms based on nonparametric statistical tests, in: Proc. IEEE Congr. Evol. Comput., Brisbane, QLD, Australia, 2012, pp. 1–7.
[29] W. Gong, Z. Cai, Differential evolution with ranking-based mutation operators, IEEE Trans. Cybern. 43 (6) (2013) 2066–2081.
[30] N. Padhye, P. Mittal, K. Ded, Differential evolution: performances and analysis, in: Proc. IEEE Congr. Evol. Comput., Cancun, Mexico, 2013, pp. 1960–1967.
[31] J. Lampinen, I. Zelinka, On stagnation of the differential evolution algorithm, in: Proc. 6th Int. Mendel Conf. Soft Comput., Shenyang, china, 2000, pp. 76–83.
[32] J. Ronkkonen, S. Kukkonen, K.V. Price, Real-parameter optimization with differential evolution, in: Proc. IEEE Congr. Evol. Comput., vol. 1, Edinburgh, UK, 2005, pp. 506–513.
[33] 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. Cybern. 10 (6) (2006) 646–657.
[34] H. Abbass, The self-adaptive Pareto differential evolution algorithm, in: Proc. IEEE Congr. Evol. Comput., vol. 1, Honolulu, HI, USA, 2002, pp. 813–836.
[35] J. Teo, Exploring dynamic self-adaptive populations in differential evolution, Soft Comput. 10 (8) (2006) 673–686.
[36] L.X. Tang, Y. Dong, J.Y. Liu, Differential evolution with an individual-dependent mechanism, IEEE Trans. Evol. Comput. 19 (4) (2015) 560–574.
[37] A. Qamar, A. Jamil, J. Hajira, A novel tournament selection based differential evolution variant for continuous optimization problem, Math. Probl. Eng. 2015 (2015) 1–21.
[38] M. Yang, C.H. Li, Z.H. Cai, J. Guan, Differential evolution with auto-enhanced population diversity, IEEE Trans. Evol. Cybern. 45 (2) (2015) 302–315.
[39] K. Price, R. Storn, J. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, vol. 141, no. 2, Springer, Heidelberg, 2005, pp. 1–24.
[40] 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, Tech. Rep. KanGAL, 2005005, Nanyang Technol. Univ. Singapore, IIT Kanpur, India, 2005.
[41] J.J. Liang, P.N. Suganthan, K. Deb, Novel composition test functions for numerical global optimization, in: Proceedings of IEEE Swarm Intelligence Symposium, Pasadena, CA, 2005, pp. 68–75.
[42] S.M. Guo, C.C. Yang, P.H. Hsu, J.S.-H. Tsai, Improving differential evolution with a successful-parent-selecting framework, IEEE Trans. Evol. Comput. 19 (5) (2015) 717–730.
[43] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195.
[44] C. Garcia-Martinez, M. Lozano, F. Herrera, D. Molina, A.M. Sanchez, Global and local real-coded genetic algorithms based on parent-centric crossover operators, Eur. J. Oper. Res. 185 (3) (2008) 1088–1113.
[45] Z.Y. Jiang, Z.X. Cai, Y. Wang, Hybrid self-adaptive orthogonal genetic algorithm for solving global optimization problems, J. Softw. 21 (6) (2010) 1296–1307.
[46] 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.
[47] S. Das, P.N. Suganthan, Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems, Technical Report, Jadavpur Univ. India and Nanyang Technological Univ., Singapore, 2010.
[48] 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./Zhenzhou Univ., Singapore/China, 2013.