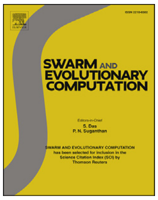
ARTICLE IN PRESS

Swarm and Evolutionary Computation xxx (xxxx) xxx

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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


SWARM AND EVOLUTIONARY COMPUTATION

A multi-role based differential evolution

Ling Gui [a], Xuewen Xia [a,*], Fei Yu [a], Hongrun Wu [a], Ruifeng Wu [b], Bo Wei [b], Yinglong Zhang [b], Xiong Li [b], Guoliang He [c]

[a] *College of Physics and Information Engineering, Minnan Normal University, Zhangzhou, 363000, China*
[b] *School of Software, East China Jiaotong University, Jiangxi, 330013, China*
[c] *School of Computer, Wuhan University, Hubei, 430072, China*

ARTICLE INFO

Keywords:
Differential evolution
Multiple roles
Adaption of population size
Control parameters
Trial vector generation strategies

ABSTRACT

Differential evolution (DE) is an efficient and powerful stochastic optimization algorithm. Extensive studies in recent years have verified that different trial vector generation strategies and associated control parameters offer distinct characteristics on different problems. To take full advantages of them, different ensemble methods of trial vector generation strategies and control parameters based on various adaptive strategies have been proposed during the last decade. Aiming to organically integrate merits of some popular generation strategies and control parameters, and then utilize distinct advantages of them, a multi-role based DE (MRDE) is proposed in this paper. In MRDE, the entire population is divided into multiple small-sized groups, and individuals in each group are assigned with different roles in each generation according to their fitness. Based on the assigned role, an individual selects its own trial vector generation strategies and control parameters from a pool to breed offspring. Moreover, an adaptive strategy for population size is used to rationally distribute the computational resources, which is beneficial for speeding up the convergence. Furthermore, a regroup strategy enables individuals to play different roles in different generations, which is favorable for diversifying the search behaviors. The performance of MRDE is compared with that of ten state-of-the-art DE variants on CEC2017 test suite with three dimension cases, and the experimental results demonstrate the competitive and reliable performance of MRDE. In addition, the effectiveness of the newly proposed strategies is also verified through comparison experiments.

1. Introduction

Differential evolution (DE) [1] is a stochastic population-based search method proposed by Storn and Price in 1997. In each generation, an individual in DE uses trial vector generation strategies (i.e., mutation and crossover strategies) and related control parameters to breed an offspring, and then using a selection operator to save a better one from the individual and the offspring to the next generation. However, unlike the traditional mutation operator applied in other evolutionary algorithms (EAs) [2–4], the mutation operator in DE provides a basic information of search direction and step size for an individual relying on the differential information (i.e., distance and direction) among different individuals. Due to the simplicity and higher efficiency, DE has been widely adopted in the field of numerous science problems and technology applications, such as scheduling problem [5,6], structural optimization [7], satellite image registration [8], biogeography [9], engineering design [10], and so on.

The performance of DE mainly depends on mutation strategies, crossover operators, and control parameters (i.e., scaling factor F and crossover rate CR) since they determine the cooperation mechanism among different individuals. Thus, many researchers pull much attention on the issues, and various characteristics of different strategies and parameters are revealed in the last decades. For example, “DE/rand/1/bin” mutation is known to be robust but less efficient in terms of convergence rate [11]. On the contrary, “DE/current-to-best/1/bin” mutation can be regarded as a more greedy DE variant, which is favorable for convergence speed. However, it may lead to premature convergence when solving multimodal problems [12,13]. Moreover, extensive studies manifest that F and CR also have different characteristics, in terms of population diversity and convergence speed [14].

Thus, it seems to be a feasible method that selecting an optimal trial vector generation strategy and associated F and CR for a specific problem based on the characteristics of the problem. Nevertheless,

[*] Corresponding author.
E-mail address: xwxia@whu.edu.cn (X. Xia).

https://doi.org/10.1016/j.swevo.2019.03.003
Received 9 June 2018; Received in revised form 13 December 2018; Accepted 3 March 2019
Available online XXX

Please cite this article as: L. Gui, et al., A multi-role based differential evolution, Swarm and Evolutionary Computation, https://doi.org/10.1016/j.swevo.2019.03.003

considering many real applications are black-box problems, it is unrealistic to predefine a trial vector generation strategy and parameters in advance. Furthermore, it is often time-consuming that applying a trial-and-error scheme to determine the most favorable strategy and its associated parameters. Therefor, many researchers pour much attention into (self-)adaptive mechanisms, which are based on experiences of the current population rather than a user's prior knowledge, in DE community during the last few years [15–18]. Moreover, considering that many popular parameters and mutation strategies have their own distinct properties, many scholars pay more attention on ensemble strategies, in which different existing parameters and mutation strategies can be organically integrated by various hybrid schemes [19–22].

Furthermore, many scholars introduced multi-population techniques into DE, and proposed some outstanding DE variants [17,23–26]. In this type of research, to organically integrate various useful information in different subpopulations, an efficient information sharing mechanism needs to be proposed. However, similar to selecting proper parameters, designing an optimal information sharing mechanism based on a trial-and-error scheme also requires high computational costs.

Inspired by the outstanding researches introduced above, we propose a multi-role based DE (MRDE) in this paper. The main characteristics of MRDE can be summarized as follows.

1. Three popular breeding strategies (i.e., trial vector generation strategy and associated control parameters) with different characteristics on exploration and exploitation are save in a pool, which can be used by any individuals.
2. The entire population in MRDE is randomly divided into multiple subpopulations. In each generation, different individuals in every subpopulation are assigned different roles according the performance of them. Based on the assigned role, an individual selects a proper trial vector generation strategy and associated control parameters from the candidate pool to breed offspring. Thus, the individual can conduct different breeding strategies in different generations.
3. An adaptive strategy is used to change population size aiming to rational allocate computational resources, and then improve the convergence speed as well as increase population diversity in different evolutionary stages.

The rest of this paper is organized as follows. Section 2 describes the framework of the canonical DE and reviews some DE variants. The details of MRDE algorithm are described in Section 3. Extensive experiments between MRDE and other ten state-of-the-art DEs on CEC2017 test suite are introduced in Section 4. Moreover, to verify the performance of the new introduced strategies in MRDE, sensitivity analysis of the strategies is detailed in Section 5. Finally, conclusions are given in Section 6.

2. DE

2.1. Canonical DE algorithm

Like the genetic algorithm (GA), DE consists of four basic steps, including initialization, mutation, crossover, and selection. The latter three steps repeated into the subsequent generations cause the population in DE to search for more favorable solutions for a specific problem.

2.1.1. Initialization

In the initialization procedure, i.e., the generation $t = 1$, the population $\{\mathbf{X}_i^t = (x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t) \mid i = 1, 2, \ldots, N\}$ is randomly generated within a feasible solution space, where D is the dimensionality of the problem and N is the population size. After the initialization, DE successively executes mutation, crossover, and selection operators in each generation.

2.1.2. Mutation

In generation t, DE creates a mutant vector $\{\mathbf{V}_i^t = (v_{i,1}^t, v_{i,2}^t, \ldots, v_{i,D}^t) \mid i = 1, 2, \ldots, N\}$ for each target vector $\mathbf{X}_i^t$ by a mutation operator. There are five most frequently referred mutation strategies according to ways of generating its corresponding mutant vector $\mathbf{V}_i^t$. The details of the strategies are listed as follows:

1) "DE/rand/1": $\mathbf{V}_i^t = \mathbf{X}_{r_1}^t + F \cdot (\mathbf{X}_{r_2}^t - \mathbf{X}_{r_3}^t)$
2) "DE/best/1": $\mathbf{V}_i^t = \mathbf{X}_{best}^t + F \cdot (\mathbf{X}_{r_1}^t - \mathbf{X}_{r_2}^t)$
3) "DE/current-to-best/1": $\mathbf{V}_i^t = \mathbf{X}_i^t + F \cdot (\mathbf{X}_{best}^t - \mathbf{X}_i^t) + F \cdot (\mathbf{X}_{r_1}^t - \mathbf{X}_{r_2}^t)$
4) "DE/best/2": $\mathbf{V}_i^t = \mathbf{X}_{best}^t + F \cdot (\mathbf{X}_{r_1}^t - \mathbf{X}_{r_2}^t) + F \cdot (\mathbf{X}_{r_3}^t - \mathbf{X}_{r_4}^t)$
5) "DE/rand/2": $\mathbf{V}_i^t = \mathbf{X}_{r_1}^t + F \cdot (\mathbf{X}_{r_2}^t - \mathbf{X}_{r_3}^t) + F \cdot (\mathbf{X}_{r_4}^t - \mathbf{X}_{r_5}^t)$

In the above equations, r_1, r_2, r_3, r_4, and r_5 are mutually exclusive integers randomly selected from the set $\{1, 2, \ldots, N\} \setminus \{i\}$. $\mathbf{X}_{best}^t$ denotes the best individual of a population at generation t. The scaling factor F is a positive parameter used to amplify difference vectors.

2.1.3. Crossover

After the mutation operator, DE executes a crossover operator to generate a trial vector $\mathbf{U}_i^t = (u_{i,1}^t, u_{i,2}^t, \ldots, u_{i,D}^t)$ based on $\mathbf{X}_i^t$ and $\mathbf{V}_i^t$. Although there are many different crossover strategies, such as the arithmetic crossover and the eigenvector-based crossover [27], the binomial crossover defined as (1) is widely accepted in many researches [11,19,20,28].

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

where $rand_j$ is a random number uniformly distributed in the range [0, 1], the crossover rate CR determines the probability that $u_{i,j}^t$ is copied from $v_{i,j}^t$ or $x_{i,j}^t$, and j_{rand} is a randomly selected integer in the range [1, D].

Note that $u_{i,j}^t$ may exceed out of the corresponding upper or lower boundaries. In this case, it needs to be reset within the feasible region. A common simple method used to deal with the problem can be detailed as follows [20].

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

where U_j and L_j are the upper and lower boundaries of the jth dimension, respectively.

2.1.4. Selection

Once the trial vector $\mathbf{U}_i^t$ is produced, a widely accepted greedy selection strategy is used to determine whether $\mathbf{X}_i^t$ or $\mathbf{U}_i^t$ can be saved into the next generation. Without loss of generality, minimization problems are considered in this paper. Thus, the selection operator can be described as follows.

$$\mathbf{X}_i^{t+1} = \begin{cases} \mathbf{X}_i^t, & \text{if } f(\mathbf{X}_i^t) < f(\mathbf{U}_i^t) \\ \mathbf{U}_i^t, & \text{otherwise} \end{cases} \tag{3}$$

where $f(\mathbf{X}_i^t)$ and $f(\mathbf{U}_i^t)$ are the objective values of $\mathbf{X}_i^t$ and $\mathbf{U}_i^t$, respectively.

2.2. Previous works

DE has been popularly studied since it is proposed due to its effectiveness and simplicity, and numerous DE variants have been developed during the last few years. Since the performance of DE highly depends on the trail vector generating strategy and associated parameters, majority of researches are focus on designing proper trail vector generating strategy and selecting optimal parameters.

2.2.1. Adjustment of parameters

There are three importance parameters involved in DE, including population size N, scaling factor F, and crossover rate CR. A few studies during the last few years have drawn different conclusions. For example, Storn and Price [29] stated that N should be between $5D$ and $10D$, F should be in the range [0.4, 1.0], and CR can be set to 0.1 or 0.9. In Ref. [30], the extensive experiments suggested that N should be between $3D$ and $8D$, F should be 0.6, and CR should be in the range [0.3, 1.0]. Although there are already various suggestions for parameter settings [12,30,31], the interaction within the different parameters is still complicated and has not been completely understood. Furthermore, the characteristics of many real applications are unknown as well as the search process of DE is dynamical, which cause these predefined fixed parameters cannot meet requirements of the dynamical optimization process. Thus, various dynamical adjustments of the parameters are proposed in recent years. For example, Zaharie [32] proposed an adaptive DE algorithm (ADE) with a variable N. The experimental results manifest that the convergence rate relates to the decreasing rate of it. In Ref. [33], a linearly reduced F was proposed to satisfy the different requirements of population diversity in different evolutionary stages. Considering that different parameters have their own merits, Wang [34] proposed a composite DE (CoDE), in which each individual randomly selects its own parameters (i.e., F and CR) from three predefined parameter settings.

To take full advantages of the population's experience, different adaptive strategies are presented to layout more satisfactory tuning methods for parameters in DE [15–17]. The main superiority of adaptive strategies is that less prior knowledge or characteristics of a specific problem is required. For instance, Liu and Lampinen [35] introduced a fuzzy adaptive DE (FADE), in which two fuzzy logic controllers are adopted to tune F and CR. The experiment results show that FADE has higher converge speed than the canonical DE especially when the dimensionality of the problem is high or a concerned problem is complicated. In jDE [16], Brest et al. set F and CR as random numbers within certain ranges or as the values of latest generation from predefined probability. The experiment results demonstrate that jDE is better than or at least same as the traditional DE. Moreover, Qin et al. [19] developed a self-adaptive DE (SaDE), in which both F and CR are gradually self-adaptive according to a novel way of learning. A more suitable generation strategy along with its parameter settings could be determined adaptively to match different phases of the search process. Based on this work, Pan et al. [6] proposed another self-adaptive algorithm, named as SspDE, in which each target vector has its own generation strategy. Under the condition, different values of F and CR match different search phases according to their previous experience. In JADE proposed by Zhang [11], information of recent successful F and CR is applied to adjust the subsequent F and CR, respectively. As a result, much helpful knowledge of elites in the population can be used to guide other individuals to conduct more efficient search behaviors.

2.2.2. Generation of trail vector

The five widely accepted mutation strategies introduced in Section 2.1 have their own merits. For instance, it has been observed that "DE/current-to-best/1" strategy performs poorly on exploration when solving multimodal problems [12]. On the contrary, "DE/rand/1" is known to be robust but less efficient in terms of convergence speed [11] while "DE/best/1" and "DE/best/2" are beneficial for convergence speed. However, considering many real applications are "black-box" problems, designing an optimal mutation strategy through trial-and-error is time-consuming and unrealistic. Thus, organic integrating different mutation strategies has captured many researchers' attention.

1) The first type of the integration is selecting proper mutation strategies to meet requirements of different evolutionary stages. For instance, in TSDE [36], three mutation strategies beneficial for exploration are used to enhance population's exploration ability at the former stage. On the contrary, in the latter stage, two mutation strategies favorable for exploitation are applied to improve the convergence speed. In UMDE [21], the entire evolutionary process is divided into three stages based on the degree of population disperse. Accordingly, distinct mutation strategies are applied in different stages aiming to satisfy the different requirements of exploration and exploitation.
2) The second type of integration is that different trial vector generation strategies have their own probabilities to breed offspring during the entire evolutionary process. For example, Wang et al. [20] choose three trial vector generation strategies as well as control parameters to constitute a strategy candidate pool in CoDE. In each generation, an individual randomly selects a strategy from the pool to create an offspring. Furthermore, to strengthen the adaptive ability of DE on different problems, Qin [19] selected "DE/current-to-best/1" and "DE/rand/1" as two basic strategies in SaDE [19]. During the optimization process, the population in SaDE adapts probabilities of generating offsprings by the two predefined strategies based on their success ratios in the last few generations. In EPSDE [22], distinct trail vector generation strategies (i.e., mutation and crossover strategies) as well as control parameters are saved in two pools, respectively, which coexists throughout the evolutionary process. Different generation strategies and control parameters compete to breed offspring based on their performance. The motivation of this type improvements is that the most suitable mutation strategy in the last few generations may display better performance in future generations. Unlike the above-mentioned adaptation mechanism, which can be regarded as low-level adaptation, a high-level adaptation method is introduced in the ensemble of multiple DE variants (EDEV) [37], in which three highly popular and efficient DE variants, including JADE, CoDE, and EPSDE [22], are selected adaptively by individuals based on the performance of the DE variants. Through this manner, the most efficient trail vector generation strategy in these DE variants is expected to obtain the most computational resources.
3) Furthermore, inspired by the population topology mechanism in particle swarm optimization (PSO) [38,39], many topology-based DE variants are emerged during the last few years. One of the earliest topological neighborhood DE works was carried out by Tasoulis [23]. The algorithm divides the whole population into many small-sized groups with a ring topology. This method was modified and further improved by Weber et al. [40,41]. In Refs. [42,43], Das et al. improved "DE/current-to-best/1" strategy to "DE/current-to-lbest/1" by introducing a local neighborhood model, in which each vector is mutated by using the best individual solution in its small neighborhood. In 2011, Elsayed [44] divided the entire population into four distinct groups, each of which executes one type of mutation strategy on their members. To realize an adapted ensemble of different mutation strategies, Wu [45] proposed a novel DE variant, named as MPEDE. In MPEDE, the entire population is dynamically partitioned into several groups. Based on the historical performance of the groups, different groups sizes are assigned for them. In this way, the best mutation strategy is expected to obtain more computational resources.

In addition, many studies verify that some novel mutation strategies [46] and the covariance matrix adaptation [47] also offer favorable performance. Furthermore, different choosing strategies for solutions when carrying out the various performance [48].

算法提出背景：1.好的策略和参数组合可以提高算法性能；2.通过调整种群大小分配资源；3.子种群可以保留种群的多样性。

3. Multi-role based DE

Extensive studies during the last few years have verified that organic integrating different outstanding strategies and parameters is a promising method to enhance the comprehensive performance of DE [19,22,34]. Furthermore, we regard that it is feasible method for speeding up of convergence that rational distributing computational resources through changing population size in different evolutionary stages. Moreover, subpopulation mechanisms have displayed their positive performance, in term of population diversity. Inspired by these observations, we develop a MRDE algorithm, in which the entire population is divided into multiple groups with a same size. In each group, an individual selects its own trial vector generation strategy and associated control parameters from a candidate pool based on its fitness. Moreover, an adaptive strategy for change of population size is performed during the evolutionary process to realize the rational allocation of computational resources. The framework of MRDE and the core idea behind it are elucidated as follows.

3.1. Framework of MRDE

At the beginning of evolutionary process, the population is randomly divided into *gn* groups, and the size of each group is *gs*. After that, an individual in each group is assigned a proper role in each generation according to the individual's relative performance in its own group. Based on the assigned role, the individual can adopt its own mutation operator and parameter settings from a candidate pool. As a result, different individuals in a same group can perform distinct search behaviors in every generation. Moreover, elites in all groups are saved in an archive which can be re-utilized by other individuals at an appropriate time. During the evolutionary process, to rational assign computational resources, the population size in MRDE is dynamically adjusted relying on the population performance. To enrich individuals' search behaviors, the multiple groups can be dynamical regrouped after the adjustment of population size. In this case, an individual can play different roles in different evolutionary periods.

The flowchart of MRDE is demonstrated in Fig. 1, and many new introduced strategies in it are detailed as following sections.

3.2. Multiple roles in each group

最好、最差的各体的role是elite、inferior，其他的是mediocrities

In each group, individuals can be categoriz including elites, inferiors, and mediocrities, based on their fitness. Specifically, the best individual and the worst individual in each group are regarded as an elite and an inferior, respectively. Other individuals are seen as mediocrities.

Generally speaking, the elite needs to pour more attention on exploitation ability aiming to enhance solutions accuracy. On the contrary, it is a favorable choice for the inferior to focus on exploration intending to find out more promising solutions in other regions. For those mediocrities, assigning a trade-off search behavior is a pleasurable choice. Hence, it is crucial for MRDE that selecting appropriate mutation operators and control parameters for different individual based on their roles since it determines whether favorable characteristics of the different operators and parameters can be effectively utilized.

Since DE is proposed in 1997, a few of mutation operators with distinct properties are proposed during the last two decades [11,14,19,20,36,49]. Based on these researches, MRDE utilizes a pool to save a few mutation strategies and corresponding control parameters with distinct characteristics. During the evolutionary process, every individual selects its own strategies and parameters from the pool to breed offspring. Concretely, settings of selected mutation strategies and control parameters are described as follows.

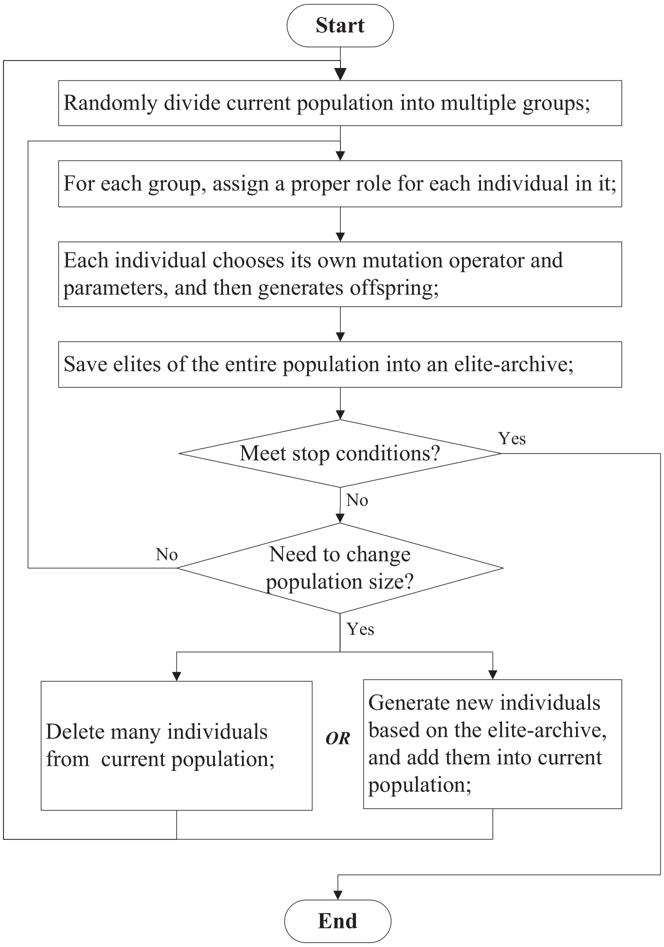

Start
Randomly divide current population into multiple groups;
For each group, assign a proper role for each individual in it;
Each individual chooses its own mutation operator and parameters, and then generates offspring;
Save elites of the entire population into an elite-archive;
Meet stop conditions?
Yes
No
Need to change population size?
No
Yes
Delete many individuals from current population;
OR
Generate new individuals based on the elite-archive, and add them into current population;
End

Fig. 1. The flowchart of MRDE.

候选策略和参数池

- *Setting* 1: [DE/current-to-best/1, $F = 0.6$, $CR = 0.1$], which is applied to improve exploitation ability;
- *Setting* 2: [DE/current-to-lbest/1, $F = 0.8$, $CR = 0.5$], which can achieve a trade-off between exploration and exploitation; and
- *Setting* 3: [DE/rand/1, $F = 1.0$, $CR = 0.9$], which is used to enhance exploration ability.

Note that, the symbol "lbest" in "DE/current-to-lbest/1" in *Setting* 2 denotes the local best solution of a group that a target vector belongs to.

In each generation, every individual chooses its own mutation strategy, *F*, and *CR* from the pool. Concretely, the elite in every group selects *Setting* 1 to breeding offspring aiming to improve exploitation ability of the population, while the inferior in every group adopts *Setting* 3 to generate offspring with high exploration ability. The mediocrities in every group employs *Setting* 2 to yield offspring with a trade-off behavior between exploration and exploitation.

Because the relatively performance of an individual in a group may be changed during the evolutionary process, the individual can play different roles in different generations. Consequently, the individual can select different mutation operators and control parameters from the pool in different generations, and then performs diverse search behaviors.

3.3. Save elites

Saving elites of all groups and making proper use of them at appropriate time may be a favorable choice. We regard there are at least two advantages. The one is beneficial for reducing computational consumption. The other one is that much helpful knowledge can be extracted from the elites.

Thus, in this work, elites in each generation are saved in an archive *Ae*. At the beginning of evolutionary process ($t = 1$), all the local best solutions $\mathbf{X}^t_{lbest_i}$ ($1 \le i \le gn$) of each group are saved in *Ae*. Thereafter, the global best solution $\mathbf{X}^t_{best}$ in each generation ($t > 1$) is added into To keep the size of *Ae* constant, an individual must be replaced by $\mathbf{X}^t_{best}$. Generally, there are two common methods for the replacement operator. The one is that using $\mathbf{X}^t_{best}$ to replace the worst individual in *Ae*. The main shortcoming of the strategy is that population diversity in *Ae* may be rapidly lost. Another one is replacing a randomly selected individual in *Ae* by $\mathbf{X}^t_{best}$. In this condition, the most outstanding individual in *Ae* may be replaced by $\mathbf{X}^t_{best}$, which is harmful for efficient utilizing those elites. Thus, in this study, a tournament strategy is adopted to update *Ae*. Specially, an inferior individual of two randomly selected elites from *Ae* is replaced by $\mathbf{X}^t_{best}$. According to above-mentioned discussions, the update process of *Ae* can be detailed as Algorithm 1.

```
Algorithm 1  Update_Ae ( ).
Input: $Ae$, $t$, $\mathbf{X}^t_{gbest}$.
If $Ae == \emptyset$ /*Initialize $Ae$ */
    $Ae = \{\mathbf{X}_{lbest_1}, \mathbf{X}_{lbest_2}, \cdots, \mathbf{X}_{lbest_{gn}}\}$;
Else /*Add $\mathbf{X}^t_{gbest}$ into $Ae$ */
    Randomly select $\mathbf{X}_{lbest_i}$ and $\mathbf{X}_{lbest_j}$ $(i \neq j)$ from $Ae$;
    If $f(\mathbf{X}_{lbest_i})$¡$f(\mathbf{X}_{lbest_j})$
        $\mathbf{X}_{lbest_j} = \mathbf{X}^t_{gbest}$;
    Else
        $\mathbf{X}_{lbest_i} = \mathbf{X}^t_{gbest}$;
    End If
End If
Output: $Ae$.
```

3.4. Adaptation of population size

Although there are relatively fewer researches on population size than the works on the adjustment of *F* and *CR*, many studies manifest that various adaptations of population size yield positive performance to some extent [14,50–52].

In MRDE, the adaptation of population size includes two processes, including decrease size process and increase size process. The details of the two processes are presented as follows.

3.4.1. Decrease size process

When carrying out the decrease size process, there are two challenges need to be dealt with. The first challenge is when to perform the process. In MRDE, when successive generations of improvement (Suc_{imp}) of $\mathbf{X}^t_{gbest}$ is greater than a predefined threshold Max_{imp}, we regard current fitness landscape is easy to be optimized. Thus, deleting a few individuals from current population is helpful for saving computational resources. The second challenge needs to be solved is which individuals should be removed from the current population. In MRDE, for simplicity, a few worst individuals are deleted from the population. Considering that all groups in MRDE have the same size *gs*, the number of deleted individuals is a multiple of *gs*.

被删除的个体数是gs的倍数？

3.4.2. Increase size process

When conducting the increase size process, there are also a few issues must be taken into account. The first one is when to perform the increase size process. Similar to the decrease size process, the number of successive stagnation generations (Suc_{stag}) of $\mathbf{X}^t_{gbest}$ is regarded as a criteria for increasing population size. Concretely, while Suc_{stag} is larger than a predefined threshold Max_{stag}, we think that individuals in current population cannot effectively search for a more optimal solution. To deal with the problem, a few new individuals with outstanding performance need to be added into the current population. However, it is undesirable that adding new individuals into population without limit. Thus, if the population size has reached the upper limit, many inferiors in the current population should be replaced by the new individuals. The second issue is how to generate the new individuals that will be added into the current population. Note that the number of added individuals is also a multiple of *gs*.

In MRDE, all elites in *Ae* are used to breed the new individuals based on GA operators, the details of which are presented in Algorithm 2.

```
Algorithm 2  Breed_NI ( ).
Input: $Ae$, $L_j$, $U_j$, $p_c$ and $p_m$.
For $j = 1$ to $D$ Do
    Randomly select $\mathbf{X}_{lbest_{i1}}$ and $\mathbf{X}_{lbest_{i2}}$ from $Ae$;
    If $rand < p_c$ /* crossover */
        $\hat{x}_j = x_{lb_{i1},j}$; /*$x_{lb_{i1},j}$ is the $j$th dimension of $\mathbf{X}_{lbest_{i1}}$ */
    Else
        $\hat{x}_j = x_{lb_{i2},j}$; /*$x_{lb_{i2},j}$ is the $j$th dimension of $\mathbf{X}_{lbest_{i2}}$ */
    End If
    If $rand < p_m$ /* mutation */
        $\hat{x}_j = rand(L_j, U_j)$;
    End If
End For
Output: $\mathbf{NI} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_D]$. /* $\mathbf{NI}$ is the new generated
individual */
```

Based on the aforementioned descriptions, the pseudocode of adaptive population size is presented in Algorithm 3. Note that, the numbers of deleted and added individuals in MRDE are $gn \times 2$ and *gn*, respectively.

```
Algorithm 3  Adaptive_PS ( ).
Input: $Suc_{imp}$, $Max_{imp}$, $Min_{ps}$, $Suc_{stag}$, $Max_{stag}$, $Max_{ps}$, $\mathbf{X}_i$
($1 \le i \le gn \times gs$).
If $Suc_{imp} \ge Max_{imp}$ & $gn \times gs > Min_{ps}$
    $Suc_{imp} = 0$;
    Delete the $gs \times 2$ worst individuals from the current
     population;
    $gn = gn - 2$;
    Randomly regroup the population into $gn$ groups;
Else
    If $Suc_{stag} \ge Max_{stag}$
        Generate $gs$ individuals $\mathbf{NI}_i$ ($1 \le i \le gs$) by
         Algorithm 2;
        If $gn \times gs < Max_{ps}$
            Add $\mathbf{NI}_i$ into the current population; $gn = gn + 1$;
        Else
            Replace the worst $gs$ individuals in population by $\mathbf{NI}_i$
            ($1 \le i \le gs$);
        End If
        $Suc_{stag} = 0$;
        Randomly regroup the current population into $gn$
         groups;
    End If
End If
Output: $\mathbf{X}_i$ ($1 \le i \le gn * gs$).
```

3.5. Pseudocode implementation of MRDE

Based on the aforementioned discussions of the new introduced strategies, the pseudocode of MRDE can be detailed as Algorithm 4.

Algorithm 4 MRDE ( ).

```
Input: $Max_{imp}$, $Max_{stag}$, $Min_{ps}$, $Max_{ps}$, $gn$, $gs$, $L_j$, $U_j$ $(1 \le j \le D)$;
Initialization: $t = 1$, $Suc_{imp} = 0$, $Suc_{stag} = 0$, $Ae = \emptyset$;
Randomly generate a population $\{\mathbf{X}_1, \mathbf{X}_2, \cdots \mathbf{X}_{ps}\}$ $(ps = gs \times gn)$;
Randomly divided the current population in $gn$ groups with the same size $gs$;
Evaluate individuals in each group;
Record $\mathbf{X}^t_{lbest_k}$ $(1 \le k \le gn)$ and $\mathbf{X}^t_{gbest}$;
While not meet stop conditions
    Perform Update_Ae ( ); /* Algorithm 1 */
    For each group $k$ $(1 \le k \le gn)$ Do
        Assign $Setting$ 1 for the elite in group $k$;
        Assign $Setting$ 2 for the inferior in group $k$;
        Assign $Setting$ 3 for the mediocrities in group $k$;
        Utilize Eqs. (1)–(3) to generate offspring for each individual in group $k$;
    End For
    Evaluate individuals in each group; $t = t+1$
    If $\mathbf{X}^t_{gbest}$ is improved
        $Suc_{imp} = Suc_{imp}+1$; $Suc_{stag} = 0$;
    Else
        $Suc_{stag} = Suc_{stag}+1$; $Suc_{imp} = 0$;
    End If
    Adaptive_Chang_PS ( ); /* Algorithm 3 */
End While
Output: $\mathbf{X}^t_{gbest}$
```

4. Experimental results and discussions

4.1. Benchmark functions and peer algorithms

In this work, CEC2017 test suite is chosen to verify characteristics and performance of MRDE. In the test suite, there are 30 test functions which can be categorized into unimodal functions (f_1-f_3), simple multimoal functions (f_4-f_{10}), hybrid functions (f_{11}-f_{20}), and composition functions (f_{21}-f_{30}). Detailed information of the functions can refer to the literature [53]. In experiments carried out in this section, three dimension cases, i.e., $D = 10$, 30, and 50, are used to provide a more comprehensive comparison among all the peer algorithms. The allowed number of the maximum function evaluations (MaxFEs) is set as 10 000 × D.

To testify the performance of MRDE, ten state-of-art DE variants are selected in this research, including JADE [11], SaDE [19], CoDE [20], DCMA-EA [47], CoBiDE [54], TSDE [36], MPEDE [45], EFADE [46], EPSDE [22], and LSHADE_cnEpSin [55]. The reason why we select these ten DE variants as peer algorithms are explained as follows. Firstly, JADE is an efficient algorithm and widely chosen as a baseline algorithm. Secondly, multiple mutation strategies are adopted in SaDE, CoDE, CoBiDE, and TSDE, which is similar to MRDE. Furthermore, a parameter candidate pool is also applied in CoDE and TSDE. Thus, it is meaningful to compare MRDE with them. Thirdly, the multipopulation mechanism is applied in MPEDE and EPSDE aiming to offer favorable exploration ability. Moreover, a triangular mutation operator is proposed in EFADE to trade-off the exploration ability and exploitation ability, while DCMA-EA is hybrid algorithm based on DE and CMAES. Hence, comparisons among MRDE, MPEDE, EFADE, and DCMA-EA can comprehensively testify the efficiency of MRDE. Lastly, LSHADE_cnEpSin, as one of the best performing algorithms from CEC 2017 special session, is selected to verify the comprehensive performance of MRDE. To obtain statistical results, each algorithm is carried out 51 independent runs on each benchmark function.

Note that all the mutation strategies and parameter settings in the peer DE variants are the same of those given in the original references. Settings of the new introduced parameters in MRDE and corresponding discussions are presented as follow:

1) $gs = 3$, $gn = 20$ for 10D cases, $gn = 25$ for 30D and 50D cases;
2) $Max_{imp} = 2$, $Max_{stag} = 3$; and
3) $Min_{ps} = ps^1/3$, $Max_{ps} = ps^1$.

Note that ps^1 is the population size in the initialization stage (i.e., $t = 1$).

It is worth to note that the above mentioned values are determined by trial-and-error. Although we have a rudimentary understanding of how the different parameters affect the performance of MRDE, the configurations of the parameters may not be the optimal settings because there may be dependency between the different parameters, which causes the combination of them no long to be an optimal setting for MRDE.

4.2. Comparison on solution accuracy

In this section, experiments are conducted to compare the performance of each algorithm in terms of the mean values (Mean) and standard deviations (S.D.) of the solutions. The corresponding results on $D = 10$, $D = 30$, and $D = 50$ are provided in Tables 1–3, respectively, where the best result on each function is shown in bold. Moreover, for each test function, rank values of all peer algorithms, in terms of mean values, are also presented in these tables. For each algorithm, an average value of the rank values (Avg. (R)) on all the test functions is also included in the tables aiming to demonstrate an overall performance for the algorithm.

1) Unimodal functions (f_1-f_3): On the three dimension cases, majority of the 11 algorithms yield very high accurate solutions. In particular, for 10D case, 7 out of the 11 algorithms find out the global optimal solutions on the three unimoal functions on all independent runs. However, along with the increase of dimensions, performance of the majority of the peer algorithms is rapidly degraded except MRDE, DCMA-EA, EFADE, and LSHADE_cnEpSin. The results provide evidence to that elites paying more attention on exploitation in MRDE is beneficial for unimodal functions.
2) Simple multimodal functions (f_4-f_{10}): For the 7 simple multimodal functions, MRDE, SaDE, TSDE, and LSHADE_cnEpSin attain more favorable performance than other peer algorithms on 10D case,

Table 1
Comparison results of solution accuracy on CEC2017 test suite (10D).

		MRDE	JADE	SaDE	CoDE	DCMA-EA	CoBiDE	TSDE	MPEDE	EFADE	EPSDE	LSHADE -cnEpSin
f_1	Mean(R)	**0.00E+00**(1)	**0.00E+00**(1)	**0.00E+00**(1)	8.64E-15(10)	**0.00E+00**(1)	**0.00E+00**(1)	**0.00E+00**(1)	6.53E-11(11)	**0.00E+00**(1)	2.79E-15(9)	**0.00E+00**(1)
	S.D.	0.00E+00	0.00E+00	0.00E+00	7.01E-15	0.00E+00	0.00E+00	0.00E+00	1.32E-10	0.00E+00	7.53E-15	0.00E+00
f_2	Mean(R)	**0.00E+00**(1)	**0.00E+00**(1)	**0.00E+00**(1)	4.31E-01(11)	**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)
	S.D.	0.00E+00	0.00E+00	0.00E+00	7.81E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_3	Mean(R)	**0.00E+00**(1)	1.11E-15(8)	**0.00E+00**(1)	2.67E-14(10)	**0.00E+00**(1)	**0.00E+00**(1)	**0.00E+00**(1)	6.69E-14(11)	**0.00E+00**(1)	1.11E-15(8)	**0.00E+00**(1)
	S.D.	0.00E+00	7.96E-15	0.00E+00	2.87E-14	0.00E+00	0.00E+00	0.00E+00	2.87E-13	0.00E+00	7.96E-15	0.00E+00
f_4	Mean(R)	**0.00E+00**(1)	1.11E-15(7)	1.37E+00(11)	4.35E-14(9)	**0.00E+00**(1)	**0.00E+00**(1)	**0.00E+00**(1)	1.49E-08(10)	**0.00E+00**(1)	6.69E-15(8)	**0.00E+00**(1)
	S.D.	0.00E+00	7.96E-15	8.04E-01	3.13E-14	0.00E+00	0.00E+00	0.00E+00	6.28E-08	0.00E+00	1.85E-14	0.00E+00
f_5	Mean(R)	3.24E+00(3)	3.62E+00(4)	3.02E+00(2)	6.03E+00(7)	5.97E+01(11)	6.26E+00(9)	4.33E+00(6)	6.24E+00(8)	6.49E+00(10)	3.87E+00(5)	1.76E+00(1)
	S.D.	1.39E+00	1.35E+00	1.26E+00	2.00E+00	9.02E+01	2.42E+00	2.40E+00	1.67E+00	1.44E+00	1.09E+00	7.05E-01
f_6	Mean(R)	2.23E-15(6)	4.46E-15(7)	**0.00E+00**(1)	2.90E-14(8)	7.65E+01(11)	**0.00E+00**(1)	**0.00E+00**(1)	2.37E-05(10)	4.68E-14(9)	**0.00E+00**(1)	**0.00E+00**(1)
	S.D.	1.59E-14	2.23E-14	0.00E+00	5.00E-14	3.22E+01	0.00E+00	0.00E+00	5.98E-06	5.65E-14	0.00E+00	0.00E+00
f_7	Mean(R)	1.28E+01(2)	1.35E+01(4)	1.35E+01(3)	1.51E+01(7)	1.55E+02(11)	1.69E+01(8)	1.44E+01(6)	1.74E+01(9)	1.91E+01(10)	1.42E+01(5)	**1.20E+01**(1)
	S.D.	2.96E+00	1.38E+00	1.70E+00	2.77E+00	3.25E+02	3.43E+00	1.79E+00	1.79E+00	1.71E+00	1.25E+00	5.63E-01
f_8	Mean(R)	3.59E+00(3)	3.89E+00(4)	3.17E+00(2)	6.15E+00(7)	8.41E+01(11)	6.26E+00(8)	4.47E+00(6)	6.34E+00(9)	7.68E+00(10)	4.31E+00(5)	**1.99E+00**(1)
	S.D.	1.27E+00	1.28E+00	1.39E+00	2.41E+00	8.01E+01	2.64E+00	2.01E+00	1.61E+00	1.99E+00	1.22E+00	6.88E-01
f_9	Mean(R)	**0.00E+00**(1)	**0.00E+00**(1)	**0.00E+00**(1)	3.34E-14(9)	2.34E+03(11)	**0.00E+00**(1)	**0.00E+00**(1)	7.13E-14(10)	**0.00E+00**(1)	**0.00E+00**(1)	**0.00E+00**(1)
	S.D.	0.00E+00	0.00E+00	0.00E+00	5.70E-14	1.85E+03	0.00E+00	0.00E+00	1.07E-13	0.00E+00	0.00E+00	0.00E+00
f_{10}	Mean(R)	9.71E+01(2)	1.38E+02(4)	2.03E+02(5)	2.29E+02(6)	1.25E+03(11)	2.10E+02(6)	1.12E+02(3)	2.92E+02(9)	2.76E+02(8)	3.03E+02(10)	**2.99E+01**(1)
	S.D.	9.97E+01	1.14E+02	1.23E+02	1.38E+02	5.06E+02	1.33E+02	9.26E+01	9.24E+01	8.44E+01	1.01E+02	4.64E+01
f_{11}	Mean(R)	3.90E-02(2)	1.76E+00(8)	5.85E-01(4)	8.14E-01(5)	2.92E+00(11)	9.56E-01(6)	2.15E-01(3)	2.37E+00(9)	1.40E+00(7)	2.47E+00(10)	**0.00E+00**(1)
	S.D.	1.95E-01	1.32E+00	6.94E-01	9.03E-01	9.30E-01	9.33E-01	4.59E-01	5.95E-01	1.05E+00	1.02E+00	0.00E+00
f_{12}	Mean(R)	**3.23E-01**(1)	3.58E+02(11)	1.63E+02(9)	2.59E+01(6)	5.16E+01(7)	1.67E+01(5)	4.33E+00(3)	1.21E+01(4)	1.62E+00(2)	2.20E+02(10)	1.03E+02(8)
	S.D.	1.72E-01	1.98E+02	1.29E+02	4.90E+01	3.32E+01	4.15E+01	1.70E+01	1.07E+01	3.54E+00	3.22E+02	7.00E+01
f_{13}	Mean(R)	**1.14E+00**(1)	5.31E+00(8)	2.91E+00(3)	4.14E+00(7)	8.18E+00(11)	3.38E+00(4)	2.15E+00(2)	5.33E+00(9)	3.60E+00(5)	5.88E+00(10)	3.61E+00(6)
	S.D.	1.89E+00	2.42E+00	2.98E+00	2.92E+00	1.12E+00	2.42E+00	2.12E+00	2.01E+00	2.89E+00	2.42E+00	2.40E+00
f_{14}	Mean(R)	**1.95E-02**(1)	4.65E+00(8)	2.36E-01(4)	9.17E-01(7)	2.20E+01(11)	7.61E-01(6)	7.80E-02(3)	4.92E+00(9)	3.85E-01(5)	1.99E+01(10)	5.85E-02(2)
	S.D.	1.39E-01	8.44E+00	4.25E-01	7.13E-01	9.59E-01	8.10E-01	2.70E-01	1.33E+00	4.52E-01	1.34E+00	2.36E-01
f_{15}	Mean(R)	**3.02E-02**(1)	2.89E-01(7)	6.76E-01(9)	2.85E-01(6)	1.44E+00(11)	2.44E-01(5)	4.46E-02(2)	7.32E-01(10)	1.26E-01(3)	6.08E-01(8)	2.16E-01(4)
	S.D.	8.21E-02	2.69E-01	7.25E-01	4.53E-01	7.75E-01	4.33E-01	1.23E-01	2.24E-01	3.11E-01	5.34E-01	2.16E-01
f_{16}	Mean(R)	2.34E-01	6.12E+00(10)	4.86E-01(3)	5.19E+00(8)	1.48E+02(11)	5.59E-01(4)	**2.27E-01**(1)	2.60E+00(7)	8.59E-01(6)	5.80E+00(9)	6.35E-01(5)
	S.D.	2.04E-01(2)	2.38E+01	2.42E-01	2.32E+01	9 1.92E+02	1.63E+00	2.20E-01	8.45E-01	3.01E-01	4.69E+00	3.07E-01
f_{17}	Mean(R)	8.67E-01(5)	8.84E-01(6)	5.15E-01(4)	4.34E+00(8)	9.20E+01(11)	3.09E-01(2)	4.21E-01(3)	8.06E+00(9)	1.59E+00(7)	1.75E+01(10)	**2.45E-01**(1)
	S.D.	6.07E-01	3.91E+00	5.35E-01	1.86E+01	1.50E+02	4.09E-01	4.55E-01	3.17E+00	8.37E-01	5.30E+00	2.27E-01
f_{18}	Mean(R)	6.34E-02(2)	9.73E+00(9)	9.58E-01(6)	1.58E-01(3)	2.03E+01(11)	2.01E-01(4)	**4.10E-02**(1)	3.19E+00(8)	2.45E-01(5)	1.88E+01(10)	2.32E+00(7)
	S.D.	1.37E-01	1.01E+01	1.16E+00	2.96E-01	4.59E-01	3.40E-01	9.72E-02	1.22E+00	3.33E-01	4.67E+00	6.06E+00
f_{19}	Mean(R)	1.87E-03(2)	2.34E-01(8)	**1.69E-03**(1)	1.42E-02(5)	1.84E+00(11)	3.88E-02(7)	1.22E-02(3)	5.66E-01(10)	1.40E-02(4)	3.32E-01(9)	1.80E-02(6)
	S.D.	5.30E-03	7.15E-01	4.91E-03	2.20E-02	2.41E-01	2.12E-01	1.01E-02	1.75E-01	1.22E-02	1.23E-01	3.32E-02
f_{20}	Mean(R)	7.35E-02(5)	4.66E-01(7)	1.22E-02(3)	8.19E-02(6)	4.03E+02(11)	5.01E-02(4)	**0.00E+00**(1)	9.75E-01(9)	**0.00E+00**(1)	1.63E+01(10)	7.48E-01(8)
	S.D.	1.48E-01	2.79E+00	6.12E-02	2.24E-01	2.00E+02	1.64E-01	0.00E+00	5.14E-01	0.00E+00	6.30E+00	2.85E+00
f_{21}	Mean(R)	**1.06E+02**(1)	1.73E+02(10)	1.44E+02(4)	1.70E+02(8)	2.63E+02(11)	1.69E+02(7)	1.59E+02(6)	1.08E+02(2)	1.26E+02(3)	1.71E+02(9)	1.53E+02(5)
	S.D.	2.51E+01	4.89E+01	5.26E+01	5.26E+01	9.21E+01	5.35E+01	5.37E+01	3.55E+01	4.68E+01	4.97E+01	5.21E+01
f_{22}	Mean(R)	**7.91E+01**(1)	9.64E+01(7)	1.00E+02(10)	8.90E+01(4)	1.35E+03(11)	9.30E+01(5)	8.81E+01(3)	9.61E+01(6)	8.06E+01(2)	9.87E+01(8)	1.00E+02(9)
	S.D.	4.09E+01	1.85E+01	1.40E-01	3.19E+01	8.45E+02	2.58E+01	3.15E+01	1.96E+01	3.89E+01	9.21E+00	6.82E-02
f_{23}	Mean(R)	3.05E+02(4)	3.06E+02(5)	3.05E+02(3)	3.08E+02(8)	5.62E+02(11)	3.07E+02(7)	**3.00E+02**(1)	3.06E+02(6)	3.08E+02(9)	3.11E+02(10)	3.02E+02(2)
	S.D.	1.93E+00	2.31E+00	2.56E+00	2.64E+00	3.63E+02	3.09E+00	4.29E+01	2.20E+00	2.00E+00	2.21E+00	1.51E+00
f_{24}	Mean(R)	**2.31E+02**(1)	2.97E+02(7)	2.50E+02(4)	3.10E+02(8)	2.86E+02(6)	3.19E+02(10)	2.85E+02(5)	2.47E+02(3)	2.39E+02(2)	3.35E+02(11)	3.11E+02(9)
	S.D.	1.20E+02	8.56E+01	1.12E+02	7.75E+01	9.82E+01	5.87E+01	9.80E+01	1.14E+02	1.17E+02	2.51E+00	6.22E+01
f_{25}	Mean(R)	4.01E+02(2)	4.20E+02(8)	4.20E+02(9)	4.14E+02(6)	4.32E+02(11)	4.07E+02(4)	4.06E+02(3)	4.10E+02(5)	**4.00E+02**(1)	4.17E+02(7)	4.24E+02(10)
	S.D.	1.10E+01	2.32E+01	2.28E+01	2.21E+01	2.03E+01	1.84E+01	1.78E+01	2.02E+01	8.92E+00	2.32E+01	2.27E+01
f_{26}	Mean(R)	**3.00E+02**(1)	3.31E+02(10)	**3.00E+02**(1)	**3.00E+02**(1)	4.62E+02(11)	**3.00E+02**(1)	**3.00E+02**(1)	**3.00E+02**(8)	**3.00E+02**(1)	3.28E+02(9)	**3.00E+02**(1)
	S.D.	0.00E+00	1.80E+02	0.00E+00	0.00E+00	2.82E+02	0.00E+00	0.00E+00	1.84E-08	0.00E+00	6.07E+01	0.00E+00
f_{27}	Mean(R)	**3.89E+02**(1)	3.91E+02(8)	3.94E+02(9)	3.91E+02(7)	4.22E+02(11)	3.90E+02(6)	3.89E+02(3)	3.89E+02(5)	3.89E+02(4)	4.04E+02(10)	3.89E+02(2)
	S.D.	1.24E+00	2.74E+00	1.71E+00	2.15E+00	4.46E+01	1.87E+00	1.38E+00	4.62E-01	1.96E+00	4.90E+01	9.61E-01
f_{28}	Mean(R)	**2.94E+02**(1)	4.58E+02(9)	3.28E+02(7)	3.28E+02(6)	4.74E+02(10)	3.17E+02(4)	3.00E+02(2)	3.18E+02(5)	3.00E+02(3)	4.76E+02(11)	3.35E+02(8)
	S.D.	4.20E+01	1.48E+02	8.70E+01	8.70E+01	7.00E+01	6.74E+01	5.84E+01	7.41E+01	0.00E+00	7.68E+00	9.19E+01
f_{29}	Mean(R)	2.35E+02(4)	2.51E+02(9)	2.43E+02(6)	2.37E+02(5)	2.72E+02(11)	2.34E+02(3)	2.32E+02(2)	2.50E+02(8)	2.52E+02(10)	2.46E+02(7)	**2.29E+02**(1)
	S.D.	4.75E+00	1.15E+01	8.09E+00	1.00E+01	8.21E+01	6.69E+00	3.51E+00	5.55E+00	5.69E+00	9.35E+00	2.28E+00
f_{30}	Mean(R)	3.95E+02(3)	1.58E+05(11)	7.02E+02(8)	1.65E+04(10)	**2.02E+02**(1)	4.24E+02(7)	4.12E+02(6)	3.96E+02(4)	3.97E+02(5)	2.13E+02(2)	3.72E+03(9)
	S.D.	7.27E-02	3.68E+05	3.45E+02	1.14E+05	5.44E-01	5.34E+01	3.37E+01	2.52E+00	7.46E+00	2.07E+01	2.36E+04
Avg.	(R)	2.07	6.90	4.50	6.97	9.00	4.60	2.70	7.47	4.57	7.77	3.80

‘(R)’ is an order number of Mean values (the lower the better). ‘Avg. (R)’ is the average rank value of a peer algorithm.

Table 2
Comparison results of solution accuracy on CEC2017 test suite (30D).

		MRDE	JADE	SaDE	CoDE	DCMA-EA	CoBiDE	TSDE	MPEDE	EFADE	EPSDE	LSHADE -cnEpSin
f_1	Mean(*R*)	1.23E-14(5)	1.39E-14(6)	1.62E+03(11)	1.45E-14(7)	**0.00E+00**(1)	7.88E-13(10)	3.62E-15(4)	1.39E-15(3)	1.50E-14(8)	2.20E-13(9)	2.79E-16(2)
	S.D.	4.94E-15	1.99E-15	1.49E+03	1.12E-14	0.00E+00	4.65E-12	9.36E-15	4.27E-15	8.24E-15	6.03E-13	1.99E-15
f_2	Mean(*R*)	**0.00E+00**(1)	1.82E+02(10)	**0.00E+00**(1)	1.02E+00(8)	**0.00E+00**(1)	**0.00E+00**(1)	**0.00E+00**(1)	4.69E+00(9)	**0.00E+00**(1)	6.92E+16(11)	**0.00E+00**(1)
	S.D.	0.00E+00	7.87E+02	0.00E+00	1.19E+00	0.00E+00	0.00E+00	0.00E+00	1.60E+01	0.00E+00	3.22E+17	0.00E+00
f_3	Mean(*R*)	3.50E-13(6)	9.28E+03(11)	7.52E-03(8)	3.15E-13(5)	8.96E+03(10)	**0.00E+00**(1)	**0.00E+00**(1)	6.30E-11(7)	2.22E-13(4)	5.53E+03(9)	1.11E-15(3)
	S.D.	8.54E-13	1.62E+04	2.00E-02	8.99E-13	3.64E+04	0.00E+00	0.00E+00	4.49E-10	2.63E-13	2.19E+04	7.96E-15
f_4	Mean(*R*)	4.26E+01(7)	5.50E+01(11)	3.66E+01(5)	3.11E+01(4)	**3.13E-01**(1)	2.22E+01(3)	5.04E+01(10)	4.91E+01(9)	3.74E+01(6)	1.25E+00(2)	4.31E+01(8)
	S.D.	2.73E+01	1.61E+01	3.51E+01	2.96E+01	1.08E+00	2.67E+01	2.25E+01	2.27E+01	2.91E+01	1.87E+00	3.06E+00
f_5	Mean(*R*)	3.17E+01(4)	2.60E+01(2)	3.71E+01(5)	3.92E+01(7)	4.10E+02(11)	4.73E+01(10)	3.91E+01(6)	2.93E+01(3)	4.20E+01(8)	4.44E+01(9)	**1.23E+01**(1)
	S.D.	9.11E+00	4.76E+00	9.20E+00	9.19E+00	3.15E+02	1.41E+01	1.14E+01	7.68E+00	1.32E+01	8.19E+00	1.83E+00
f_6	Mean(*R*)	2.68E-09(6)	1.67E-13(4)	3.38E-05(10)	1.67E-13(4)	6.59E+01(11)	2.82E-07(9)	**1.11E-13**(1)	2.69E-09(7)	1.25E-13(3)	1.14E-13(2)	2.89E-08(8)
	S.D.	1.92E-08	8.91E-14	2.42E-04	5.73E-14	3.52E+01	2.01E-06	1.59E-14	1.92E-08	3.41E-14	0.00E+00	1.12E-07
f_7	Mean(*R*)	6.40E+01(5)	5.47E+01(3)	6.40E+01(4)	6.74E+01(6)	2.01E+02(11)	8.04E+01(9)	6.88E+01(7)	5.46E+01(2)	9.51E+01(10)	7.71E+01(8)	**4.51E+01**(1)
	S.D.	8.71E+00	3.54E+00	9.98E+00	1.02E+01	3.17E+02	1.43E+01	1.03E+01	7.37E+00	2.54E+01	6.56E+00	1.73E+00
f_8	Mean(*R*)	3.15E+01(4)	2.74E+01(2)	3.86E+01(5)	3.94E+01(6)	3.52E+02(11)	4.85E+01(10)	3.95E+01(7)	2.97E+01(3)	4.54E+01(9)	4.51E+01(8)	**1.33E+01**(1)
	S.D.	9.14E+00	3.58E+00	9.32E+00	1.20E+01	2.16E+02	1.19E+01	1.27E+01	7.83E+00	1.36E+01	7.38E+00	1.89E+00
f_9	Mean(*R*)	9.14E-14(3)	1.96E-02(4)	2.12E+00(10)	9.80E-02(8)	1.21E+04(11)	1.21E-01(9)	4.45E-02(7)	1.96E-02(4)	2.23E-14(2)	4.25E-02(6)	**0.00E+00**(1)
	S.D.	4.56E-14	8.96E-02	3.90E+00	3.50E-01	5.12E+03	4.44E-01	1.36E-01	8.96E-02	4.56E-14	1.31E-01	0.00E+00
f_{10}	Mean(*R*)	2.37E+03(5)	1.92E+03(2)	3.43E+03(8)	1.96E+03(4)	4.76E+03(11)	2.41E+03(6)	1.94E+03(3)	2.65E+03(7)	3.57E+03(9)	3.72E+03(10)	**1.50E+03**(1)
	S.D.	7.86E+02	2.20E+02	6.78E+02	5.30E+02	6.90E+02	5.42E+02	4.35E+02	4.54E+02	7.36E+02	3.41E+02	1.96E+02
f_{11}	Mean(*R*)	1.38E+01(2)	3.13E+01(9)	7.52E+01(10)	2.23E+01(5)	1.06E+02(11)	2.51E+01(6)	1.70E+01(4)	2.71E+01(7)	**1.30E+01**(1)	2.86E+01(8)	1.39E+01(3)
	S.D.	1.44E+01	2.45E+01	3.19E+01	2.06E+01	1.93E+02	1.68E+01	1.66E+01	1.40E+01	4.99E+00	2.02E+01	2.02E+01
f_{12}	Mean(*R*)	3.59E+03(5)	1.16E+03(3)	1.67E+04(10)	8.30E+03(8)	1.57E+03(4)	1.05E+04(9)	8.27E+03(7)	9.31E+02(2)	7.33E+03(6)	4.11E+04(11)	**3.75E+02**(1)
	S.D.	4.98E+03	4.33E+02	7.46E+03	5.87E+03	4.66E+02	9.91E+03	9.72E+03	4.49E+02	7.21E+03	4.11E+04	2.25E+02
f_{13}	Mean(*R*)	1.89E+01(2)	1.92E+02(8)	7.35E+03(10)	5.66E+01(6)	4.87E+06(11)	1.65E+02(7)	2.88E+01(4)	2.12E+01(3)	3.29E+01(5)	3.10E+03(9)	**1.87E+01**(1)
	S.D.	7.81E+00	1.09E+03	6.17E+03	9.96E+01	4.94E+06	2.84E+02	1.28E+01	8.14E+00	2.70E+01	9.14E+03	6.30E+00
f_{14}	Mean(*R*)	**9.19E+00**(1)	4.58E+03(10)	7.36E+01(9)	1.32E+01(4)	1.61E+05(11)	2.51E+01(7)	1.23E+01(3)	1.64E+01(5)	1.11E+01(2)	6.24E+01(8)	2.12E+01(6)
	S.D.	5.44E+00	8.11E+03	2.73E+01	8.85E+00	3.69E+05	1.19E+01	7.91E+00	1.08E+01	4.26E+00	3.51E+01	4.26E+00
f_{15}	Mean(*R*)	7.30E+00(2)	1.13E+03(10)	1.22E+02(8)	1.13E+01(6)	6.68E+05(11)	1.75E+01(7)	8.58E+00(5)	7.96E+00(4)	7.91E+00(3)	1.32E+02(9)	**3.22E+00**(1)
	S.D.	3.00E+00	2.97E+03	1.05E+02	5.09E+00	5.11E+05	1.05E+01	3.81E+00	3.89E+00	3.22E+00	2.30E+02	1.38E+00
f_{16}	Mean(*R*)	3.53E+02(2)	4.30E+02(4)	4.35E+02(5)	4.84E+02(7)	1.00E+03(11)	5.12E+02(9)	4.92E+02(8)	3.72E+02(3)	4.47E+02(6)	5.94E+02(10)	**2.68E+01**(1)
	S.D.	1.60E+02	1.25E+02	1.38E+02	2.26E+02	2.75E+02	1.84E+02	1.92E+02	1.97E+02	1.76E+02	1.29E+02	3.61E+01
f_{17}	Mean(*R*)	3.99E+01(2)	7.36E+01(8)	4.65E+01(3)	5.19E+01(4)	5.78E+02(11)	7.96E+01(9)	7.18E+01(7)	5.69E+01(5)	6.17E+01(6)	2.00E+02(10)	**2.66E+01**(1)
	S.D.	1.20E+01	2.76E+01	2.60E+01	5.14E+01	2.11E+02	7.92E+01	7.60E+01	3.49E+01	3.89E+01	7.43E+01	7.04E+00
f_{18}	Mean(*R*)	**1.91E+01**(1)	1.93E+04(11)	5.79E+03(10)	1.38E+02(7)	3.03E+01(4)	4.31E+02(8)	8.45E+01(5)	2.31E+01(3)	9.96E+01(6)	1.07E+03(9)	2.09E+01(2)
	S.D.	9.43E+00	5.64E+04	5.69E+03	1.51E+02	1.91E+01	5.91E+02	6.06E+01	6.88E+00	1.11E+02	1.15E+03	5.74E-01
f_{19}	Mean(*R*)	**5.14E+00**(1)	1.95E+03(10)	4.51E+01(9)	5.54E+00(2)	5.06E+06(11)	8.40E+00(7)	5.92E+00(3)	6.77E+00(6)	6.35E+00(5)	2.74E+01(8)	5.93E+00(4)
	S.D.	1.58E+00	4.44E+03	3.21E+01	1.73E+00	6.69E+06	3.29E+00	1.83E+00	2.19E+00	2.27E+00	2.23E+01	1.28E+00
f_{20}	Mean(*R*)	4.30E+01(2)	9.76E+01(5)	6.64E+01(4)	1.14E+02(8)	7.64E+02(11)	1.11E+02(7)	1.10E+02(6)	6.61E+01(3)	1.16E+02(9)	1.63E+02(10)	**3.32E+01**(1)
	S.D.	5.88E+01	4.96E+01	6.14E+01	9.15E+01	2.97E+02	8.86E+01	8.99E+01	5.18E+01	6.08E+01	7.16E+01	1.04E+01
f_{21}	Mean(*R*)	2.36E+02(5)	2.27E+02(2)	2.36E+02(4)	2.38E+02(6)	3.66E+02(11)	2.53E+02(10)	2.40E+02(7)	2.29E+02(3)	2.42E+02(8)	2.46E+02(9)	**2.13E+02**(1)
	S.D.	1.04E+01	4.17E+00	8.34E+00	9.27E+00	1.61E+02	1.47E+01	8.80E+00	8.49E+00	1.10E+01	7.90E+00	2.39E+00
f_{22}	Mean(*R*)	1.00E+02(5)	1.01E+02(8)	1.01E+02(7)	1.00E+02(4)	5.05E+03(11)	1.68E+02(9)	1.00E+02(6)	1.00E+02(1)	1.00E+02(3)	3.31E+03(10)	**1.00E+02**(1)
	S.D.	3.26E-13	4.48E+00	1.10E+00	5.01E-13	1.48E+03	4.85E+02	0.00E+00	0.00E+00	4.49E-13	1.74E+03	0.00E+00
f_{23}	Mean(*R*)	3.84E+02(5)	3.73E+02(2)	3.83E+02(4)	3.85E+02(6)	8.54E+02(11)	3.97E+02(9)	3.86E+02(7)	3.79E+02(3)	3.92E+02(8)	3.98E+02(10)	**3.55E+02**(1)
	S.D.	1.26E+01	5.50E+00	1.17E+01	1.08E+01	8.73E+02	1.18E+01	9.14E+00	9.49E+00	1.21E+01	1.04E+01	2.94E+00
f_{24}	Mean(*R*)	4.60E+02(7)	4.42E+02(2)	4.51E+02(4)	4.56E+02(5)	5.27E+02(11)	4.69E+02(9)	4.60E+02(6)	4.45E+02(3)	4.70E+02(10)	4.67E+02(8)	**4.29E+02**(1)
	S.D.	1.05E+01	4.80E+00	1.14E+01	9.81E+00	7.22E+01	1.13E+01	1.10E+01	7.50E+00	1.35E+01	7.87E+00	2.56E+00
f_{25}	Mean(*R*)	3.87E+02(5)	3.87E+02(9)	3.92E+02(11)	3.87E+02(8)	3.78E+02(2)	3.87E+02(10)	3.87E+02(7)	**3.78E+02**(1)	3.87E+02(6)	3.78E+02(3)	3.87E+02(4)
	S.D.	4.93E-02	1.32E-01	1.15E+01	1.03E-01	1.15E-01	1.73E-01	5.13E-01	6.37E-01	3.79E-02	9.97E-01	7.61E-03
f_{26}	Mean(*R*)	1.27E+03(6)	1.17E+03(3)	1.29E+03(7)	1.37E+03(8)	1.62E+03(11)	1.51E+03(10)	1.39E+03(9)	1.25E+03(5)	1.07E+03(2)	1.21E+03(4)	**9.48E+02**(1)
	S.D.	2.27E+02	6.60E+01	4.41E+02	2.17E+02	5.28E+02	1.43E+02	1.26E+02	1.06E+02	5.37E+02	1.26E+02	4.86E+01
f_{27}	Mean(*R*)	**4.92E+02**(1)	5.03E+02(8)	5.17E+02(11)	5.01E+02(6)	5.00E+02(5)	5.07E+02(10)	4.96E+02(3)	5.03E+02(7)	4.98E+02(2)	5.00E+02(4)	5.06E+02(9)
	S.D.	9.62E+00	8.17E+00	8.14E+00	7.45E+00	8.76E-05	7.65E+00	7.61E+00	6.24E+00	7.27E+00	9.37E-05	5.20E+00
f_{28}	Mean(*R*)	3.22E+02(2)	3.36E+02(8)	3.24E+02(4)	3.30E+02(5)	4.90E+02(10)	3.34E+02(7)	3.31E+02(6)	3.41E+02(9)	**3.14E+02**(1)	5.00E+02(11)	3.22E+02(3)
	S.D.	4.52E+01	5.23E+01	4.54E+01	5.04E+01	4.09E+01	5.33E+01	5.14E+01	5.61E+01	3.59E+01	2.03E+00	4.45E+01
f_{29}	Mean(*R*)	**4.07E+02**(1)	4.84E+02(9)	4.49E+02(7)	4.32E+02(2)	4.59E+02(8)	5.08E+02(11)	4.32E+02(3)	4.49E+02(6)	4.40E+02(4)	4.89E+02(10)	4.43E+02(5)
	S.D.	4.31E+01	2.77E+01	2.67E+01	5.28E+01	1.57E+02	8.99E+01	6.79E+01	2.93E+01	5.17E+01	8.73E+01	9.52E+00
f_{30}	Mean(*R*)	1.99E+03(2)	2.15E+03(8)	2.96E+03(10)	2.09E+03(5)	2.08E+05(11)	2.15E+03(7)	2.09E+03(6)	2.02E+03(4)	2.27E+03(9)	**2.22E+02**(1)	2.00E+03(3)
	S.D.	4.20E+01	1.44E+02	7.69E+02	1.01E+02	1.73E+05	1.34E+02	9.79E+01	6.77E+01	1.94E+02	9.00E+00	5.83E+01
Avg.	(*R*)	3.50	6.40	7.13	5.70	8.87	7.87	5.30	4.57	5.40	7.87	2.57

while LSHADE_cnEpSin offers the best results on 30D and 50D cases. Although MRDE cannot display the most favorable performance in the two higher dimension cases, it shows relatively favorable performance on the majority of the functions.

3) Hybrid functions (f_{11}-f_{20}): For the 10 hybrid functions, MRDE yields the best results on 10D case, in terms of mean value. Moreover, MRDE and LSHADE_cnEpSin both offer more promising results than other peer DE variants on 30D and 50D cases. Thus, we regard

Table 3
Comparison results of solution accuracy on CEC2017 test suite (50D).

		MRDE	JADE	SaDE	CoDE	DCMA-EA	CoBiDE	TSDE	MPEDE	EFADE	EPSDE	LSHADE -cnEpSin
f_1	Mean(R)	1.73E-07(6)	2.65E-14(3)	1.17E+03(10)	2.99E+03(11)	**0.00E+00**(1)	6.76E+00(9)	1.65E-04(7)	2.65E-14(3)	8.41E-01(8)	3.72E-09(5)	**0.00E+00**(1)
	S.D.	1.23E-06	8.05E-15	1.46E+03	2.92E+03	0.00E+00	4.38E+01	6.83E-04	8.05E-15	1.68E+00	5.71E-09	0.00E+00
f_2	Mean(R)	1.96E-02(4)	1.08E+10(7)	8.07E+11(9)	1.23E+33(10)	**0.00E+00**(1)	3.05E+10(8)	1.75E+06(5)	9.02E+08(6)	**0.00E+00**(1)	7.89E+45(11)	**0.00E+00**(1)
	S.D.	1.40E-01	7.70E+10	1.95E+12	6.67E+33	0.00E+00	1.35E+11	6.77E+06	5.04E+09	0.00E+00	5.49E+46	0.00E+00
f_3	Mean(R)	3.35E-07(5)	2.92E+04(10)	1.94E+02(8)	2.21E+02(9)	3.34E-14(2)	2.04E-07(4)	2.53E-08(3)	2.50E-04(7)	7.75E-05(6)	6.83E+04(11)	**5.57E-15**(1)
	S.D.	4.72E-07	5.12E+04	1.73E+02	1.86E+02	3.05E-14	1.94E-07	3.11E-08	1.39E-03	2.28E-04	2.31E+05	1.71E-14
f_4	Mean(R)	3.39E+01(3)	5.92E+01(7)	1.03E+02(11)	9.05E+01(10)	**7.82E-02**(1)	6.84E+01(9)	6.23E+01(8)	4.57E+01(5)	4.20E+01(4)	2.73E+01(2)	4.94E+01(6)
	S.D.	3.80E+01	4.76E+01	4.25E+01	5.45E+01	5.58E-01	4.44E+01	3.82E+01	4.80E+01	3.03E+01	2.10E+01	2.32E+00
f_5	Mean(R)	7.52E+01(3)	8.09E+01(5)	1.99E+02(9)	4.65E+02(10)	6.81E+02(11)	8.77E+01(6)	7.53E+01(4)	5.89E+01(2)	9.56E+01(7)	1.51E+02(8)	**1.16E+01**(1)
	S.D.	1.60E+01	8.36E+00	1.31E+01	1.10E+02	4.62E+02	2.03E+01	1.70E+01	1.56E+01	2.61E+01	1.76E+01	2.45E+00
f_6	Mean(R)	3.41E-08(6)	3.33E-06(8)	**1.11E-13**(1)	7.92E-03(10)	8.74E+01(11)	1.77E-08(4)	2.02E-08(5)	1.67E-03(9)	1.83E-13(3)	1.45E-13(2)	1.03E-06(7)
	S.D.	1.37E-07	6.44E-06	1.59E-14	3.39E-03	1.55E+01	3.88E-08	2.63E-08	5.24E-03	5.61E-14	5.12E-14	3.75E-07
f_7	Mean(R)	1.21E+02(3)	1.33E+02(5)	2.70E+02(9)	5.58E+02(11)	3.25E+02(10)	1.37E+02(6)	1.22E+02(4)	1.05E+02(2)	1.77E+02(7)	2.07E+02(8)	**4.32E+01**(1)
	S.D.	1.70E+01	8.33E+00	1.32E+01	7.94E+01	7.17E+01	2.14E+01	1.60E+01	1.18E+01	7.00E+01	1.45E+01	2.48E+00
f_8	Mean(R)	7.53E+01(4)	8.00E+01(5)	2.00E+02(9)	4.61E+02(10)	6.45E+02(11)	8.96E+01(6)	7.33E+01(3)	5.22E+01(2)	9.09E+01(7)	1.50E+02(8)	**2.44E+01**(1)
	S.D.	1.66E+01	7.75E+00	1.28E+01	1.08E+02	4.59E+02	1.99E+01	1.62E+01	1.12E+01	1.94E+01	1.79E+01	1.87E+00
f_9	Mean(R)	5.14E-01(7)	4.41E-02(4)	2.26E+00(10)	8.35E-02(6)	2.81E+04(11)	1.77E-02(3)	7.82E-02(5)	8.71E-01(9)	5.27E-03(2)	5.26E-01(8)	**0.00E+00**(1)
	S.D.	8.79E-01	1.03E-01	1.72E+00	2.67E-01	9.51E+03	7.02E-02	1.93E-01	7.14E-01	2.13E-02	2.17E+00	0.00E+00
f_{10}	Mean(R)	5.03E+03(5)	5.43E+03(6)	8.74E+03(9)	1.71E+04(11)	7.51E+03(7)	**4.24E+03**(1)	4.32E+03(2)	4.80E+03(4)	8.02E+03(8)	8.79E+03(10)	4.42E+03(3)
	S.D.	1.54E+03	3.10E+02	3.02E+02	1.77E+03	8.53E+02	7.23E+02	5.88E+02	7.56E+02	1.61E+03	4.95E+02	2.66E+02
f_{11}	Mean(R)	5.10E+01(4)	1.22E+02(10)	1.12E+02(9)	6.20E+01(6)	9.18E+03(11)	5.14E+01(5)	4.50E+01(2)	9.72E+01(7)	4.66E+01(3)	1.07E+02(8)	**2.28E+01**(1)
	S.D.	9.66E+00	7.21E+01	2.49E+01	5.24E+01	1.28E+04	1.02E+01	7.58E+00	2.42E+01	7.34E+00	6.52E+01	1.95E+01
f_{12}	Mean(R)	5.02E+04(7)	3.67E+03(3)	6.29E+04(8)	2.53E+05(11)	2.59E+03(2)	7.53E+04(9)	4.57E+04(5)	1.16E+04(4)	4.83E+04(6)	1.99E+05(10)	**1.49E+03**(1)
	S.D.	3.11E+04	1.80E+03	3.85E+04	1.72E+05	5.43E+02	4.27E+04	3.09E+04	1.14E+04	2.97E+04	1.53E+05	2.17E+02
f_{13}	Mean(R)	**7.82E+01**(1)	1.24E+02(4)	5.42E+02(5)	6.09E+02(6)	5.49E+08(11)	1.04E+03(8)	6.68E+02(7)	9.15E+01(3)	2.71E+03(9)	1.39E+04(10)	8.00E+01(2)
	S.D.	4.26E+01	6.44E+01	6.52E+02	3.77E+02	7.51E+08	1.79E+03	8.28E+02	4.00E+01	2.51E+03	3.06E+04	6.40E+00
f_{14}	Mean(R)	4.15E+01(3)	1.50E+04(10)	1.17E+02(8)	1.05E+02(7)	2.17E+05(11)	4.55E+01(5)	3.79E+01(2)	6.56E+01(6)	4.42E+01(4)	1.84E+03(9)	**2.59E+01**(1)
	S.D.	1.04E+01	3.89E+04	3.00E+01	8.83E+01	9.52E+05	1.27E+01	8.22E+00	1.50E+01	1.36E+01	3.52E+03	1.10E+00
f_{15}	Mean(R)	3.98E+01(2)	1.63E+02(9)	1.30E+02(8)	1.09E+02(6)	4.77E+08(11)	5.68E+01(4)	4.41E+01(3)	6.83E+01(5)	1.20E+02(7)	5.81E+03(10)	**2.86E+01**(1)
	S.D.	8.78E+00	8.00E+01	5.99E+01	7.69E+01	3.30E+08	2.47E+01	2.00E+01	2.38E+01	2.36E+02	2.53E+04	1.37E+00
f_{16}	Mean(R)	8.40E+02(2)	9.76E+02(5)	1.03E+03(8)	1.62E+03(11)	1.62E+03(10)	9.48E+02(4)	1.00E+03(7)	9.91E+02(6)	8.99E+02(3)	1.26E+03(9)	**2.73E+01**(1)
	S.D.	2.54E+02	1.85E+02	1.40E+02	1.52E+03	1.09E+03	2.76E+02	3.20E+02	2.88E+02	2.06E+02	1.99E+02	3.90E+01
f_{17}	Mean(R)	**5.17E+02**(1)	7.47E+02(7)	7.87E+02(8)	2.01E+03(10)	2.46E+03(11)	5.83E+02(3)	5.86E+02(4)	5.96E+02(5)	6.02E+02(6)	8.68E+02(9)	5.21E+02(2)
	S.D.	1.57E+02	1.16E+02	1.10E+02	1.45E+03	1.60E+03	2.12E+02	1.92E+02	1.62E+02	1.38E+02	1.73E+02	5.67E+00
f_{18}	Mean(R)	8.88E+01(2)	1.98E+02(4)	4.60E+02(7)	6.92E+02(8)	2.57E+02(5)	9.83E+02(9)	3.52E+02(6)	1.34E+02(3)	3.02E+03(10)	4.62E+03(11)	**2.09E+01**(1)
	S.D.	6.23E+01	9.20E+01	4.58E+02	4.01E+02	1.84E+02	7.08E+02	3.03E+02	9.67E+01	3.19E+03	3.23E+03	6.56E-01
f_{19}	Mean(R)	1.42E+01(2)	5.10E+02(9)	3.06E+01(6)	8.72E+01(8)	1.72E+08(11)	1.73E+01(4)	**1.37E+01**(1)	4.09E+01(7)	2.17E+01(5)	7.44E+02(10)	1.72E+01(3)
	S.D.	3.58E+00	2.09E+03	1.27E+01	1.05E+02	9.97E+07	5.12E+00	3.88E+00	1.83E+01	2.47E+01	2.16E+03	1.73E+00
f_{20}	Mean(R)	3.84E+02(2)	6.07E+02(9)	5.51E+02(7)	5.93E+02(8)	1.94E+03(11)	4.06E+02(6)	**3.83E+02**(1)	3.93E+02(4)	3.95E+02(5)	6.33E+02(10)	3.84E+02(3)
	S.D.	1.74E+02	1.27E+02	1.21E+02	4.95E+02	4.58E+02	1.77E+02	1.76E+02	1.81E+02	1.67E+02	1.33E+02	1.20E+02
f_{21}	Mean(R)	2.77E+02(4)	2.81E+02(5)	3.90E+02(9)	6.59E+02(11)	4.27E+02(10)	2.86E+02(6)	2.70E+02(3)	2.50E+02(2)	2.91E+02(7)	3.53E+02(8)	**2.12E+02**(1)
	S.D.	1.72E+01	8.57E+00	1.27E+01	1.16E+02	3.26E+02	1.78E+01	1.68E+01	9.46E+00	2.10E+01	2.12E+02	2.51E+00
f_{22}	Mean(R)	4.96E+03(7)	4.50E+03(6)	**1.47E+03**(1)	1.43E+04(11)	8.71E+03(9)	3.77E+03(4)	4.28E+03(5)	2.45E+03(3)	6.44E+03(8)	9.54E+03(10)	1.62E+03(2)
	S.D.	2.19E+03	2.45E+03	3.23E+03	6.29E+03	1.33E+03	1.96E+03	1.45E+03	2.68E+03	3.78E+03	3.90E+02	1.77E+02
f_{23}	Mean(R)	5.03E+02(5)	5.02E+02(4)	6.18E+02(9)	8.89E+02(10)	1.09E+03(11)	5.15E+02(6)	4.97E+02(3)	**4.79E+02**(1)	5.18E+02(7)	5.62E+02(8)	4.83E+02(2)
	S.D.	1.86E+01	1.09E+01	3.19E+01	1.12E+02	1.22E+03	2.11E+01	1.87E+01	1.58E+01	2.67E+01	2.88E+01	4.09E+00
f_{24}	Mean(R)	5.81E+02(8)	5.52E+02(4)	5.48E+02(3)	5.59E+02(5)	5.59E+02(6)	5.82E+02(9)	5.59E+02(7)	5.40E+02(2)	5.84E+02(10)	6.41E+02(11)	**4.78E+02**(1)
	S.D.	1.83E+01	8.98E+00	1.45E+01	2.28E+01	9.95E+01	1.93E+01	1.52E+01	1.52E+01	1.78E+01	2.02E+01	2.51E+00
f_{25}	Mean(R)	5.13E+02(10)	5.03E+02(6)	5.57E+02(11)	5.09E+02(9)	**4.32E+02**(2)	5.05E+02(7)	5.07E+02(8)	4.44E+02(3)	4.89E+02(5)	4.46E+02(4)	3.87E+02(1)
	S.D.	3.18E+01	3.22E+01	3.17E+01	3.41E+01	6.18E+00	3.13E+01	3.31E+01	1.73E+01	1.92E+01	1.93E+01	6.03E-03
f_{26}	Mean(R)	1.91E+03(4)	1.76E+03(2)	2.56E+03(9)	4.60E+03(11)	2.24E+03(8)	2.02E+03(6)	1.84E+03(3)	**1.60E+03**(1)	2.08E+03(7)	2.58E+03(10)	1.94E+03(5)
	S.D.	1.69E+02	1.04E+02	4.76E+02	1.49E+03	1.07E+03	2.31E+02	2.14E+02	1.25E+02	2.03E+02	4.46E+02	5.17E+01
f_{27}	Mean(R)	5.18E+02(5)	5.31E+02(9)	5.63E+02(11)	5.08E+02(3)	5.00E+02(2)	5.31E+02(8)	5.12E+02(4)	5.44E+02(10)	5.29E+02(7)	**5.00E+02**(1)	5.25E+02(6)
	S.D.	1.37E+01	1.45E+01	2.80E+01	8.18E+00	8.83E-05	1.88E+01	9.54E+00	1.89E+01	1.07E+01	1.01E-04	5.92E+00
f_{28}	Mean(R)	4.76E+02(5)	4.90E+02(8)	5.19E+02(11)	4.63E+02(3)	4.99E+02(9)	4.74E+02(4)	4.63E+02(2)	4.89E+02(7)	**4.59E+02**(1)	5.00E+02(10)	4.83E+02(6)
	S.D.	2.27E+01	2.29E+01	2.81E+01	1.35E+01	4.93E+00	2.16E+01	1.33E+01	2.60E+01	5.31E-13	1.12E-04	3.86E+01
f_{29}	Mean(R)	**4.15E+02**(1)	5.09E+02(6)	6.51E+02(8)	2.70E+03(11)	1.06E+03(10)	5.28E+02(7)	4.39E+02(3)	4.44E+02(4)	4.84E+02(5)	9.56E+02(9)	4.35E+02(2)
	S.D.	9.79E+01	6.98E+01	7.66E+01	2.16E+03	1.24E+03	1.59E+02	1.13E+02	1.21E+02	1.13E+02	1.97E+02	8.16E+00
f_{30}	Mean(R)	5.94E+05(3)	6.41E+05(7)	6.92E+05(8)	6.16E+05(6)	5.81E+07(11)	5.97E+05(4)	5.93E+05(2)	7.14E+05(10)	6.11E+05(5)	**1.30E+03**(1)	6.55E+05(8)
	S.D.	2.46E+04	6.01E+04	6.45E+04	2.88E+04	1.22E+08	2.59E+04	2.61E+04	1.15E+05	5.22E+04	1.61E+03	5.20E+04
Avg.	(R)	4.13	6.23	8.00	8.63	7.93	5.80	4.13	4.73	5.77	8.03	2.43

that adaptive population size is beneficial for this type of functions. Although JADE, SaDE, and MPEDE display promising characteristics on the simple multimodal functions, they offer unfavorable results on the hybrid functions. The comparison results manifest that MRDE can solve the complicated hybrid functions more efficiently.

4) Composition functions (f_{21}-f_{30}): For the 10 composition functions, which are very difficult to be optimized, all the peer algorithms display almost the same performance, in terms of the mean values. However, we also find out that MRDE yields the best performance in 10D case, in terms of the number of the achieved best results.

Table 4
Scores and final rank of all peer algorithms on CEC2017 test suite.

Algorithms	*Score1*	*Score2*	*Score*	*Final Rank*
LSHADE-cnEpSin	44.45	38.75	83.20	1
MRDE	19.56	35.72	55.28	2
TSDE	16.25	29.64	45.89	3
MPEDE	20.23	24.05	44.28	4
EFADE	15.96	22.20	38.16	5
CoBiDE	14.42	19.88	34.30	6
SaDE	10.70	17.59	28.29	7
CoDE	8.14	16.77	24.91	8
JADE	2.70	18.77	21.47	9
EPSDE	7.85	13.03	20.88	10
DCMA-EA	3.03	14.31	17.34	11

Table 6
Results of the multiple-problem Wilcoxon's test.

MRDE vs.	R^+	R^-	*p*-value	Significant?
JADE	68	18	0.000	Yes
SaDE	68	12	0.000	Yes
CoDE	77	10	0.000	Yes
DCMA-EA	72	13	0.000	Yes
CoBiDE	69	13	0.000	Yes
TSDE	49	30	0.007	Yes
MPEDE	64	22	0.003	Yes
EFADE	64	16	0.000	Yes
EPSDE	74	73	0.000	Yes
LSHAD_cnEpSin	33	45	0.076	No

Moreover, we still can find out that LSHADE_cnEpSin and MRDE achieve more outstanding performance than other DE algorithms on 30D and 50D cases.

From the Avg. (R) in Tables 1–3 we can observe that MRDE presents the best performance on 10D case, while it archives the second best results on 30D and 50D cases, which is slightly worsen than LSHADE_cnEpSin. From the comparison results on the three dimension cases, we find out that LSHADE_cnEpSin displays the best comprehensive performance, followed by MRDE.

To give an overall performance of the eleven peer algorithms on the three dimension cases, a final rank based on two scores are listed in Table 4. Note that the details of the evaluation criteria are presented in the Appendix part. Table 4 shows that LSHADE_cnEpSin and MRDE yield the most favorable overall performance, followed by TSDE and MPEDE, while DCMA-EA offers the most unpromising overall performance though it attains very outstanding results on the unimodal functions.

4.3. Nonparametric test results

Generally, it is necessary to use statistical tests to analyze the experimental results. In this section, two nonparametric tests, including Friedman test and Wilcoxon's test, are carried out to validate the performance of all the algorithms, in terms of the mean value of independent 51 runs. The results of Friedman test on the three dimension cases are listed in Table 5, in which each algorithm and its ranking values are listed in ascending order (the lower the better). The results of Wilcoxon' test between MRDE and other 10 peer algorithms are presented in Table 6. Note that the significance level of the two tests is $\alpha = 0.05$. If the p-value obtained in the hypothesis test is less than α, the difference of results is statistically significant.

Friedman test results listed in Table 5 manifest that MRDE yields the overall best performance on 10D case. On 30D and 50D case, it offers the second best and the third best comprehensive characteristics, respectively.

The results of multiple-problem Wilcoxon's test presented in Table 6 is adopted to check the performance difference between MRDE and other 10 peer algorithms on all the three dimension cases. From the results we can see that there is no significant difference between MRDE and LSHAD_cnEpSin. Moreover, the test results also show that MRDE dominates other 9 peer algorithms.

5. Sensitivity analysis of new introduced strategies

There are many additional strategies in MRDE. Due to the space limitation, we only analyze the performance of the two key strategies in this section, including the multiple roles of individuals and change of population size strategy. Furthermore, an additional experiment is carried out intending to illustrate the performance of the group size gs. Due to the space limitation we conduct the experiments only on CEC2017 test suite on 30D case. Note that, for the fairness of the comparison, while $gs = 4$ or 5, gn are set as 19 or 15, respectively. In this condition, both of them have almost the same population size as MRDE. The experimental results, in terms of mean and standard deviation of independent 51 runs, are listed in Tables 7 and 8. Furthermore, results of t-tests between MRDE and other MRDE variants are also display in the tables.

In Table 7, MRDE/MR(RR) denotes that the multiple roles (MR) is replaced by a random role assignment (RR). In RR, the three roles are randomly assigned to three individuals in a group rather than relying on the individuals' fitness. In MRDE/AS(US), the adaptive population size (AS) is replaced by an unchangeable population size (US). Note that, there is also no randomly regroup operator in MRDE/AS(US).

Table 5
Friedman-test on CEC2017 test suite.

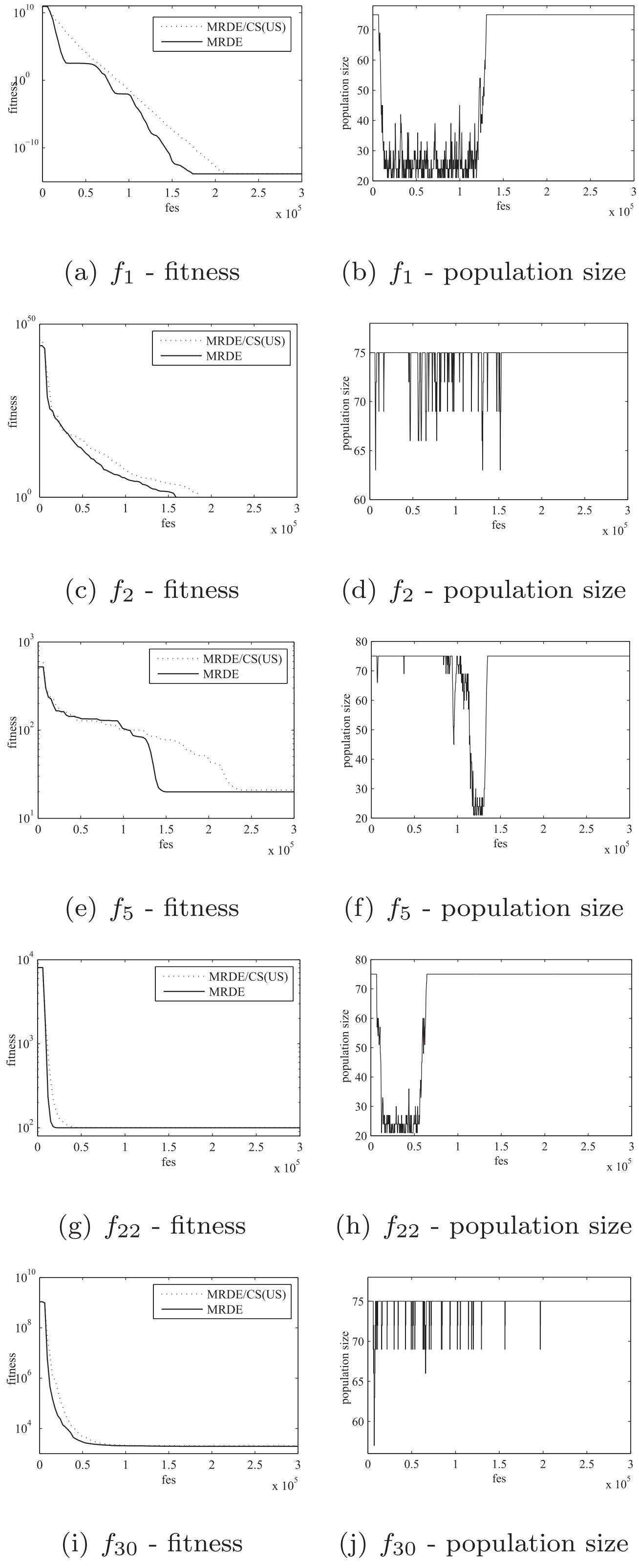
	10D		30D		50D	
	Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking
1	MRDE	2.80	LSHAD_cnEpSin	2.83	LSHAD_cnEpSin	2.48
2	TSDE	3.48	MRDE	3.58	TSDE	4.09
3	LSHAD_cnEpSin	4.60	MPEDE	4.73	MRDE	4.16
3	SaDE	5.15	TSDE	5.28	MPEDE	4.71
4	EFADE	5.20	EFADE	5.58	CoBiDE	5.74
5	CoBiDE	5.35	CoDE	5.70	EFADE	5.93
6	CoDE	7.13	JADE	6.32	JADE	6.21
7	JADE	7.30	SaDE	7.27	EPSDE	7.95
8	MPEDE	7.42	EPSDE	7.85	SaDE	7.97
10	EPSDE	8.12	CoBiDE	7.90	DCMA-EA	8.19
9	DCMA-EA	9.45	DCMA-EA	8.95	CoDE	8.59
Statistic	124.200		99.831		106.602	
p value	0.000		0.000		0.000	

Table 7
Performance of the two introduced strategies on CEC2017 test suite (30D).

	MRDE	MRDE/MR(RR)	MRDE/AS(US)
f_1	1.23E-14 ± 4.94E-15	6.35E-11 ± 6.77E-11(+)	1.25E-14 ± 4.62E-15(≈)
f_2	0.00E+00 ± 0.00E+00	8.32E+03 ± 4.80E+04(+)	0.00E+00 ± 0.00E+00(≈)
f_3	3.50E-13 ± 8.54E-13	7.90E-05 ± 1.08E-04(+)	4.54E-12 ± 5.57E-12(+)
f_4	4.26E+01 ± 2.73E+01	5.43E+01 ± 1.77E+01(+)	4.67E+01 ± 2.54E+01(≈)
f_5	3.17E+01 ± 9.11E+00	8.19E+01 ± 1.94E+01(+)	3.15E+01 ± 8.21E+00(≈)
f_6	2.68E-09 ± 1.92E-08	2.17E-07 ± 1.02E-07(+)	2.68E-09 ± 1.92E-08(≈)
f_7	6.40E+01 ± 8.71E+00	1.29E+02 ± 1.84E+01(+)	6.29E+01 ± 6.68E+00(≈)
f_8	3.15E+01 ± 9.14E+00	8.33E+01 ± 1.51E+01(+)	3.53E+01 ± 9.62E+00(+)
f_9	9.14E-14 ± 4.56E-14	1.78E-14 ± 4.18E-14(−)	1.07E-02 ± 6.46E-02(≈)
f_{10}	2.37E+03 ± 7.86E+02	4.32E+03 ± 3.56E+02(+)	2.54E+03 ± 7.17E+02(≈)
f_{11}	1.38E+01 ± 1.44E+01	1.26E+01 ± 9.23E+00(≈)	1.51E+01 ± 1.75E+01(≈)
f_{12}	3.59E+03 ± 4.98E+03	4.38E+03 ± 3.18E+03(≈)	2.20E+03 ± 3.03E+03(≈)
f_{13}	1.89E+01 ± 7.81E+00	4.55E+01 ± 2.06E+01(+)	1.84E+01 ± 8.13E+00(≈)
f_{14}	9.19E+00 ± 5.44E+00	1.38E+01 ± 8.79E+00(+)	9.36E+00 ± 6.71E+00(≈)
f_{15}	7.30E+00 ± 3.00E+00	8.82E+00 ± 2.60E+00(+)	6.42E+00 ± 2.48E+00(≈)
f_{16}	3.53E+02 ± 1.60E+02	3.18E+02 ± 1.42E+02(≈)	3.22E+02 ± 1.63E+02(≈)
f_{17}	3.99E+01 ± 1.20E+01	5.29E+01 ± 1.13E+01(+)	3.92E+01 ± 1.05E+01(≈)
f_{18}	1.91E+01 ± 9.43E+00	2.45E+01 ± 1.93E+00(+)	1.92E+01 ± 8.40E+00(≈)
f_{19}	5.14E+00 ± 1.58E+00	1.25E+01 ± 3.41E+00(+)	5.84E+00 ± 1.54E+00(+)
f_{20}	4.30E+01 ± 5.88E+01	4.77E+01 ± 5.31E+01(≈)	2.98E+01 ± 4.98E+01(≈)
f_{21}	2.36E+02 ± 1.04E+01	2.69E+02 ± 2.61E+01(+)	2.32E+02 ± 9.14E+00(≈)
f_{22}	1.00E+02 ± 3.26E-13	1.00E+02 ± 5.95E-13(≈)	1.00E+02 ± 3.05E-13(≈)
f_{23}	3.84E+02 ± 1.26E+01	4.02E+02 ± 2.33E+01(+)	3.84E+02 ± 1.02E+01(≈)
f_{24}	4.60E+02 ± 1.05E+01	4.61E+02 ± 1.39E+01(≈)	4.56E+02 ± 1.11E+01(−)
f_{25}	3.87E+02 ± 4.93E-02	3.87E+02 ± 4.71E-01(≈)	3.87E+02 ± 5.36E-02(≈)
f_{26}	1.27E+03 ± 2.27E+02	1.38E+03 ± 2.00E+02(+)	1.21E+03 ± 2.31E+02(≈)
f_{27}	4.92E+02 ± 9.62E+00	4.87E+02 ± 1.10E+01(−)	4.94E+02 ± 9.56E+00(≈)
f_{28}	3.22E+02 ± 4.52E+01	3.13E+02 ± 3.48E+01(≈)	3.23E+02 ± 4.57E+01(≈)
f_{29}	4.07E+02 ± 4.31E+01	4.89E+02 ± 5.50E+01(+)	4.18E+02 ± 3.79E+01(≈)
f_{30}	1.99E+03 ± 4.20E+01	1.99E+03 ± 3.53E+01(≈)	1.98E+03 ± 3.45E+01(≈)
(#)+		19	3
(#) ≈		9	26
(#) -		2	1

Table 8
Performance of three different *gs* values on CEC2017 test suite (30D).

	MRDE($gs = 3$)	MRDE($gs = 4$)	MRDE($gs = 5$)
f_1	1.23E-14 ± 4.94E-15	5.26E+01 ± 1.82E+02(+)	4.98E+02 ± 5.22E+02(+)
f_2	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00(≈)	0.00E+00 ± 0.00E+00(≈)
f_3	3.50E-13 ± 8.54E-13	4.57E-14 ± 2.28E-14(−)	5.35E-14 ± 1.35E-14(−)
f_4	4.26E+01 ± 2.73E+01	5.45E+01 ± 1.88E+01(+)	1.24E+01 ± 2.17E+01(−)
f_5	3.17E+01 ± 9.11E+00	3.35E+01 ± 9.74E+00(≈)	4.11E+01 ± 9.50E+00(+)
f_6	2.68E-09 ± 1.92E-08	8.33E-07 ± 1.71E-06(+)	1.78E-05 ± 2.04E-05(+)
f_7	6.40E+01 ± 8.71E+00	6.62E+01 ± 6.84E+00(≈)	7.14E+01 ± 1.15E+01(+)
f_8	3.15E+01 ± 9.14E+00	3.63E+01 ± 1.15E+01(+)	3.73E+01 ± 8.95E+00(+)
f_9	9.14E-14 ± 4.56E-14	1.98E-01 ± 4.25E-01(+)	8.28E-01 ± 6.93E-01(+)
f_{10}	2.37E+03 ± 7.86E+02	1.90E+03 ± 4.70E+02(+)	2.11E+03 ± 3.62E+02(≈)
f_{11}	1.38E+01 ± 1.44E+01	1.88E+01 ± 1.62E+01(≈)	1.76E+01 ± 8.25E+00(≈)
f_{12}	3.59E+03 ± 4.98E+03	4.07E+03 ± 6.94E+03(≈)	1.23E+04 ± 9.83E+03(+)
f_{13}	1.89E+01 ± 7.81E+00	1.81E+01 ± 7.67E+00(≈)	4.32E+01 ± 4.24E+01(+)
f_{14}	9.19E+00 ± 5.44E+00	1.14E+01 ± 7.47E+00(≈)	1.59E+01 ± 1.06E+01(+)
f_{15}	7.30E+00 ± 3.00E+00	6.31E+00 ± 3.33E+00(≈)	1.17E+01 ± 4.40E+00(+)
f_{16}	3.53E+02 ± 1.60E+02	4.54E+02 ± 1.81E+02(+)	5.50E+02 ± 2.58E+02(≈)
f_{17}	3.99E+01 ± 1.20E+01	7.84E+01 ± 6.97E+01(+)	6.70E+01 ± 6.84E+01(+)
f_{18}	1.91E+01 ± 9.43E+00	1.98E+01 ± 1.00E+01(≈)	2.62E+01 ± 1.18E+01(+)
f_{19}	5.14E+00 ± 1.58E+00	5.06E+00 ± 1.59E+00(≈)	7.95E+00 ± 5.95E+00(+)
f_{20}	4.30E+01 ± 5.88E+01	8.10E+01 ± 8.24E+01(+)	1.00E+02 ± 8.54E+01(+)
f_{21}	2.36E+02 ± 1.04E+01	2.38E+02 ± 1.30E+01(≈)	2.38E+02 ± 6.67E+00(≈)
f_{22}	1.00E+02 ± 3.26E-13	1.01E+02 ± 1.52E+00(+)	1.02E+02 ± 2.30E+00(+)
f_{23}	3.84E+02 ± 1.26E+01	3.86E+02 ± 1.04E+01(≈)	3.88E+02 ± 1.03E+01(≈)
f_{24}	4.60E+02 ± 1.05E+01	4.55E+02 ± 1.72E+01(≈)	4.60E+02 ± 9.45E+00(≈)
f_{25}	3.87E+02 ± 4.93E-02	3.87E+02 ± 3.85E-01(≈)	3.88E+02 ± 7.73E-01(≈)
f_{26}	1.27E+03 ± 2.27E+02	1.45E+03 ± 1.35E+02(+)	1.29E+03 ± 2.86E+02(≈)
f_{27}	4.92E+02 ± 9.62E+00	5.00E+02 ± 1.01E+01(+)	4.95E+02 ± 9.73E+00(≈)
f_{28}	3.22E+02 ± 4.52E+01	3.31E+02 ± 5.04E+01(≈)	3.52E+02 ± 7.15E+01(+)
f_{29}	4.07E+02 ± 4.31E+01	4.25E+02 ± 6.16E+01(≈)	4.48E+02 ± 5.83E+01(+)
f_{30}	1.99E+03 ± 4.20E+01	2.01E+03 ± 5.70E+01(+)	2.04E+03 ± 8.07E+01(+)
(#)+		13	18
(#) ≈		16	10
(#) -		1	2



(a) f_1 - fitness (b) f_1 - population size

(c) f_2 - fitness (d) f_2 - population size

(e) f_5 - fitness (f) f_5 - population size

(g) f_{22} - fitness (h) f_{22} - population size

(i) f_{30} - fitness (j) f_{30} - population size

Fig. 2. Change trends of fitness and population size on the 5 test functions.

5.1. Performance of multi-role strategy

From the comparison results between MRDE and MRDE/MR(RR) we can see that MRDE dominates MRDE/MR(RR) on the majority of test functions except f_9, f_{11}, f_{16}, f_{27}, and f_{28}. However, the t-test results manifest that MRDE/MR(RR) yields significant better performance than MRDE only on f_9 and f_{27}. On the contrary, MRDE is significantly better than MRDE/MR(RR) on 19 out of the 30 test function. Thus, we regard assigning proper roles for different individuals enables the individuals to focus on their own search objectives (i.e., exploration and exploitation), and then help the population conduct a more efficient diverse search behavior.

5.2. Performance of adaptive population size

The comparison results between MRDE and MRDE/AS(US) indicate that the adaptive population size strategy has no significant effect, in terms of the mean values, because MRDE and MRDE/AS(US) offer almost the same performance on the 26 out of the 30 functions, in terms of the t-test results. However, it is not to say that the adaptive population size has no effect on the MRDE. In fact, the motivation of the strategy is to rationally distribute the computational resource during the evolutionary process, and then speed up the convergence. Thus, to verify the performance of the adaptive strategy on convergence speed, we choose f_1, f_2, f_5, f_{22}, and f_{30} as test functions to carry out the comparison experiments since MRDE and MRDE/AS(US) achieve almost the same accurate solutions on the 5 functions. The results are demonstrated in Fig. 2, in which the left subfigures denote the change of fitness while the right subfigures illustrate the fluctuation of the population size.

From Fig. 2(a) we can observe that MRDE offers a higher convergence speed than MRDE/AS(US) during the first half evolutionary process, though both of them yield almost the same performance at the latter evolutionary stage, in terms of convergence speed and accuracy. The experiment result illustrated in Fig. 2(b) manifests that the population size of MRDE fluctuates between 21 and 42 in the initial evolutionary stage, which means that a relatively smaller population size is enough to help the population keep a steady improvement. Together with the results in Fig. 2(a) we can find out that it is the small population size that causes MRDE to yield a higher convergence speed during the initial stage. On the contrary, the population size in MRDE reaches the upper limitation during the later evolutionary stage because the entire population cannot find out a more accurate solution in the stage. As a result, MRDE and MRDE/AS(US) display almost the same performance in the later stage. Moreover, the experiment on f_{22} also verifies that a smaller population size resulted by the adaptive adjustment enables MRDE to achieve a higher convergence speed during the initial evolutionary stage.

Although MRDE and MRDE/AS(US) find out the global optimal solution for f_2, MRDE has higher convergency speed than MRDE/AS(US). The change trend of population size of MRDE demonstrated in Fig. 2(d) can explain the slightly difference. During the first half evolutionary stage, the population size of MRDE fluctuates between 63 and 75. Under the condition, MRDE achieves a similar performance as MRDE/AS(US) with much less fitness evaluations. Hence, MRDE can display slightly better convergence performance than MRDE/AS(US). Furthermore, the phenomenon extends to the composition function f_{30}.

For the results on function f_5, we can capture an interesting phenomenon from Fig. 2(d) that the population size of MRDE is greatly fluctuated at the middle of evolutionary stage, rather than at the initial stage. Moreover, Fig. 2(c) shows that the convergence speed of MRDE is dramatically accelerated in the middle evolutionary stage, and significantly faster than MRDE/AS(US), though both of them display almost the same convergence performance at other evolutionary stages. Thus, we regard that it is the smaller population size accelerates the convergence speed of MRDE during the middle stage.

From the above experimental results, we can draw a conclusion that the adaptive population size is favorable for different functions to a certain extent. However, it needs to be further study that how to effectively adjust population size according to various fitness landscapes appeared in different evolutionary stages.

Table 9
Performance of the selected mutation operators and control parameters on CEC2017 test suite (30D).

	MRDE	EPSDE/(*Settings*)	EPSDE
f_1	1.23E-14±4.94E-15	1.29E-12±3.32E-12	2.20E-13±6.03E-13
f_2	0.00E+00±0.00E+00	5.24E+17±2.35E+18	6.92E+16±3.22E+17
f_3	3.50E-13±8.54E-13	1.75E-12±1.06E-12	5.53E+03±2.19E+04
f_4	4.26E+01±2.73E+01	2.25E+00±2.39E+00	1.25E+00±1.87E+00
f_5	3.17E+01±9.11E+00	5.02E+01±8.50E+00	4.44E+01±8.19E+00
f_6	2.68E-09±1.92E-08	7.68E-02±3.10E-01	1.14E-13±0.00E+00
f_7	6.40E+01±8.71E+00	9.45E+01±1.48E+01	7.71E+01±6.56E+00
f_8	3.15E+01±9.14E+00	5.18E+01±7.16E+00	4.51E+01±7.38E+00
f_9	9.14E-14±4.56E-14	2.91E+00±8.99E+00	4.25E-02±1.31E-01
f_{10}	2.37E+03±7.86E+02	3.77E+03±4.51E+02	3.72E+03±3.41E+02
f_{11}	1.38E+01±1.44E+01	1.91E+02± 8.88E+01	2.86E+01±2.02E+01
f_{12}	3.59E+03±4.98E+03	1.91E+04±1.88E+04	4.11E+04±4.11E+04
f_{13}	1.89E+01±7.81E+00	1.59E+03±8.74E+02	3.10E+03±9.14E+03
f_{14}	9.19E+00±5.44E+00	2.08E+02±9.00E+01	6.24E+01±3.51E+01
f_{15}	7.30E+00±3.00E+00	5.64E+02±3.74E+02	1.32E+02±2.30E+02
f_{16}	3.53E+02±1.60E+02	5.17E+02±1.54E+02	5.94E+02±1.29E+02
f_{17}	3.99E+01±1.20E+01	1.49E+02±7.18E+01	2.00E+02±7.43E+01
f_{18}	1.91E+01±9.43E+00	1.68E+03±1.69E+03	1.07E+03±1.15E+03
f_{19}	5.14E+00±1.58E+00	2.86E+02±2.29E+02	2.74E+01±2.23E+01
f_{20}	4.30E+01±5.88E+01	1.57E+02±5.80E+01	1.63E+02±7.16E+01
f_{21}	2.36E+02±1.04E+01	2.51E+02±7.98E+00	2.46E+02±7.90E+00
f_{22}	1.00E+02±3.26E-13	3.51E+03±1.56E+03	3.31E+03±1.74E+03
f_{23}	3.84E+02±1.26E+01	4.15E+02±1.36E+01	3.98E+02±1.04E+01
f_{24}	4.60E+02±1.05E+01	5.10E+02±2.37E+01	4.67E+02±7.87E+00
f_{25}	3.87E+02±4.93E-02	3.79E+02±5.55E+00	3.78E+02±9.97E-01
f_{26}	1.27E+03±2.27E+02	1.67E+03±3.06E+02	1.21E+03±1.26E+02
f_{27}	4.92E+02±9.62E+00	5.00E+02±9.40E-05	5.00E+02±9.37E-05
f_{28}	3.22E+02±4.52E+01	5.00E+02±1.25E+00	5.00E+02±2.03E+00
f_{29}	4.07E+02±4.31E+01	4.31E+02±9.90E+01	4.89E+02±8.73E+01
f_{30}	1.99E+03±4.20E+01	6.27E+02±3.58E+02	2.22E+02±9.00E+00

5.3. Performance of group size

The comparison results listed in Table 8 show that the performance of MRDE steadily deteriorate along with increase of *gs*. Specifically, MRDE($gs = 4$) is dominated by MRDE on 13 out of the 30 functions, while MRDE($gs = 5$) is significantly worsen than MRDE on 18 functions. Although the 10 composition functions (f_{21}-f_{30}) is very difficult to be optimized, the MRDE($gs = 3$) dominates MRDE($gs = 4$) and MRDE($gs = 5$) on 4 out of the 10 problems, in terms of the *t*-test results. Thus, a few preliminary conclusions can be obtained through the experimental results. Firstly, there are not enough elites (or inferiors) in the population to perform exploitation (or exploration) ability while *gs* increasing because there are only one elite and one inferior included in each group. In addition, different values of *gs* lead to the different number of individuals that delete from or insert into the current population. Lastly, we must admit that effect on *gs* need to be further studied in our future works.

5.4. Performance of selected mutation operators and control parameters

In MRDE, three settings of mutation operators and control parameters (i.e., *Setting* 1, *Setting* 2, and *Setting* 3), which have been detailed in Section 3.2, are used to help individuals to breed offspring. In this section, an experiment is conducted to verify whether the combination of the settings can still offer a favorable performance without other strategies applied in MRDE.

EPSDE is a general method and any mutation operators and control parameters can be included. Thus, mutation operators and parameters involved in EPSDE are replaced by the mutation operators and control parameters applied in MRDE aiming to discern the performance of them. The new EPSDE variant is named as EPSDE/(*Settings*) in this part. Considering that there is no local best solution of a group in the EPSDE framework, the symbol "lbest" in "DE/current-to-lbest/1" in EPSDE/(Settings) denotes the best solution of three randomly selected target vectors.

The experimental results of MRDE, EPSDE, and EPSDE/(*Settings*) on CEC2017 test in 30D case are illustrated in Table 9, in which a result with a shaded background denotes that EPSDE/(*Settings*) attains better performance than EPSDE.

The experimental results show that EPSDE/(*Settings*) yields more favorable performance, in terms of solution accuracy, than EPSDE on 7 out of the 30 test functions. On the contrary, the EPSDE variant is dominated by the original EPSDE on 21 test functions. Furthermore, although EPSDE/(*Settings*) attains more accurate solutions on the 7 test functions than EPSDE, none of the results is outperforms than that of MRDE.

From the comparison results between EPSDE/(*Settings*) and EPSDE we can see that the mutation operators and control parameters applied in MRDE cannot offer more promising performance with the EPSDE framework. In other works, the EPSDE framework does not take full advantages of distinct characteristics of the various mutation operator and control parameters involved in MRDE. Furthermore, the comparison results between MRDE and EPSDE/(*Settings*) indicate that the MRDE framework is more suitable for the operators and parameters since it can efficiently utilize diverse characteristics of them, and then enhance the comprehensive performance of the algorithm.

6. Conclusion

Inspired by a cooperation phenomenon between different roles in society, a multi-role based DE (MRDE) was proposed in this literature aiming to take full advantages of different trial vector generation strategies and control parameters in DE. In MRDE, the entire population is divided into multiple small-sized groups, and individuals in each group can be assigned with different roles according their fitness. Based on the assigned roles, different individuals select corresponding trial vector generation strategy and control parameters from a candidate pool, and then to breed offspring. In this case, different individuals in each generation can perform different search behaviors. Furthermore, a simple adaptive technique is used to change the population size in MRDE during the evolutionary process. Relying on the adaptive population size, the computational resources can be rational distributed in different evolutionary stages. In addition, the regroup operator conducted follow the change of population size enables an individual to play different roles in different evolutionary stages.

The experimental studies in this paper were carried out on CEC2017 test suite on 10D, 30D, and 50D cases. The comparison results between MRDE and other ten state-of-art DE variants suggested that MRDE yields very favorable performance. Although LSHADE_cnEpSin, which is one of the best performing algorithms from CEC2017 special session, yields the most performance in terms of solution accuracy, the Wilcoxon's test results suggested that there is no significant difference between LSHADE_cnEpSin and MRDE. Moreover, the efficiency and characteristics of new introduced strategies were experimentally studied. The experimental results indicated there are few merits of MRDE. The one is that it is helpful for MRDE to deal with various problems that assigning different trial vector generation strategies and control parameters to different individuals relying on their roles. Another one is that the adaptive population size is beneficial for speeding up convergence to different problems in various degree.

Although MRDE obtains a competitive performance testified by the extensive experiments, there are two problems need to be further studied in our future works. The one is that the efficiency and effectiveness of different trial vector generation strategies and control parameters applied in MRDE need to in-depth analysis based on different benchmark functions, although preliminary discussion has been proposed in this paper. When and how to change population size is another challenge because it is highly related to the characteristics of fitness landscapes.

Acknowledgment

This study was funded by the National Natural Science Foundation of China (Grant Nos.: 61663009, 61602174, 61762036, 61876136, 61806024), the National Natural Science Foundation of Jiangxi Province (Grant Nos.: 20171BAB202012, 20171BAB202019, 20161BAB202064, 20161BAB212052), and the Research Project of Jiangxi Provincial Department of Communication and Transportation (No. 2017D0038).

Appendix. Overall evaluation criteria

The overall evaluation method for each algorithm is based on a score of 100 which is based on two criteria as follows taking into account higher weights will be given for higher dimensions:

1) The one criterion is 50% summation of error (*SE*) values for all dimensions, which is defined as follow:

$$SE = 0.2 \times \sum_{i=1}^{30} f_{10D} + 0.3 \times \sum_{i=1}^{30} f_{30D} + 0.5 \times \sum_{i=1}^{30} f_{50D}$$

where f is the error values for all the functions and SE is the sum of errors, based on which the score for this part as follows.

$$Score1 = (1 - \frac{SE - SE_{min}}{SE}) \times 50$$

where SE_{min} is the minimal sum of errors from all the algorithms.

2) Another one criterion is 50% summation of rank (*SR*) results for each problem in each dimension, which is defined as follow:

$$SR = 0.2 \times \sum_{i=1}^{30} rank_{10D} + 0.3 \times \sum_{i=1}^{30} rank_{30D} + 0.5 \times \sum_{i=1}^{30} rank_{50D}$$

where SR is the sum of ranks, and then the score for this part as follows:

$$Score2 = (1 - \frac{SR - SR_{min}}{SE}) \times 50$$

where SR_{min} is the minimal sum of ranks from all the algorithms.

Then, we combine the above two parts to find the final score, i.e., the overall performance, of each algorithm as follows:

$$Score = Score1 + Score2$$

References

[1] 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.

[2] M. Srinivas, L. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Trans. Syst., Man, Cybern. 24 (4) (1994) 656–667.

[3] J. Jägersküpper, How the (1+1) ES using isotropic mutations minimizes positive definite quadratic forms, Theor. Comput. Sci. 361 (1) (2006) 38–56.

[4] J. Zhang, H. Chung, W.L. Lo, Clustering-based adaptive crossover and mutation probabilities for genetic algorithms, IEEE Trans. Evol. Comput. 11 (3) (2007) 326–335.

[5] G. Onwubolu, D. Davendra, Scheduling flow shops using differential evolution algorithm, Eur. J. Oper. Res. 171 (2) (2006) 674–692.

[6] Q.-K. Pan, L. Wang, L. Gao, et al., An effective hybrid discrete differential evolution algorithm for the flow shop scheduling with intermediate buffers, Inf. Sci. 181 (3) (2011) 668–685.

[7] K. Satoshi, M. Arakawa, K. Yamazaki, Differential evolution as the global optimization technique and its application to structural optimization, Appl. Soft Comput. 11 (4) (2011) 3792–3803.

[8] I. De Falco, A. DellaCioppa, D. Maisto, et al., Differential evolution as a viable tool for satellite image registration, Appl. Soft Comput. 8 (4) (2005) 1453–1462.

[9] I. Boussaïd, A. Chatterjee, P. Siarry, et al., Two-stage update biogeography-based optimization using differential evolution algorithm, Comput. Oper. Res. 38 (8) (2011) 1188–11198.

[10] P. Ghosh, S. Das, H. Zafar, Adaptive-differential-evolution-based design of two-channel quadrature mirror filter banks for sub-band coding and data transmission, IEEE Trans. Syst. Man Cybern. C Appl. Rev. 42 (6) (2012) 1613–1623.

[11] J. Zhang, A. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.

[12] E. Mezura-Montes, J. Velázquez-Reyes, C.A.C. Coello, A comparative study of differential evolution variants for global optimization, in: Proceedings of the 2006 Annual Conference on Genetic and Evolutionary Computation, 2006, pp. 485–492.

[13] R. Mendes, I. Rocha, E.C. Ferreira, et al., A comparison of algorithms for the optimization of fermentation processes, in: Proceedings of IEEE Congress on Evolutionary Computation, 2006, pp. 2018–2025.

[14] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution-an updated survey, Swarm Evol. Comput. 27 (2016) 1–30.

[15] V.L. Huang, A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for constrained real-parameter optimization, in: Proceedings of IEEE Congress on Evolutionary Computation, 2006, pp. 17–24.

[16] J. Brest, S. Greiner, B. Bosković, et al., Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657.

[17] G.Y. Yang, Z.Y. Dong, K.P. Wong, A modified differential evolution algorithm with fitness sharing for power system planning, IEEE Trans. Power Syst. 23 (2) (2008) 514–522.

[18] Z.Z. Liu, Y. Wang, S.X. Yang, et al., An adaptive framework to tune the coordinate systems in nature-inspired optimization algorithms, IEEE Trans. Cyber. (2018), https://doi.org/10.1109/TCYB.2018.2802912.

[19] 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.

[20] Y. Wang, Z.X. Cai, Q.F. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. Evol. Comput. 15 (1) (2011) 55–67.

[21] X.G. Zhou, G.J. Zhang, Abstract convex underestimation assisted multistage differential evolution, IEEE Trans. Cyber. 47 (9) (2017) 2730–2741.

[22] 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.

[23] D.K. Tasoulis, N.G. Pavlidis, V.P. Plagianakos, et al., Parallel differential evolution, in: Proceedings of IEEE Congress on Evolutionary Computation, 2004, pp. 142–151.

[24] M.F. Tasgetiren, P.N. Suganthan, A multi-populated differential evolution algorithm for solving constrained optimization problem, in: Proceedings of IEEE Congress on Evolutionary Computation, 2006, pp. 33–40.

[25] M.N. Omidvar, X. Li, Y. Mei, et al., Cooperative co-evolution with differential grouping for large scale optimization, IEEE Trans. Evol. Comput. 18 (3) (2014) 378–393.

[26] 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. 33 (C) (2015) 304–327.

[27] S.M. Guo, C.C. Yang, Enhancing differential evolution utilizing eigenvector-based crossover operator, IEEE Trans. Evol. Comput. 19 (1) (2015) 31–49.

[28] B.Y. Qu, P.N. Suganthan, J.J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, IEEE Trans. Evol. Comput. 16 (5) (2012) 601–614.

[29] K. Price, R. Storn, Differential Evolution: a Practical Approach to Global Optimization, Springer-Verlag, New York, 2005.

[30] R. Gämperle, S.D. Muller, P. Koumoutsakos, A parameter study for differential evolution, in: Proceedings of Advances Intelligent Systerms Fuzzy Systerms, 2002, pp. 293–298.

[31] J. Ronkkonen, S. Kukkonen, K.V. Price, Real parameter optimization with differential evolution, in: Proceedings of IEEE Congress on Evolutionary Computation, 2005, pp. 506–513.

[32] D. Zaharie, Control of population diversity and adaption in differential evolution algorithms, in: Proceedings of Mendel 2003, Ninth International Conference on Soft Computing, 2003, pp. 41–46.

[33] S. Das, A. Konar, U.K. Chakraborty, Two improved differential evolution schemes for faster global search, in: Proceedings of the 2005 Annual Conference on Genetic and Evolutionary Computation, 2005, pp. 991–998.

[34] Y. Wang, Z.X. Cai, Q.F. Zhang, Enhancing the search ability of differential evolution through orthogonal crossover, Inf. Sci. 185 (1) (2012) 153–177.

[35] J. Liu, J. Lampinen, A fuzzy adaptive differential evolution algorithm, Soft Comput. - A Fusion Found. Methodol. Appl. 9 (6) (2005) 448–462.

[36] Z.Z. Liu, Y. Wang, S.X. Yang, et al., Differential evolution with a two-stage optimization mechanism for numerical optimization, in: Proceedings of IEEE Congress on Evolutionary Computation, 2016, pp. 3170–3177.

[37] G.H. Wu, X. Shen, H.F. Li, et al., Ensemble of differential evolution variants, Inf. Sci. 423 (2018) 172–186.

[38] S.Z. Zhao, J.J. Liang, P.N. Suganthan, M.F. Tasgetiren, Dynamic multiswarm particle optimizer with local search for large scale global optimization, in: Proceedings of IEEE Congress on Evolutionary Computation, 2008, pp. 3845–3852.

[39] X.W. Xia, L. Gui, Z.H. Zhan, A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting, Appl. Soft Comput. 67 (2018) 126–140.

[40] M. Weber, F. Neri, V. Tirronen, Distributed differential evolution with explorative-exploitative population families, Genet. Program. Evolvable Mach. 10 (4) (2009) 343–471.

[41] M. Weber, V. Tirronen, F. Neri, Scale Factor Inheritance Mechanism in Distributed Differential Evolution, Springer-Verlag, 2010, pp. 1187–1297.

[42] S. Das, A. Abraham, U.K. Chakraborty, et al., Differential evolution using a neighborhood-based mutation operator, IEEE Trans. Evol. Comput. 13 (3) (2009) 526–553.

[43] U.K. Chakraborty, S. Das, A. Konar, Differential evolution with local neighborhood, in: Proceedings of IEEE Congress on Evolutionary Computation, 2006, pp. 2042–2049.

[44] S.M. Elsayed, R.A. Sarker, D.L. Essam, Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems, in: Proceedings of IEEE Congrress on Evolutionary Computation, 2011, pp. 1041–1048.

[45] G.H. Wu, R. Mallipeddi, P.N. Suganthan, et al., Differential evolution with multi population based ensemble of mutation strategies, Inf. Sci. 329 (C) (2016) 329–345.

[46] A.W. Mohamed, P.N. Suganthan, Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation, Soft Comput. 22 (10) (2018) 3215–3235.

[47] S. Ghosh, S. Das, S. Roy, et al., A differential covariance matrix adaptation evolutionary algorithm for real parameter optimization, Inf. Sci. 182 (1) (2012) 199–219.

[48] Y. Wang, Z.Z. Liu, J.B. Li, et al., On the selection of solutions for mutation in differential evolution, Front. Comput. Sci. 12 (2) (2018) 297–315.

[49] A.W. Iorio, X.D. Li, Solving rotated multi-objective optimization problems using differential evolution, in: Proceedings of Australian Conference on Artificial Intelligence, 2004, pp. 861–872.

[50] G. Iacca, R. Mallipeddi, E. Mininno, et al., Super-fit and population size reduction in compact differential evolution, in: Proceedings of IEEE Symposium Series on Computational Intelligence-MC: IEEE Workshop on Memetic Computing, 2011, pp. 21–28.

[51] A. Zamuda, J. Brest, Population reduction differential evolution with multiple mutation strategies in real world industry challenges, in: Proceedings of IEEE Symposium on Swarm and Evolutionary Computation, 2012, pp. 154–161.

[52] M. Yang, Z. Cai, C. Li, et al., An improved adaptive differential evolution algorithm with population adaptation, in: Proceedings of the 2013 Annual Conference on Genetic and Evolutionary Computation, 2013, pp. 145–152.

[53] N.H. Awad, M.Z. Ali, J.J. Liang, et al., Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Nanyang Technological Univ., Singapore, 2016. Tech. Rep.

[54] Y. Wang, H.-X. Li, T. Huang, et al., Differential evolution based on covariance matrix learning and bimodal distribution parameter setting, Appl. Soft Comput. 18 (1) (2014) 232–247.

[55] N.H. Awad, M.Z. Ali, P.N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems, in: Proceedings of IEEE Congrress on Evolutionary Computation, 2017, pp. 372–379.