Differential evolution with dynamic combination based mutation operator and two-level parameter adaptation strategy

Deng, Libao; Li, Chunlei; Lan, Yanfei; Sun, Gaoji; Shang, Changjing

Published in:
Expert Systems with Applications

DOI:
10.1016/j.eswa.2021.116298

Publication date:
2022

Citation for published version (APA):
Deng, L., Li, C., Lan, Y., Sun, G., & Shang, C. (2022). Differential evolution with dynamic combination based mutation operator and two-level parameter adaptation strategy. *Expert Systems with Applications*, *192*, Article 116298. https://doi.org/10.1016/j.eswa.2021.116298

Differential Evolution with Dynamic Combination based Mutation Operator and Two-level Parameter Adaptation Strategy

Libao Deng, Chunlei Li

School of Information Science and Engineering, Harbin Institute of Technology, Weihai, China.

Yanfei Lan

College of Management & Economics, Tianjin University, Tianjin, China.

Gaoji Sun*

College of Economic and Management, Zhejiang Normal University, Jinhua, China.

Changjing Shang

Department of Computer Science, Aberystwyth University, Wales, U.K.

*Corresponding author

Email addresses: denglibao_paper@163.com (Libao Deng), lcl980826@163.com (Chunlei Li), lanyf@tju.edu.cn (Yanfei Lan), gsun@zjnu.edu.cn (Gaoji Sun), cns@aber.ac.uk (Changjing Shang)

Abstract

Differential evolution (DE) is a simple yet effective algorithm for numerical optimization, and its performance significantly depends on mutation operator and control parameters. Therefore, designing appropriate mutation operator and parameter regulation strategy is an important and necessary task. To improve the performance of DE algorithm, we propose a novel DE variant called DCDE based on a dynamic combination based mutation operator and a two-level parameter regulation strategy. More specifically, the newly proposed mutation operator contains a dynamic base vector that consists of two individuals, one is the current optimal individual while the other, called elite individual, is the best one among three randomly selected individuals, and they are dynamically combined by a weight parameter associated with the evolution process and the ranking status of the elite individual in the current population. Moreover, the scale factor and crossover rate in DCDE depend on the combined effect of a population-level parameter and one individual-level parameter, respectively. Both mutation operator and control parameters in DCDE are designed to achieve an appropriate balance between global exploration ability and local exploitation ability. To evaluate the performance of DCDE, comparison experiments are conducted with five state-of-the-art DE variants and three non-DE algorithms on solving 29 functions in IEEE CEC 2017 benchmark suite. The comparison results indicate that the proposed DCDE is significantly better than, or at least comparable to the adopted competitors.

Keywords: Differential evolution, Mutation operator, Two-level Parameter, Numerical optimization.

1. Introduction

Differential evolution (DE), first proposed by Storn and Price (Storn & Price, 1997), is a very effective evolutionary algorithm (EA). DE has exhibited outstanding performance due to its simple structure, rapid convergence speed as well as strong robustness and it has been successfully applied in various domains of science and engineering, such as signal processing (Hancer et al., 2018; Zhou et al., 2019; Hancer, 2019), neural network (Duchanoy et al., 2017; Arce et al., 2018), power system (Sakr et al., 2017; Biswas et al., 2017; Reddy, 2019), clustering analysis (Dong et al., 2014; Alswaitti et al., 2019), scheduling problems (Zhu et al., 2017; Liang et al., 2019) and many other practical optimization problems.

Similar to other evolutionary algorithms, DE continuously repeats three basic operators, i.e., mutation, crossover and selection, to evolve its solution toward to the global optima until the stopping criteria is met. Its performance is not only sensitive to the trial vector generation strategy (mutation and crossover), but also the associated control parameters including the population size NP, scale factor F and crossover rate CR. Inappropriate descendant generation strategy or parameter settings may directly impact the convergence speed and searching precision. Therefore, it is essential to adopt an appropriate strategy for a specific optimization task. However, it is a time-consuming trial-and-error process, since we cannot find a best suitable parameter setting and descendant generation strategy for all cases when solving problems with diverse characteristics according to the NFL theorem (Adam et al., 2019). Consequently, many researchers are devoted to put forward different adaptive or self-adaptive mechanisms to adjust the control parameters and tune mutation strategies during the evolution process. For instance, Zhang and Sanderson (Zhang & Sanderson, 2009) proposed an adaptive algorithm (JADE) characterized by a novel greedy mutation strategy, namely "DE/Current-to-*p*best" with an optional external archive. Sun et al. (Sun et al., 2018) introduced a novel mutation operator called DE/better/2, which uses the best one among five randomly selected individuals as the base vector. Meanwhile, it combines a sinusoidal function with the individual's rank information to adjust the parameters F and CR. Li et al. (Li et al., 2017) integrated the clearing niche mechanism with the existing mutation strategy to enhance the population diversity and searching ability. Additionally, many multi-mutation strategies were also proposed to improve the performance of DE. MPEDE (Wu et al., 2016b) implemented a dynamic ensemble of three mutation strategies with different characteristics. On the basis of MPEDE, Wu et al. (Wu et al., 2018) introduced a variant of MPEDE named EDEV, which attempted to realize a high-level ensemble of multiple DE variants including JADE (Zhang & Sanderson, 2009), EPSDE (Mallipeddi et al., 2011) and CoDE (Wang et al., 2011). Sun et al. (Sun et al., 2020b) proposed two novel mutation strategies with different search feature and combined a decreasing function with a periodic function to adjust the size of the elite population.

The superior performance of the aforementioned DE variants proved that adaptive or self-adaptive mechanisms can significantly improve the optimization ability of DE. As pointed out in (Črepinšek et al., 2013), the balance between global exploration ability and local exploitation ability has a significant impact on the performance of the evolutionary algorithms. However, maintaining a proper tradeoff between these two abilities is an intractable problem, especially for the higher dimensional problems (Jian et al., 2020). In fact, both the trail vector generation strategies and control parameters directly affect the exploration ability and exploitation ability of algorithm, so it is important to design appropriate trail vector generation strategy and parameter control strategy and exploit the synergy between the two in balancing global exploration work and local exploitation work.

In view of above considerations, we propose a new DE variant (DCDE) with a dynamic combination based mutation operator and two-level parameter control strategy to balance the exploration ability and exploitation ability. In fact, the base vector in the mutation operator determines the current searching center, and the region around the individuals with good fitness values is often a potential area that needs to be developed. At the same time, the difference vectors have an great influence on the population diversity. For an effective DE algorithm, the population diversity should be maintained in the early stage of the evolutionary process, and the search work of potential areas should be emphasized in the later stage. To achieve this purpose, we combine the optimal individual and an elite individual which performs the best among three randomly selected individuals to construct a dynamic base vector. The combination is realized by a dynamic weight parameter which is determined by not only the evolution process but also the ranking status of the elite individual. What's more, the values of parameters F and CR in DCDE are adapted based on a global parameter from perspective of the whole population and a individual-dependent parameter obtained by the ranking status of the corresponding individual. To verify the effectiveness of the proposed DCDE, it's compared with five state-of-the-art DE variants and three non-DE algorithms. Extensive experiments are carried based on $30D$, $50D$ and $100D$ benchmark functions from IEEE CEC 2017 test suite. The results indicate that the proposed DCDE algorithm is superior or competitive to all the compared algorithms. Summarily, the main contributions of this paper can be concluded in the following four aspects:

(1) In order to fully exploit the role of the base vector in the mutation operator during the search process, we construct a new base vector using the dynamic combination between the optimal individual and the best one among the three randomly selected individuals, and then design a novel mutation operator;

(2) In order to fully exploit the role of the two core parameters F and CR in balancing the global exploration ability and the local exploitation ability, we design a novel adaptive strategy for them through the joint action of a population-level macro-parameter and an individual-level micro-parameter;

(3) In order to achieve the best performance of the designed algorithm, we design the mutation operator and the adaptive strategy for the core parameters F and CR with full consideration of the synergy effect between them in balancing the global exploration ability and the local exploitation ability;

(4) Comparison experiments with eight outstanding competitors in a public test platform (IEEE CEC 2017) show that our proposed DCDE algorithm has the best overall performance.

The remainder of this paper is arranged as follows: Section 2 introduces the basic DE algorithm. Section 3 briefly reviews some advanced methods related to our research. The proposed DCDE algorithm is described explicitly in Section 4. In Section 5, the experimental results are presented and analysed. At last, Section 6 provides the conclusion of our research.

2. Differential Evolution

The classic differential evolution algorithm consists of four basic operators: initialization, mutation, crossover and selection, which will be briefly reviewed in this section.

The global minimization task of D dimensions is considered in this paper and can be defined as follows:

$$f(\mathbf{x}^*) = min\, f(\mathbf{x}_i), \mathbf{x}_i \in [\mathbf{x}_{min}, \mathbf{x}_{max}] \tag{1}$$

where $f(\cdot)$ denotes the objective function, and $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,D})$ is named as the target vector, $\mathbf{x}^*$ is the global optima of the optimization task. Usually, the target vectors will be restricted in the searching space by a pre-defined lower bound $\mathbf{x}_{min} = (x_{min,1}, x_{min,2}, \cdots, x_{min,D})$ and a upper bound $\mathbf{x}_{max} = (x_{max,1}, x_{max,2}, \cdots, x_{max,D})$.

2.1. Initialization operator

To begin the optimization process, an initial population consisting of NP individuals $\mathbf{P}^0 = (\mathbf{x}_1^0, \mathbf{x}_2^0, \cdots, \mathbf{x}_{NP}^0)$ needs to be created by uniformly randomizing these individuals through the following formula:

$$x_{i,j} = x_{min,j} + rand[0,1] \cdot (x_{max,j} - x_{min,j}) \tag{2}$$

where $rand[0, 1]$ denotes a uniformly distributed random number within the interval [0,1].

2.2. Mutation operator

At each iteration, the mutation operator will be employed for each target vector to produce the corresponding mutant vector $\mathbf{v}_i^g = (v_{i,1}^g, v_{i,2}^g, \cdots, v_{i,D}^g)$. The five most frequently used mutation strategies are listed below, and more detailed information of them can be found in Ref. (Das & Suganthan, 2011).

- DE/Rand/1:

$$\mathbf{v}_i^g = \mathbf{x}_{r_1}^g + F \cdot (\mathbf{x}_{r_2}^g - \mathbf{x}_{r_3}^g) \tag{3}$$

- DE/Best/1:

$$\mathbf{v}_i^g = \mathbf{x}_{best}^g + F \cdot (\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) \tag{4}$$

- DE/Current-to-best/1:

$$\mathbf{v}_i^g = \mathbf{x}_i^g + F \cdot (\mathbf{x}_{best}^g - \mathbf{x}_i^g) + F \cdot (\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) \tag{5}$$

- DE/Rand/2:

$$\mathbf{v}_i^g = \mathbf{x}_{r_1}^g + F \cdot (\mathbf{x}_{r_2}^g - \mathbf{x}_{r_3}^g) + F \cdot (\mathbf{x}_{r_4}^g - \mathbf{x}_{r_5}^g) \tag{6}$$

- DE/Best/2:

$$\mathbf{v}_i^g = \mathbf{x}_{best}^g + F \cdot (\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) + F \cdot (\mathbf{x}_{r_3}^g - \mathbf{x}_{r_4}^g) \tag{7}$$

In above equations, the indices r_1, r_2, r_3, r_4 and $r_5 \in \{1, 2, \cdots, NP\}$ are mutually exclusive integers and are different from the index i. The vector $\mathbf{x}_{best}^g$ represents the optimal individual in the population at the gth generation. The parameter F is called the scale factor, which is a positive constant within the range [0,1].

2.3. Crossover operator

The crossover operator is implemented on each pair of target vector $\mathbf{x}_i^g$ and corresponding mutant vector $\mathbf{v}_i^g$ to generate a trial vector $\mathbf{u}_i^g = (u_{i,1}^g, u_{i,2}^g, \cdots, u_{i,D}^g)$. The binomial crossover which is commonly utilized can be defined as follows:

$$u_{i,j}^g = \begin{cases} v_{i,j}^g & \text{if } rand[0,1] \leqslant CR \text{ or } j = j_{rand} \\ x_{i,j}^g & \text{otherwise} \end{cases} \tag{8}$$

where $CR \in [0, 1]$ is named as the crossover rate, and the index $j_{rand} \in \{1, 2, \cdots, D\}$ is a randomly selected integer to ensure that at least one dimension of trial vector is inherited from mutant vector.

2.4. Selection operator

In this step, DE applies a one-to-one greedy mechanism to choose a better one from the target vector $\mathbf{x}_i^g$ and trial vector $\mathbf{u}_i^g$ to survive into the next generation. For a minimization problem, this operator can be defined as follows:

$$\mathbf{x}_i^{g+1} = \begin{cases} \mathbf{u}_i^g & \text{if } f(\mathbf{u}_i^g) \leqslant f(\mathbf{x}_i^g) \\ \mathbf{x}_i^g & \text{otherwise} \end{cases} \tag{9}$$

3. Related Works

Recent two decades have witnessed the significant development of DE, and many variants have been proposed by researchers to enhance DE's optimization ability. In this section, we will only briefly review some modified methods on the trial vector generation strategy and the control parameter adjustment which are relevant to our work. For a comprehensive up-to-date survey on DE, please refer to Refs. (Das et al., 2016) and (Opara & Arabas, 2019).

The generation of trial vector is one of the most critical steps of DE, and has attracted the attention of many scholars to propose various improvement variants. Zhang and Sanderson (Zhang & Sanderson, 2009) introduced a classical and effective DE variant JADE, which executed a novel mutation operator "DE/Current-to-*p*best" with an optional external archive providing direction information. Tanabe and Fukunaga (Tanabe & Fukunaga, 2013) proposed a new variant of JADE, namly SHADE, which applied archives techniques based on the historical memory to store a set of well-performed scale factors and crossover rates. Tanabe and Fukunaga (Tanabe & Fukunaga, 2014) further introduced an improved version called LSHADE by applying a linear reduction scheme for population size. Mohamed et al. (Mohamed et al., 2019) proposed two novel mutation operators including a more greedy strategy "DE/Current-to-ord_*p*best" and a less greedy strategy "DE/Current-to-ord_best" and incorporated them into SHADE and LSHADE to enhance their performance. Islam et al. (Islam et al., 2012) developed a new mutation operator namely "DE/Current-to-gr_best/1" and a novel crossover operator to surmount the premature convergence and stagnation problems in the canonical DE. Otani et al. (Otani et al., 2013) introduced a "DE/isolated/1" mutation operator to generates new individuals near an isolated individual which would be helpful to evenly allocate the search resources. Sallam et al. (Sallam et al., 2017) established an adaptive operator selection mechanism, which measured the landscape and the historical performance to select appropriate DE mutation strategies from a pool of operators. In order to improve the global and local search abilities, as well as the convergence speed of the basic DE, Mohamed (Mohamed, 2017) presented an efficient modified DE with applying a new triangular mutation operator, which was on the bias of a convex combination vector.

Selecting appropriate control parameters is an indispensable task and plays a pivotal role for enhancing the performance of DE. Therefore, various adjustment methods have been introduced by the researchers to lessen the sensitivity of DE to the control parameters when solving different optimization tasks. For instance, Yu et al. (Yu et al., 2013) implemented a two-level adaptive parameters control mechanism, in which the population-level parameters were adaptively adjusted based on the optimization state and the individual-level parameters were generated in accordance with fitness values and its distance from the optimal individual. Tang et al. (Tang et al., 2014) introduced a new DE variant which embedded an individual-dependent scheme to determine the control parameter values. Draa et al. (Draa et al., 2015) proposed a sinusoidal differential evolution, which was based on two preset sinusoidal formulas to automatically adjust F and CR. What's more, Draa et al. (Draa et al., 2019) proposed a new variant of SinDE, namely OCSinDE, which utilized a compound sine-based formula to adjust F and CR. Compared with original SinDE, OCSinDE could make parameters' variation less monotone. Tatsis et al. (Tatsis & Parsopoulos, 2017) proposed a grid-based parameter adaptation scheme to dynamically select the most appropriate crossover type and parameter values, and the core technique of the scheme lies in the discrete parameter searching space. More recently, Meng et al. (Meng et al., 2018) established a DE variant called PALM-DE to generate new parameters with an adaptive learning mechanism for the inconvenience in selecting control parameters. However, this variant was heavily relied on the number of individuals to generate a suitable value for CR. Thus, on the basis of PALM-DE, Meng et al. (Meng et al., 2019) introduced a new parameter adaptive DE variant called PaDE to resolve inappropriate adaptation schemes with a novel grouping strategy. In order to provide explicit guidelines for generating appropriate control parameters, Wang et al. (Wang et al., 2019) proposed a parameter adaption strategy based on Association Rule Mining methodology, which could extract precise effectual F and CR couple associations during different evolving phases. Sun et al. (Sun et al., 2020a) constructed an adaptive parameter settings using decreasing and periodic functions for balancing the global exploration ability and the local exploitation ability.

Great efforts have also been made to achieve an ideal balance between the global exploration and the local exploitation for designing an effective DE variant. Li and Yin (Li & Yin, 2016) presented a modified DE by combing two mutation rules(i.e., DE/Rand/2 and DE/current-to-*p*best (Zhang & Sanderson, 2009)) by an alternate probability rule to balance the exploitation and exploration of the algorithm. Liu et al. (Liu et al., 2016) introduced a new DE on the bias of a two-stage mechanism, called TSDE, which divided the whole evolving process into two stages, and applied different trial vector generation strategies in these two stages. Bhattacharya and Chattopadhyay (Bhattacharya & Chattopadhyay, 2010) integrated DE with the biogeography-based optimization algorithm (BBO) to make use of the exploration ability of DE and the exploitation ability of BBO. Similarly, Liu et al. (Liu et al., 2019) presented a hierarchical DE with implementing multi-cross operation in the top layer which is featured with fast convergence speed and superior local exploitation ability while executing SHADE (Tanabe & Fukunaga, 2013) in the bottom layer to make use of its global exploration ability. Sun et al. (Sun et al., 2021) designed two regeneration strategies based on the dynamic adjustment of the search space focusing on global exploration and local exploitation respectively to improve the optimization performance of the DE algorithm. Zhao et al. (Zhao et al., 2018) introduced a new DE variant by integrating DE with the gravitational search algorithm. In this variant, two mutation strategies and a novel disruption scheme on the bias of Levy flight were employed to strengthen the local exploitation ability, meanwhile a new self-adaptive tuning mechanism was designed to balance the exploration and exploitation.

4. Description of DCDE

In this part, we will provide a detail description of the proposed DCDE algorithm. Firstly, we will present our designing motivation of DCDE, and then introduce the adopted mutation operator based on the dynamic combination scheme and the two-level adaptation strategy for scale factor F and crossover rate CR. Finally, the overall operation process of DCDE will be introduced.

4.1. Motivation

For DE algorithms, the mutation operator plays the vital role in its searching ability. A superior mutation operator should maintain a proper balance between the global exploration ability and the local exploitation ability during different stages of the evolution process. Generally, the mutation operator is composed of base vector and difference vector, where the base vector determines the reference point of searching while the difference vector is utilized to provide the searching direction and perturb base vector to find a new position. As pointed out in (Khaparde et al., 2015), the information of base vectors is utilized for cluster formation, which further has an influence on the final solutions. For instance, mutation strategy DE/Rand/1 (i.e. Eq. (3)) takes a random individual as the base vector which can enrich the population diversity and maintain the outstanding global exploration ability, but it is highly likely to lead to stagnation problems. Mutation strategy DE/Rand/2 (i.e. Eq. (6)) is a variant of DE/Rand/1 by adding an extra difference vector, which is expected to have a better perturbation to the base vector (Mohamed et al., 2019). Mutation strategy DE/Best/1 (i.e. Eq. (4)) is a greedy scheme by choosing the best individual as the base vector, which can enhance exploitation ability and increase the convergence speed, but it is against the population diversity and is easy to cause premature convergence problem (Li et al., 2017). It can be seen that suitable base vector can effectively improve the ability of mutation strategy in balancing global exploration and local exploitation, which in turn improves the optimization performance of the DE algorithm.

4.2. Dynamic combination based mutation operator

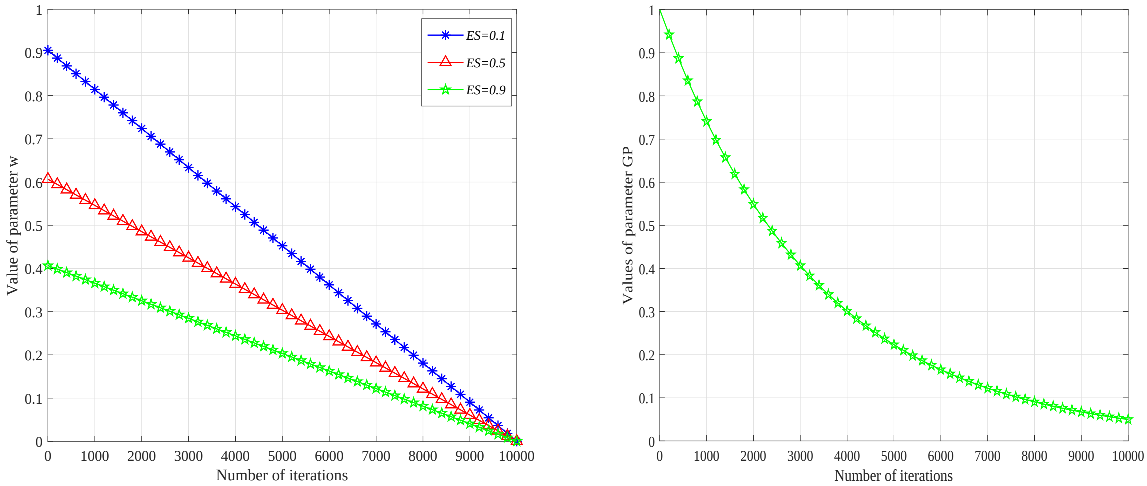
According to the above considerations, we propose a novel mutation operator in which the base vector is constructed from a dynamic combination of the optimal individual and the best one (named elite individual) among three randomly selected individuals. In order to more fully exploit the function of the base vector, we try to incorporate the fitness information of individuals into the combinatorial control parameters, i.e., we need to sort the populations in ascending order according to the fitness values of individuals. Then, the mutation operation will be executed and can be expressed by the following formula:

$$\mathbf{v}_i^g = w \cdot \mathbf{x}_{r^*}^g + (1-w) \cdot \mathbf{x}_{best}^g + F_1 \cdot (\mathbf{x}_{r_1}^g - \mathbf{x}_i^g) + F_2 \cdot (\mathbf{x}_{r_2}^g - \mathbf{x}_i^g) \tag{10}$$

where the indexes r^*, r_1 and r_2 are mutually exclusive integers randomly selected from the set $\{1, 2, \cdots, NP\}$ and are different form the index i. What's more, $\mathbf{x}_{r^*}^g$ is called the elite individual in DCDE, which is the best one among the three randomly selected individuals, while $\mathbf{x}_{best}^g$ is the currently best individual in the population. Parameters F_1 and F_2 are two mutually independent scale factors, and we will describe them in detail later. The variable w is a weight parameter and can be calculated as follows:

$$w = \frac{\delta}{\exp(m \cdot ES)} \tag{11}$$

where $\delta = (G_{max} - g + 1)/G_{max}$ is a process parameter relied to the evolutionary process, where G_{max} is the maximum allowed number of iterations, and g is the index of the current iteration. The parameter $ES = r^*/NP$ indicates the current ranking status of the selected elite individual $\mathbf{x}_{r^*}^g$. It is easy to see that the better the ranking of elite individuals, the smaller the parameter value ES. The constant m is applied to control the influence of individual-level parameter ES on weight parameter w, and it's set to 1 in DCDE, which will be further analyzed in the parameter analysis section.



(a) (b)

Fig. 1. The changing process of weight parameter w (a) and macro parameter GP (b).

From the Eq. (10), it's evident to see that the new base vector includes two individuals, i.e., $\mathbf{x}_{r^*}^g$ and $\mathbf{x}_{best}^g$, which are collaborated by one dynamic combination scheme. Obviously, parameter w is a key control variable in the combination strategy, which is used to regulate the influence degree of the random-based elite individual $\mathbf{x}_{r^*}^g$ and the optimal individual $\mathbf{x}_{best}^g$ in the mutation operator respectively. In order to present the combination strategy more intuitively, the changing process of parameter w along with the the iteration for three elite individuals with different ranking status is plotted in Fig. 1 (a). As we can see in this figure, the value of w decreases linearly along with the evolution process while it increases when the ranking status of the selected elite individual gets better, which is reflected in the smaller value of the parameter ES. In the early stage, a lager value of w implies that the elite individual $\mathbf{x}_{r^*}^g$ plays a relatively large role, which means that the population need to maintain a relatively high level of diversity. It should be specified that the use of elite individual in the base vector can provide a more promising region for the search work and effectively improve the search quality. In the latter stage, the focus is transferred to the exploitation work by making the optimal individual $\mathbf{x}_{best}^{g}$ plays a greater role. In addition, regions around elite individuals with better fitness values are more worthy of deeper development and therefore need to be assigned greater weights, and the individual-level parameter ES is applied to achieve this consideration. What's more, the use of two difference vectors can enhance the diversity of populations. Finally, it should be note that if the component of mutant vector generated by Eq. (10) exceeds the limits of the searching space, it will be set to the middle of selected elite individual $\mathbf{x}_{r^*}$ and the parent vector $\mathbf{x}_i$. In summary, the new mutation operator based on the dynamic combination scheme has fully considered the function of base vector and difference vector, thus it is expected to can effectively regulate the global exploration ability and local exploitation ability in the optimization process.

4.3. Two-level parameter regulation strategy

The control parameters also paly an important role in the success of DE. The scale factor F is utilized to determine the searching radius centred on the base vector, and the crossover rate CR controls the components inherited from mutant vector. As pointed out in (Tang et al., 2014), F should be arranged a smaller value for better base vectors to exploit around the base vectors with a small radius; a relatively large value of F is more suitable for inferior base vectors to help them out of the local areas and explore more promising areas. Similarly, CR should be a smaller value for a superior parent vector to inherit more valuable information to the offspring. On the contrary, if the parent vector does not perform well, the offspring should inherit more information from the trial vector for improving their own genes. It is worth mentioning that the probability distribution model is attracting more and more attention on solving the optimization problems, and the Gaussian distribution is one of the most commonly used models (Zhang & Sanderson, 2007; Mallipeddi et al., 2014; Jena et al., 2016; Sun et al., 2019; Zhang et al., 2020). According to the study of Ref. (Omran et al., 2005), the mutation operator perturbed by Gaussian distribution has less chance to lead the population to the local optimum.

Based on the above analysis, a population-level parameter, two types of individual-level parameters and a Gaussian distribution model are employed to modulate the core parameters F and CR during the evolution process, where the population-level parameter GP is utilized to control the trend of involved parameters from the perspective of the whole population, and it can be defined as follows:

$$GP = \exp[n \cdot (\delta - 1)] \tag{12}$$

The changing process of GP is depicted in Fig. 1 (b) when $n = 3$. Obviously, the value of GP decreases exponentially along with the evolution process, and its descent speed is determined by the parameter n. In our proposed DCDE variant, the value of n is set to 3, and the reason for this choice will be illustrated in the parameter analysis part. Moreover, the parameter F is associated with ranking status of the elite individual while CR is influenced by that of target vector $\mathbf{x}_i$. In fact, this setting is derived from the functions of the parameters F and CR themselves, i.e., F is used to regulate the search radius centered on the base vector, while CR is used to control the proportion of information inherited from the current individual to the next generation. The calculation formula of them are expressed as follows:

$$F = GP \cdot N(\mu, \sigma^2) + (1 - GP) \cdot ES \tag{13}$$

$$CR = GP \cdot N(\mu, \sigma^2) + (1 - GP) \cdot IS \tag{14}$$

where $N(\mu, \sigma^2)$ returns a random value which is based on the Gaussian distribution with $\mu = 0.5$ and $\sigma = 0.1$. It should be noted that F_1 and F_2 were generated independently using Eq. (13), which could increase the diversity of the population to some extent. The parameter ES is the same as defined in Eq. (11), and $IS = i/NP$ is utilized to indicate the ranking status of parent vector $\mathbf{x}_i$ and it is seen that the better parent vector corresponds the smaller value of IS.

From the Eq. (13) and Eq. (14), we can observe that in the early stage, the random number generated by the Gaussian function has more intense impact on the two control parameters F and CR by arranging a larger value of GP, but in the latter phase, the ranking status parameters ES and IS will fully play their function to generate F and CR respectively. The mechanism behind this design is that in the early stage, individuals accumulate less information, and reducing the influence of individual information on the search work will reduce the probability of falling into the local optimum; while in the later stage, individuals gradually have more valuable information, and making full use of this information will help improve the search efficiency. In summary, the two-level control parameters regulation scheme can not only make use of evolutionary process information but also utilize the feedback information from different individuals and the scheme is also designed on the principle of achieving a suitable balance between the global exploration ability and the local exploitation ability.

The detailed description of the dynamic combination scheme for mutation operator and the two-level regulation scheme for control parameters in our proposed DCDE have been provided. Now, the complete operation flow of the proposed DCDE algorithm will be summarized in Algorithm 1, which shows that DCDE has a simple structure and it is easy to implement and operate.

5. Experimental Study

In this section, the computational results of the proposed DCDE on the adopted experimental platform as well as the comparisons results with eight state-of-the-art algorithms will be presented in detail. Additionally, the parametric sensitivity analysis experiment will also be provided to confirm the optimal parameter setting in DCDE.

5.1. Benchmark functions

To verify the performance of the proposed DCDE algorithm, extensive experiments are carried out based on 29 benchmark functions in IEEE CEC 2017 test suite, which can be divided into the following four categories with different characteristics: f_1, f_3: unimodal functions; $f_4 - f_{10}$: simple multimodal functions; $f_{11} - f_{20}$: hybrid functions; $f_{21} - f_{30}$: composition functions. It should be noted that function f_2 has been detached from the IEEE CEC 2017 test suite for its unstable characteristic especially for high-dimensional problem, and more detail information about these optimization functions can be found in Ref. (Wu et al., 2016a).

5.2. Comparison algorithms and parameter setting

In our experiment, five state-of-the-art DE variants including JADE (Zhang & Sanderson, 2009), SinDE (Draa et al., 2015), EFADE (Mohamed & Suganthan, 2018), AGDE (Mohamed & Mohamed, 2019), GODE (Zhao et al., 2021), and three recently proposed meta-heuristic algorithms including MPA (Faramarzi et al., 2020a), EO (Faramarzi et al., 2020b), SGLSCA (Zhou et al., 2021) are used for performance comparison. To show that the comparison results are convincing, parameters of all the compared algorithms are configured as recommended in the corresponding original paper. In DCDE, the population size NP is set to $2 \cdot D$, which will be discussed in the parameter analysis part. The comparison experiments are carried out on 30D, 50D, and 100D functions, and the maximum of function evaluations is set to 10,000 $\cdot D$, which is also used as the stopping criteria.

Algorithm 1: Algorithm Description of DCDE

Input: Maximum generations: G_{max}; Population size: NP;

Generate an initial population $\mathbf{P}^0$ by Eq. (2);
Find the initial optimal individual $\mathbf{x}_{best}$ and calculate its fitness $f(\mathbf{x}_{best})$;
for $(g = 1 : G_{max})$ **do**
    Reorder the whole population in an ascending order based on the fitness values;
    Execute $\delta = (G_{max} - g + 1)/G_{max}$;
    Compute the population-level parameter GP by Eq. (12);
    for $(i = 1 : NP)$ **do**
        Randomly select three individuals r^*, r_1 and r_2 from the set $\{1, 2, \cdots, NP\}$, and set the best one to 1;
        Set the individual-level parameter ES to r^*/NP;
        Set the individual-level parameter IS to i/NP;
        Compute the weight parameter w by Eq. (11);
        Compute the scale factors F_1 and F_2 independently by Eq. (13);
        Compute the crossover rate CR by Eq. (14);
        Randomly select an integer j_{rand} from the set $\{1, 2, \cdots, D\}$;
        for $(j = 1 : D)$ **do**
            if $(rand[0, 1] < CR \parallel j == j_{rand})$ **then**
                Compute $v_{i,j}$ by Eq. (10);
                Execute $u_{i,j} = v_{i,j}$;
                if $(u_{i,j} < x_{min,j} \parallel u_{i,j} > x_{max,j})$ **then**
                    $u_{i,j} = 0.5 \cdot (x_{r^*,j} + x_{i,j})$;
            else
                Execute $u_{i,j} = x_{i,j}$
        Execute selection operator by Eq. (9);
        if $(f(\boldsymbol{u}_i^g) < f(\boldsymbol{x}_{best}))$ **then**
            Replace $\mathbf{x}_{best}$ by $\mathbf{u}_i^g$;

Output: Optimal solution: $\mathbf{x}_{best}$; Optimal fitness value: $f(\mathbf{x}_{best})$.

5.3. *Performance metric*

Several evaluation criteria are employed in our experiment to evaluate the performance of DCDE and they can be divided into the following groups:

(1) *Function Error Value*: The function error value is utilized to reflect the effectiveness of test algorithms for searching the solution within limited evaluations and can be computed by $f(\mathbf{x}_{best}) - f(\mathbf{x}^*)$, where $f(\mathbf{x}_{best})$ denotes the optimal fitness value obtained by the end of optimization procedure, and $f(\mathbf{x}^*)$ is the fitness value of the actual global optimal solution. In our experiment, all the compared algorithms are conducted 50 independent runs to calculate the average value and standard deviation of the function error value, which are denoted by "Mean" and "Std.", respectively.

(2) *Statistical Test*: To gain a statistical conclusion of the comparison results, the single-problem analysis by Wilcoxon rank-sum test (Derrac et al., 2011) is conducted on the experimental results at the 0.05 significance level to distinguish whether there is significant difference between DCDE and each compared DE variant and the markers "$+/=/-$" denote that the performance of the DCDE is significantly better than, equal to, or inferior to each competitor, respectively. What's more, the multiple-problem analysis is conducted by both the Wilcoxon signed-rank test and the Friedman test at the 0.05 significance level. The difference between these two non-parametric test methods lies in the fact that Wilcoxon signed-rank test is a pairwise test aiming to distinguish the significant difference between the performance of only two algorithms while Friedman test (Derrac et al., 2011) is a multiple comparison method aiming to detect significant difference and calculate the average ranking between all the compared algorithms.

(3) *Convergence Graphics*: Convergence graphics are employed in our experiment to investigate the mean error performance of the best solution over the total runs and compare the convergence speed of each algorithm in the respective experiments.

5.4. Results and analysis

The comparison results of mean and standard deviation obtained by DCDE and other algorithms for solving 30D, 50D and 100D test functions are summarized in Tables 1-3. In these tables, the best results are highlighted in boldface to indicate the best performing algorithm for each problem.

For 30D test functions, DCDE performs best among the compared algorithms by obtaining results with minimum mean value for 10 functions which is followed by EFADE with getting the smallest mean value on 8 functions. For other algorithms, JADE, SinDE and AGDE perform best on 4 functions; the performance of EO, SGLSCA and MPA is relatively poor and they only obtain 1, 1, 0 best mean value respectively. From the statistical results based on single-problem Wilcoxon rank-sum test, we can find that DCDE provides significantly better results compared with both DE and non-DE variants in most functions. Specifically, for non-DE algorithms SGLSCA, MPA, EO, DCDE is significantly better on 29, 26, 26 functions respectively, and DCDE is only beaten by EO on f_4, and by MPA on f_{18}, f_{26}. Compared with all the selected DE variants JADE, SinDE, AGDE, EFADE and GODE, DCDE yields significantly better results on more than half of the functions.

Although it's more difficult to deal with higher dimensional problems, DCDE still shows much more superior performance compared with other algorithms on 50D test functions. From the perspective of minimum mean value, DCDE still remains the first-ranking position by yielding best results on 12 test functions, which is increased by 2 compared with that of 30D test functions. For other algorithms, the number of optimal results obtained by JADE, SinDE and EFADE is 4, and that obtained by MPA, AGDE, GODE, EO, SGLSCA is 3, 1, 1, 0, 0 respectively. What's more, according to the Wilcoxon rank-sum test for 50D experimental results, the number of significantly better results provided by DCDE remains the same or increases compared with JADE, SinDE, AGDE, EFADE, GODE and EO for solving 30D functions. And DCDE is only beaten by SGLSCA, MPA on 1 and 4 functions respectively.

When problem's dimension increases to 100D, DCDE shows a similar performance to the results of 50D problems. More specifically, DCDE still stands first since it obtains the smallest fitness value for 13 functions, which is far more than other algorithms. JADE ranks the second place by getting 7 optimal results which is only about half of DCDE. In terms of the Wilcoxon rank-sum test, DCDE performs significantly better than all compared DE variants on most test functions. The significant difference is much more obvious when comparing with non-DE algorithms, i.e., DCDE variant is significantly better than EO, MPA and SGLSCA on 26, 24 and 27 functions respectively.

To draw a comprehensive comparison based on statistical analysis, multiple-problem analysis by the Wilcoxon signed-rank test is conducted at the 0.05 significance level using the KEEL software (Alcalá-Fdez et al., 2009). For the sake of clarity, the experimental results for different dimensions are presented in Table 4, where R^+ and R^- represents the sum of ranks that DCDE is significantly better and worse than the compared algorithm respectively, and the significance marker "YES" means that there is 95% probability to ensure that there is significant difference between

Table 1. Comparison results between proposed DCDE variant and other algorithms on 30D test functions.

Func.	Metric	JADE		SinDE		AGDE		EFADE		GODE		EO		MPA		SGLSCA		DCDE
f_1	Mean	1.22E-14	+	2.74E+02	+	2.84E-16	−	4.77E-14	+	3.13E-15	−	4.39E+03	+	1.18E+07	+	3.13E+03	+	**1.13E-22**
	Std.	4.98E-15		8.33E+02		2.01E-15		2.37E-13		5.95E-15		5.26E+03		1.61E+07		3.83E+03		5.93E-22
f_3	Mean	2.06E+04	=	7.46E+02	+	**3.75E-14**	−	2.50E-13	+	1.14E+01	+	3.08E+01	+	2.75E+02	+	4.74E+03	+	9.14E-07
	Std.	2.08E+04		4.29E+02		2.72E-14		3.39E-13		1.20E+01		5.74E+01		4.50E+02		2.38E+03		5.66E-06
f_4	Mean	3.50E+01	−	9.18E+01	+	**3.14E+01**	−	3.55E+01	−	5.95E+01	−	4.42E+01	−	8.55E+01	+	1.03E+02	+	7.18E+01
	Std.	3.00E+01		1.48E+01		3.09E+01		2.93E+01		2.06E+00		3.28E+01		2.80E+01		2.43E+01		1.07E+01
f_5	Mean	2.79E+01	+	4.28E+01	+	7.34E+01	+	4.21E+01	+	1.12E+02	+	7.53E+01	+	7.10E+01	+	9.35E+01	+	**2.23E+01**
	Std.	3.22E+00		7.93E+00		8.04E+00		8.22E+00		6.23E+01		2.07E+01		1.55E+01		2.60E+01		8.10E+00
f_6	Mean	1.57E-13	+	**0.00E+00**	−	1.14E-13	+	1.23E-13	+	6.07E-09	+	2.74E-01	+	3.78E+00	+	1.23E+01	+	1.42E-07
	Std.	6.03E-14		0.00E+00		7.65E-29		3.12E-14		1.04E-08		3.45E-01		1.52E+00		5.43E+00		3.81E-07
f_7	Mean	5.44E+01	+	7.98E+01	+	1.15E+02	+	7.34E+01	+	2.05E+02	+	1.17E+02	+	1.30E+02	+	1.33E+02	+	**5.24E+01**
	Std.	3.83E+00		8.09E+00		9.08E+00		1.01E+01		1.11E+01		2.75E+01		1.99E+01		2.75E+01		6.25E+00
f_8	Mean	2.55E+01	+	3.46E+01	+	7.99E+01	+	5.67E+01	+	3.79E+01	+	7.22E+01	+	7.69E+01	+	9.69E+01	+	**2.21E+01**
	Std.	3.89E+00		8.90E+00		7.32E+00		1.06E+01		3.69E+01		1.95E+01		1.54E+01		2.72E+01		7.66E+00
f_9	Mean	9.09E-03	−	1.12E-28	+	0.00E+00	−	9.09E-03	=	**0.00E+00**	−	1.05E+02	+	2.42E+02	+	3.78E+02	+	7.20E-29
	Std.	6.43E-02		1.32E-28		0.00E+00		6.43E-02		0.00E+00		2.36E+02		1.08E+02		2.99E+02		3.37E-30
f_{10}	Mean	**1.85E+03**	=	2.64E+03	+	3.84E+03	+	3.14E+03	+	3.28E+03	+	3.65E+03	+	2.76E+03	+	3.50E+03	+	1.96E+03
	Std.	2.18E+02		4.63E+02		2.73E+02		8.34E+02		1.47E+03		7.31E+02		3.62E+02		6.38E+02		5.06E+02
f_{11}	Mean	2.37E+01	+	5.40E+01	+	1.76E+01	+	**1.56E+01**	+	4.15E+01	+	7.00E+01	+	6.53E+01	+	1.16E+02	+	1.89E+01
	Std.	2.12E+01		3.35E+01		1.83E+01		9.76E+00		3.19E+01		4.56E+01		2.81E+01		3.71E+01		2.43E+01
f_{12}	Mean	**1.25E+03**	−	6.84E+04	+	1.01E+04	+	7.53E+03	=	8.24E+03	=	6.38E+04	+	4.35E+05	+	5.25E+06	+	5.75E+03
	Std.	3.51E+02		9.84E+04		7.69E+03		6.93E+03		7.03E+03		4.62E+04		6.23E+05		3.23E+06		3.10E+03
f_{13}	Mean	3.65E+01	+	6.08E+03	+	5.43E+01	=	4.71E+02	+	8.04E+01	+	2.13E+04	+	3.33E+02	+	9.66E+04	+	**2.36E+01**
	Std.	1.45E+01		5.80E+03		1.56E+02		1.44E+03		6.21E+00		1.77E+04		1.60E+02		4.52E+04		1.04E+01
f_{14}	Mean	7.73E+03	+	3.36E+01	=	2.57E+01	=	**1.19E+01**	−	5.14E+01	+	7.44E+03	+	3.86E+01	+	8.70E+03	+	2.89E+01
	Std.	8.61E+03		2.58E+01		1.10E+01		5.25E+00		1.61E+01		5.55E+03		1.03E+01		1.06E+04		3.52E+00
f_{15}	Mean	1.44E+03	+	1.50E+02	+	1.15E+01	−	**8.85E+00**	−	3.09E+01	+	4.58E+03	+	7.92E+01	+	1.31E+04	+	1.20E+01
	Std.	3.25E+03		7.69E+02		1.48E+01		4.05E+00		1.14E+01		4.65E+03		3.06E+01		8.04E+03		4.08E+00
f_{16}	Mean	4.51E+02	+	2.99E+02	+	3.83E+02	+	4.38E+02	+	3.45E+02	+	6.95E+02	+	4.12E+02	+	9.38E+02	+	**1.61E+02**
	Std.	1.28E+02		1.98E+02		1.63E+02		1.87E+02		2.26E+02		2.81E+02		1.62E+02		2.79E+02		1.51E+02
f_{17}	Mean	8.10E+01	+	**2.27E+01**	−	8.35E+01	+	5.33E+01	=	8.42E+01	+	2.42E+02	+	7.76E+01	+	3.08E+02	+	3.61E+01
	Std.	3.51E+01		1.13E+01		2.74E+01		5.17E+01		4.60E+01		1.13E+02		4.93E+01		1.63E+02		8.87E+00
f_{18}	Mean	2.72E+04	=	4.08E+04	+	5.61E+02	+	1.45E+03	+	**3.57E+01**	−	1.47E+05	+	5.61E+01	−	1.31E+05	+	9.92E+01
	Std.	7.54E+04		2.47E+04		1.18E+03		1.90E+03		4.29E+00		1.32E+05		2.09E+01		7.33E+04		7.50E+01
f_{19}	Mean	1.49E+03	+	4.88E+02	=	1.19E+01	+	**6.35E+00**	−	1.34E+01	+	5.45E+03	+	1.83E+01	+	1.91E+04	+	1.03E+01
	Std.	4.56E+03		2.10E+03		3.29E+00		2.34E+00		4.92E+00		7.57E+03		4.67E+00		2.83E+04		2.59E+00
f_{20}	Mean	9.71E+01	+	**5.14E+01**	−	1.02E+02	+	1.11E+02	=	8.36E+01	−	2.85E+02	+	1.09E+02	+	2.93E+02	+	9.00E+01
	Std.	5.43E+01		5.87E+01		6.01E+01		7.18E+01		1.06E+02		1.41E+02		5.79E+01		7.99E+01		5.99E+01
f_{21}	Mean	2.29E+02	+	2.35E+02	+	2.78E+02	+	2.46E+02	+	3.63E+02	+	2.65E+02	+	2.42E+02	+	2.79E+02	+	**2.21E+02**
	Std.	5.79E+00		7.96E+00		7.88E+00		1.02E+01		1.21E+01		2.24E+01		4.86E+01		2.34E+01		6.01E+00
f_{22}	Mean	**1.00E+02**	=	**1.00E+02**	=	**1.00E+02**	=	**1.00E+02**	=	**1.00E+02**	=	2.24E+03	+	1.17E+02	+	**1.00E+02**	+	**1.00E+02**
	Std.	0.00E+00		0.00E+00		0.00E+00		0.00E+00		0.00E+00		1.95E+03		3.10E+00		1.05E+00		0.00E+00
f_{23}	Mean	3.74E+02	+	3.90E+02	+	4.20E+02	+	3.94E+02	+	5.15E+02	+	4.22E+02	+	3.97E+02	+	4.41E+02	+	**3.66E+02**
	Std.	5.87E+00		7.51E+00		7.73E+00		1.15E+01		2.11E+01		1.93E+01		1.34E+01		2.20E+01		8.89E+00
f_{24}	Mean	**4.40E+02**	=	4.46E+02	+	4.94E+02	+	4.75E+02	+	5.88E+02	+	4.85E+02	+	4.72E+02	+	4.96E+02	+	4.41E+02
	Std.	5.10E+00		9.02E+00		1.70E+01		1.29E+01		9.32E+00		1.96E+01		1.41E+01		2.47E+01		8.38E+00
f_{25}	Mean	**3.87E+02**	+	**3.87E+02**	=	**3.87E+02**	−	**3.87E+02**	+	**3.87E+02**	−	3.90E+02	=	3.91E+02	=	4.08E+02	+	**3.87E+02**
	Std.	2.09E-01		2.01E-01		8.02E-02		5.75E-02		2.32E-02		1.44E+01		1.35E+01		1.77E+01		9.54E-02
f_{26}	Mean	1.18E+03	+	1.13E+03	+	1.57E+03	+	1.25E+03	+	2.21E+03	+	1.56E+03	+	**4.14E+02**	−	1.82E+03	+	1.09E+03
	Std.	7.13E+01		8.43E+01		1.18E+02		4.36E+02		7.31E+02		6.01E+02		3.38E+02		6.56E+02		9.10E+01
f_{27}	Mean	5.02E+02	−	4.95E+02	−	5.02E+02	=	5.00E+02	−	**4.88E+02**	−	5.23E+02	+	5.11E+02	+	5.38E+02	+	5.04E+02
	Std.	8.32E+00		7.78E+00		1.02E+01		7.92E+00		1.02E+01		1.60E+01		1.06E+01		2.23E+01		4.85E+00
f_{28}	Mean	3.33E+02	=	3.37E+02	=	3.70E+02	+	**3.11E+02**	−	3.29E+02	=	3.31E+02	=	3.99E+02	+	4.40E+02	+	3.35E+02
	Std.	5.47E+01		5.05E+01		5.36E+01		3.20E+01		4.91E+01		5.49E+01		4.26E+01		2.26E+01		5.09E+01
f_{29}	Mean	4.81E+02	+	4.85E+02	=	5.48E+02	+	**4.52E+02**	−	5.38E+02	+	7.45E+02	+	5.24E+02	+	9.16E+02	+	4.58E+02
	Std.	2.47E+01		5.37E+01		1.96E+01		5.50E+01		1.09E+02		1.63E+02		6.72E+01		2.11E+02		1.38E+01
f_{30}	Mean	2.12E+03	−	4.33E+03	+	2.78E+03	−	2.48E+03	−	**2.00E+03**	−	8.11E+03	+	3.26E+03	+	2.68E+05	+	2.96E+03
	Std.	1.64E+02		1.37E+03		1.30E+03		7.23E+02		6.04E+01		6.15E+03		7.47E+02		1.68E+05		5.60E+02
+/=/−		18/6/5		19/6/4		18/4/7		15/5/9		18/3/8		26/2/1		26/1/2		29/0/0		- - -

Table 2. Comparison results between proposed DCDE variant and other algorithms on 50D test functions.

Func.	Metric	JADE		SinDE		AGDE		EFADE		GODE		EO		MPA		SGLSCA		DCDE
f_1	Mean	**5.46E-14**	+	3.07E+03	+	5.42E-06	+	1.66E+00	+	1.28E+02	+	3.75E+03	+	4.78E+07	+	3.08E+03	+	1.37E-12
	Std.	4.30E-14		3.08E+03		2.07E-05		5.56E+00		5.29E+02		3.90E+03		8.89E+07		3.85E+03		5.06E-12
f_3	Mean	1.52E+04	−	2.97E+04	+	**2.02E-07**	−	8.55E-05	−	7.08E+04	+	3.84E+03	+	4.93E+03	+	1.90E+04	+	1.68E+01
	Std.	3.31E+04		6.43E+03		4.67E-07		1.72E-04		1.26E+04		3.15E+03		2.44E+03		4.50E+03		2.15E+01
f_4	Mean	6.21E+01	−	6.87E+01	=	5.05E+01	−	**3.79E+01**	−	8.52E+01	=	7.72E+01	=	1.48E+02	+	1.88E+02	+	7.34E+01
	Std.	5.11E+01		5.70E+01		4.43E+01		2.53E+01		5.30E+01		5.08E+01		5.62E+01		5.11E+01		5.19E+01
f_5	Mean	5.40E+01	+	6.74E+01	+	2.20E+02	+	9.46E+01	+	1.36E+02	+	1.85E+02	+	1.67E+02	+	2.35E+02	+	**3.57E+01**
	Std.	8.11E+00		1.58E+01		1.21E+01		1.74E+01		1.22E+02		3.20E+01		3.88E+01		6.86E+01		8.34E+00
f_6	Mean	1.27E-13	−	**0.00E+00**	−	1.68E-13	−	1.84E-13	−	1.76E-07	−	3.39E+00	+	1.17E+01	+	2.10E+01	+	1.80E-06
	Std.	3.73E-14		0.00E+00		5.74E-14		5.57E-14		7.48E-07		3.29E+00		1.79E+00		8.72E+00		2.50E-06
f_7	Mean	1.04E+02	+	1.30E+02	+	2.88E+02	+	1.94E+02	+	3.98E+02	+	2.65E+02	+	2.82E+02	+	2.61E+02	+	**8.37E+01**
	Std.	7.19E+00		1.60E+01		1.52E+01		8.02E+01		1.37E+01		7.10E+01		3.62E+01		4.05E+01		8.61E+00
f_8	Mean	5.48E+01	+	6.78E+01	+	2.15E+02	+	8.70E+01	+	1.72E+02	+	1.85E+02	+	1.62E+02	+	2.43E+02	+	**4.03E+01**
	Std.	7.86E+00		1.18E+01		1.13E+01		1.70E+01		1.24E+02		3.67E+01		3.44E+01		5.76E+01		1.21E+01
f_9	Mean	1.02E+00	+	**5.54E-28**	+	1.02E-13	+	8.17E-02	=	1.27E-02	−	1.80E+03	+	1.57E+03	+	2.74E+03	+	8.95E-03
	Std.	9.97E-01		2.92E-28		3.45E-14		4.28E-01		6.62E-02		1.73E+03		5.40E+02		1.24E+03		2.71E-02
f_{10}	Mean	**3.71E+03**	−	5.13E+03	+	8.52E+03	+	7.32E+03	+	6.79E+03	+	6.29E+03	+	6.13E+03	+	5.66E+03	+	4.15E+03
	Std.	3.36E+02		7.99E+02		3.36E+02		1.99E+03		2.94E+03		1.00E+03		6.86E+02		1.01E+03		7.35E+02
f_{11}	Mean	1.31E+02	+	3.88E+01	+	5.22E+01	+	4.44E+01	+	9.01E+01	+	1.54E+02	+	1.80E+02	+	2.65E+02	+	**3.62E+01**
	Std.	3.48E+01		5.09E+00		1.37E+01		7.52E+00		4.84E+01		5.51E+01		3.97E+01		6.64E+01		4.48E+00
f_{12}	Mean	**5.32E+03**	−	8.98E+05	+	7.03E+04	+	4.76E+04	+	6.00E+04	+	7.68E+05	+	6.99E+06	+	2.09E+07	+	2.20E+04
	Std.	3.61E+03		4.86E+05		3.67E+04		2.78E+04		3.61E+04		6.13E+05		5.01E+06		1.28E+07		1.30E+04
f_{13}	Mean	**2.85E+02**	−	1.51E+03	=	2.68E+03	=	2.22E+03	=	3.24E+02	−	1.07E+04	+	1.17E+03	=	9.52E+04	+	1.39E+03
	Std.	1.95E+02		1.85E+03		4.38E+03		3.08E+03		2.35E+02		9.38E+03		3.16E+02		6.66E+04		1.20E+03
f_{14}	Mean	1.91E+04	+	5.67E+03	+	1.95E+02	=	**4.43E+01**	−	1.25E+02	+	5.00E+04	+	7.33E+01	=	1.20E+05	+	7.75E+01
	Std.	5.71E+04		6.19E+03		3.98E+02		1.46E+01		8.44E+00		3.73E+04		1.46E+01		1.19E+05		1.58E+01
f_{15}	Mean	4.35E+02	=	9.41E+02	=	4.68E+02	−	**9.62E+01**	−	1.06E+02	−	1.17E+04	+	3.02E+02	−	3.76E+04	+	7.71E+02
	Std.	7.55E+02		1.07E+03		1.47E+03		1.03E+02		9.34E+00		6.95E+03		6.97E+01		2.40E+04		7.94E+02
f_{16}	Mean	8.25E+02	+	7.54E+02	+	9.94E+02	+	9.06E+02	+	1.06E+03	+	1.43E+03	+	8.24E+02	+	1.57E+03	+	**5.96E+02**
	Std.	1.62E+02		2.21E+02		1.97E+02		2.28E+02		4.02E+02		4.49E+02		1.60E+02		3.89E+02		2.30E+02
f_{17}	Mean	5.87E+02	+	3.59E+02	=	7.00E+02	+	5.56E+02	+	7.41E+02	+	1.14E+03	+	6.48E+02	+	1.16E+03	+	**3.43E+02**
	Std.	1.39E+02		1.68E+02		1.27E+02		1.81E+02		2.76E+02		3.13E+02		1.39E+02		3.08E+02		1.64E+02
f_{18}	Mean	7.80E+04	−	2.34E+05	+	2.87E+03	−	1.86E+03	−	5.46E+02	−	2.76E+05	+	**1.33E+02**	−	8.02E+05	+	4.42E+03
	Std.	3.03E+05		1.97E+05		2.40E+03		2.14E+03		4.33E+02		2.51E+05		3.37E+01		4.39E+05		3.94E+03
f_{19}	Mean	7.05E+02	+	6.77E+03	+	6.49E+01	−	**2.02E+01**	−	5.15E+01	+	2.07E+04	+	5.52E+01	+	3.34E+04	+	1.85E+03
	Std.	3.37E+03		4.84E+03		2.51E+02		1.41E+01		1.33E+01		1.12E+04		1.28E+01		1.43E+04		4.21E+03
f_{20}	Mean	4.51E+02	+	2.26E+02	=	5.20E+02	+	4.20E+02	+	5.23E+02	+	9.72E+02	+	3.91E+02	+	7.85E+02	+	**2.01E+02**
	Std.	1.15E+02		1.41E+02		1.38E+02		1.67E+02		2.54E+02		3.45E+02		1.61E+02		2.55E+02		1.47E+02
f_{21}	Mean	2.53E+02	+	2.65E+02	+	4.27E+02	+	2.92E+02	+	4.66E+02	+	3.55E+02	+	3.09E+02	+	3.60E+02	+	**2.37E+02**
	Std.	9.43E+00		1.23E+01		1.45E+01		1.83E+01		1.21E+02		3.84E+01		1.82E+01		3.55E+01		8.33E+00
f_{22}	Mean	3.61E+03	=	4.56E+03	+	7.93E+03	+	5.81E+03	+	6.51E+03	+	7.06E+03	+	**1.26E+03**	−	4.11E+03	+	3.29E+03
	Std.	1.67E+03		1.81E+03		3.04E+03		3.88E+03		3.60E+03		9.69E+02		2.58E+03		3.24E+03		2.00E+03
f_{23}	Mean	4.79E+02	+	4.81E+02	+	6.47E+02	+	5.17E+02	+	7.56E+02	+	5.76E+02	+	5.31E+02	+	6.39E+02	+	**4.59E+02**
	Std.	1.10E+01		1.59E+01		1.49E+01		2.03E+01		1.54E+01		3.81E+01		1.70E+01		5.24E+01		8.62E+00
f_{24}	Mean	5.42E+02	+	5.50E+02	+	6.85E+02	+	5.78E+02	+	8.39E+02	+	6.43E+02	+	6.01E+02	+	6.82E+02	+	**5.29E+02**
	Std.	9.44E+00		1.27E+01		5.34E+01		1.88E+01		1.42E+01		4.41E+01		1.80E+01		4.07E+01		1.14E+01
f_{25}	Mean	5.33E+02	+	4.89E+02	−	5.18E+02	+	4.93E+02	=	5.08E+02	+	5.63E+02	+	5.44E+02	+	5.75E+02	+	**4.80E+02**
	Std.	3.36E+01		2.50E+01		3.80E+01		2.49E+01		3.57E+01		3.94E+01		4.24E+01		2.77E+01		2.98E-02
f_{26}	Mean	1.63E+03	+	1.60E+03	+	3.02E+03	+	2.08E+03	+	4.02E+03	+	2.92E+03	+	**8.63E+02**	−	1.68E+03	−	1.38E+03
	Std.	1.14E+02		1.58E+02		1.46E+02		2.05E+02		6.47E+02		1.11E+03		8.24E+02		2.11E+03		1.38E+02
f_{27}	Mean	5.51E+02	+	**5.12E+02**	−	5.13E+02	−	5.30E+02	+	5.13E+02	−	6.87E+02	+	6.35E+02	+	8.46E+02	+	5.22E+02
	Std.	2.44E+01		6.82E+00		1.54E+01		1.19E+01		1.52E+01		7.50E+01		6.16E+01		1.11E+02		7.94E+00
f_{28}	Mean	4.91E+02	+	4.60E+02	=	4.78E+02	+	4.59E+02	=	4.66E+02	+	5.00E+02	+	5.16E+02	+	5.29E+02	+	**4.59E+02**
	Std.	2.38E+01		6.91E+00		2.31E+01		1.15E-13		1.68E+01		2.32E+01		3.68E+01		3.65E+01		1.15E-13
f_{29}	Mean	4.78E+02	+	**3.60E+02**	−	6.06E+02	+	5.23E+02	+	4.72E+02	+	1.13E+03	+	7.41E+02	+	1.52E+03	+	4.02E+02
	Std.	7.90E+01		4.39E+01		8.78E+01		1.22E+02		1.14E+02		4.35E+02		1.73E+02		2.95E+02		3.79E+01
f_{30}	Mean	6.67E+05	+	6.94E+05	+	6.16E+05	=	6.13E+05	+	**5.89E+05**	−	1.07E+06	+	6.83E+05	+	6.89E+06	+	6.07E+05
	Std.	6.90E+04		6.54E+04		3.74E+04		3.94E+04		1.81E+04		2.22E+05		8.28E+04		2.89E+06		3.63E+04
+/=/−		20/2/7		19/6/4		19/3/7		18/4/7		21/1/7		28/1/0		23/2/4		28/0/1		- - -

Table 3. Comparison results between proposed DCDE variant and other algorithms on $100D$ test functions.

Func.	Metric	JADE		SinDE		AGDE		EFADE		GODE		EO		MPA		SGLSCA		DCDE
f_1	Mean	**5.17E-14**	–	4.52E+03	+	4.86E+02	+	1.91E+01	+	4.87E+03	+	8.29E+03	+	1.00E+09	+	6.20E+03	+	1.52E-04
	Std.	1.18E-14		4.11E+03		8.41E+02		3.48E+01		5.14E+03		1.16E+04		4.65E+08		7.45E+03		4.98E-04
f_3	Mean	1.85E+05	=	2.75E+05	+	**2.45E+02**	–	4.60E+03	–	3.41E+05	+	6.72E+04	–	4.57E+04	–	1.17E+05	=	1.17E+05
	Std.	2.04E+05		2.37E+04		2.53E+02		1.87E+03		3.87E+04		1.47E+04		9.75E+03		1.48E+04		1.69E+04
f_4	Mean	**1.23E+02**	–	2.18E+02	+	1.84E+02	–	2.01E+02	–	2.15E+02	=	2.07E+02	=	4.56E+02	+	2.91E+02	+	2.15E+02
	Std.	6.75E+01		1.23E+01		3.45E+01		7.50E+00		2.29E+01		4.72E+01		8.34E+01		4.79E+01		1.34E+01
f_5	Mean	2.60E+02	+	1.21E+02	+	7.26E+02	+	1.65E+02	+	5.72E+02	+	5.30E+02	+	5.30E+02	+	6.49E+02	+	**6.37E+01**
	Std.	1.65E+01		1.93E+01		2.43E+01		7.89E+01		3.21E+02		7.07E+01		3.81E+01		1.33E+02		1.23E+01
f_6	Mean	1.19E-05	–	2.46E-08	–	**2.84E-13**	–	2.10E-08	–	2.15E-03	+	2.22E+01	+	2.65E+01	+	3.70E+01	+	3.31E-05
	Std.	7.48E-05		4.70E-08		5.74E-14		9.72E-09		4.20E-03		7.63E+00		2.43E+00		9.79E+00		2.11E-05
f_7	Mean	3.74E+02	+	2.40E+02	+	8.48E+02	+	5.83E+02	+	9.36E+02	+	1.04E+03	+	1.09E+03	+	7.81E+02	+	**1.67E+02**
	Std.	1.97E+01		2.34E+01		2.40E+01		2.48E+02		2.18E+01		2.27E+02		8.51E+01		8.36E+01		1.38E+01
f_8	Mean	2.63E+02	+	1.28E+02	+	7.18E+02	+	1.98E+02	+	7.77E+02	+	5.49E+02	+	5.25E+02	+	5.42E+02	+	**6.91E+01**
	Std.	2.10E+01		2.52E+01		2.79E+01		1.42E+02		1.59E+02		7.15E+01		3.18E+01		1.39E+02		1.26E+01
f_9	Mean	2.19E+00	+	**3.58E-03**	=	1.01E+00	+	4.38E+00	+	9.79E+00	+	1.52E+04	+	1.16E+04	+	1.61E+04	+	1.03E-01
	Std.	1.57E+00		1.77E-02		1.21E+00		4.63E+00		1.17E+01		3.26E+03		3.44E+03		3.60E+03		1.65E-01
f_{10}	Mean	1.70E+04	+	1.54E+04	+	2.40E+04	+	2.46E+04	+	1.17E+04	–	1.47E+04	+	1.75E+04	+	1.32E+04	+	**1.15E+04**
	Std.	4.66E+02		2.59E+03		5.81E+02		5.65E+02		5.63E+03		1.50E+03		1.03E+03		1.55E+03		2.46E+03
f_{11}	Mean	1.62E+04	+	3.71E+02	–	1.52E+02	–	**1.29E+02**	–	6.25E+02	–	7.70E+02	=	1.41E+03	+	1.50E+03	+	7.00E+02
	Std.	1.03E+04		9.18E+01		5.68E+01		5.25E+01		7.21E+01		2.43E+02		1.74E+02		2.15E+02		1.11E+02
f_{12}	Mean	**2.01E+04**	–	1.71E+06	+	2.63E+05	=	2.29E+05	–	2.58E+05	=	1.27E+06	+	5.41E+07	+	1.10E+08	+	3.16E+05
	Std.	6.76E+03		7.40E+05		1.07E+05		7.45E+04		1.04E+05		5.18E+05		2.30E+07		4.69E+07		1.54E+05
f_{13}	Mean	2.05E+03	=	2.22E+03	=	4.08E+03	+	2.17E+03	+	3.08E+03	=	1.29E+04	+	8.32E+03	+	6.01E+04	+	**1.63E+03**
	Std.	1.63E+03		1.52E+03		4.62E+03		1.34E+03		3.66E+03		7.61E+03		1.83E+03		2.62E+04		1.11E+03
f_{14}	Mean	4.40E+02	–	3.76E+05	+	8.98E+03	–	6.08E+03	–	6.90E+03	–	2.26E+05	+	**2.58E+02**	–	3.52E+05	+	1.56E+04
	Std.	8.09E+01		1.92E+05		9.76E+03		6.42E+03		6.61E+03		9.27E+04		3.40E+01		1.12E+05		8.23E+03
f_{15}	Mean	**2.95E+02**	=	5.13E+02	=	2.39E+03	+	7.96E+02	=	3.05E+03	+	4.32E+03	+	8.72E+02	+	4.49E+04	+	4.27E+02
	Std.	7.65E+01		4.99E+02		2.76E+03		9.26E+02		5.29E+03		3.35E+03		1.62E+02		2.02E+04		3.31E+02
f_{16}	Mean	3.59E+03	+	2.51E+03	+	3.52E+03	+	3.74E+03	+	6.51E+03	+	3.76E+03	+	3.61E+03	+	3.77E+03	+	**2.04E+03**
	Std.	3.12E+02		5.52E+02		2.78E+02		9.60E+02		1.79E+03		9.06E+02		4.29E+02		7.24E+02		7.17E+02
f_{17}	Mean	2.73E+03	+	1.44E+03	=	2.43E+03	+	2.40E+03	+	2.59E+03	+	3.42E+03	+	2.37E+03	+	2.82E+03	+	**1.30E+03**
	Std.	2.41E+02		3.16E+02		2.66E+02		5.81E+02		5.34E+02		7.00E+02		2.68E+02		5.54E+02		4.44E+02
f_{18}	Mean	**2.97E+02**	–	1.03E+06	+	2.23E+04	–	4.19E+04	–	1.17E+05	=	5.60E+05	+	7.63E+02	–	6.24E+05	+	1.08E+05
	Std.	6.63E+01		4.76E+05		1.20E+04		2.49E+04		5.65E+04		1.85E+05		2.14E+02		2.19E+05		4.19E+04
f_{19}	Mean	**2.15E+02**	–	7.82E+02	=	2.64E+03	+	8.38E+02	=	6.26E+03	+	3.51E+03	+	2.62E+02	–	4.29E+05	+	6.76E+02
	Std.	4.33E+01		9.18E+02		3.71E+03		1.01E+03		7.43E+03		4.44E+03		5.42E+01		3.73E+05		5.64E+02
f_{20}	Mean	2.79E+03	+	1.58E+03	=	2.24E+03	+	2.42E+03	+	1.95E+03	+	3.03E+03	+	2.32E+03	+	2.48E+03	+	**1.52E+03**
	Std.	2.26E+02		2.89E+02		2.42E+02		3.75E+02		4.31E+02		6.20E+02		2.43E+02		4.48E+02		3.03E+02
f_{21}	Mean	4.81E+02	+	3.60E+02	+	9.53E+02	+	3.98E+02	+	1.03E+03	+	6.62E+02	+	6.98E+02	+	6.84E+02	+	**2.96E+02**
	Std.	1.80E+01		2.16E+01		2.46E+01		6.65E+01		2.16E+01		8.85E+01		3.37E+01		6.48E+01		1.94E+01
f_{22}	Mean	1.82E+04	+	1.62E+04	+	2.51E+04	+	2.57E+04	+	1.76E+04	=	1.59E+04	+	1.05E+04	=	**7.31E+03**	=	1.22E+04
	Std.	4.39E+02		2.04E+03		5.19E+02		6.86E+02		9.24E+03		1.32E+03		1.02E+04		8.25E+03		1.41E+03
f_{23}	Mean	7.44E+02	+	6.48E+02	+	1.04E+03	+	9.29E+02	+	6.13E+02	+	9.49E+02	+	1.00E+03	+	1.10E+03	+	**6.03E+02**
	Std.	1.06E+01		2.24E+01		1.96E+01		1.88E+02		2.22E+01		7.85E+01		2.61E+01		1.00E+02		1.61E+01
f_{24}	Mean	1.10E+03	+	9.75E+02	+	1.54E+03	+	1.04E+03	+	1.29E+03	+	1.32E+03	+	1.42E+03	+	1.45E+03	+	**9.23E+02**
	Std.	1.97E+01		2.21E+01		2.43E+01		2.99E+01		3.31E+02		8.84E+01		8.92E+01		1.10E+02		1.16E+01
f_{25}	Mean	7.44E+02	=	**7.08E+02**	–	7.33E+02	=	7.40E+02	=	7.51E+02	=	7.71E+02	+	1.00E+03	+	8.63E+02	+	7.30E+02
	Std.	3.84E+01		4.80E+01		5.29E+01		3.48E+01		5.89E+01		5.40E+01		7.58E+01		5.89E+01		4.47E+01
f_{26}	Mean	5.10E+03	+	4.00E+03	+	9.69E+03	+	4.47E+03	+	5.96E+03	+	8.89E+03	+	8.45E+03	+	8.88E+03	+	**3.39E+03**
	Std.	1.69E+02		2.49E+02		2.93E+02		3.00E+02		3.08E+03		1.94E+03		8.93E+02		2.47E+03		1.94E+02
f_{27}	Mean	6.42E+02	=	6.36E+02	=	6.27E+02	–	6.47E+02	+	**6.03E+02**	–	8.10E+02	+	7.81E+02	+	9.81E+02	+	6.39E+02
	Std.	1.94E+01		1.38E+01		2.23E+01		1.41E+01		2.26E+01		6.43E+01		5.10E+01		1.10E+02		1.27E+01
f_{28}	Mean	5.46E+02	=	**5.33E+02**	=	5.57E+02	+	5.51E+02	+	5.57E+02	+	5.62E+02	+	7.61E+02	+	6.28E+02	+	5.37E+02
	Std.	3.90E+01		1.88E+01		2.84E+01		1.49E+01		2.97E+01		3.08E+01		4.69E+01		4.30E+01		1.81E+01
f_{29}	Mean	2.55E+03	+	1.61E+03	=	2.77E+03	+	2.91E+03	+	1.89E+03	+	3.45E+03	+	3.76E+03	+	4.48E+03	+	**1.48E+03**
	Std.	2.05E+02		3.34E+02		2.33E+02		7.04E+02		9.00E+02		6.17E+02		2.97E+02		4.93E+02		2.61E+02
f_{30}	Mean	**3.31E+03**	–	4.02E+03	+	5.12E+03	+	4.14E+03	+	3.58E+03	–	3.78E+04	+	1.20E+05	+	6.46E+06	+	3.51E+03
	Std.	1.08E+03		8.34E+02		2.66E+03		1.27E+03		1.52E+03		7.09E+04		8.72E+04		2.60E+06		6.57E+02
+/=/–		15/6/8		17/9/3		20/2/7		19/3/7		18/6/5		26/2/1		24/1/4		27/2/0		- - -

Table 4. Results of the multiple-problem Wilcoxon signed-rank test between DCDE and other algorithms at the 0.05 significance level for 30*D*, 50*D* and 100*D* functions.

Dimension	DCDE *VS.*	R^+	R^-	Asymptotic p-value	$\alpha = 0.05$
30*D*	JADE	319.50	115.50	0.0247	YES
	SinDE	393.50	41.50	0.0001	YES
	AGDE	344.50	90.50	0.0058	YES
	EFADE	302.50	132.50	0.0645	NO
	GODE	344.50	90.50	0.0058	YES
	EO	427.00	8.00	0.0000	YES
	MPA	396.00	39.00	0.0001	YES
	SGLSCA	406.00	0.00	0.0000	YES
50*D*	JADE	312.00	123.00	0.0394	YES
	SinDE	405.00	30.00	0.0000	YES
	AGDE	347.00	88.00	0.0049	YES
	EFADE	313.00	93.00	0.0119	YES
	GODE	314.00	121.0	0.0355	YES
	EO	435.00	0.00	0.0000	YES
	MPA	309.00	126.00	0.0443	YES
	SGLSCA	435.00	0.00	0.0000	YES
100*D*	JADE	304.00	131.00	0.0599	NO
	SinDE	399.50	35 .50	0.0001	YES
	AGDE	307.00	128.00	0.0516	NO
	EFADE	303.50	131.50	0.0607	NO
	GODE	340.00	66.00	0.0017	YES
	EO	408.00	27.00	0.0000	YES
	MPA	334.00	101.00	0.0114	YES
	SGLSCA	388.00	18.00	0.0000	YES

DCDE and other algorithms at the significance level of 0.05 and otherwise for "NO". From the results in Table 4, we can see that DCDE can provide higher R^+ than R^- compared with all the other algorithms when solving 30*D*, 50*D* and 100*D* functions, which indicates that DCDE performs significantly better than other algorithms. Moreover, the p values in different dimensions are less than the significance level 0.05 in most cases, which implies that there is significant difference between DCDE and other algorithms for all the three kinds of dimensions. Besides, the average ranking of DCDE and other compared algorithms by Friedman test at the 0.05 significance level for 30*D*, 50*D* and 100*D* functions is listed in Table 5, which is also obtained by the software tool KEEL (Alcalá-Fdez et al., 2009), and the first ranking values are shown in boldface. The p value computed by Friedman test for different dimensional functions are all approximately equal to zero, which indicates that there is significant difference in terms of the performances of the test algorithms. From the ranking results, we can conclude that DCDE ranks the first position for all the three kinds of dimensional functions. The ranking on 50D and 100D is lower than that on 30D, which proves that the performance of DCDE becomes better with the increase of problem's dimension.

Table 5. Results of the multi-problem Friedman test between DCDE and other algorithms at the 0.05 significance level for 30D, 50D and 100D functions.

Dimension	JADE	SinDE	AGDE	EFADE	GODE	EO	MPA	SGLSCA	DCDE	p-value
30D	3.7931	4.3276	4.5345	3.4483	4.8621	7.0345	5.8276	8.4138	**2.7586**	0.0000
50D	3.7586	3.8621	5.4655	3.6552	5.3966	7.2069	5.2069	8.0000	**2.4483**	0.0000
100D	3.8276	3.5862	5.1897	4.1379	5.4483	6.7069	6.1207	7.5000	**2.4828**	0.0000

Figs. 2-4 respectively depicts the convergence progress of DCDE and corresponding compared DE variants and non-DE algorithms on several typical 30D, 50D and 100D functions which are all selected from IEEE CEC 2017 benchmark test suite with different characteristics. As we can see in these graphics, the performance of DCDE is not remarkable in the early stage of evolution but it has the best performance among all these competitors in terms of the convergence speed and the final accuracy during the latter evolutionary process for most cases. The reason for this situation is that the design philosophy of both mutation operator and core parameters in DCDE is to focus on global exploration ability in the early stages and on local exploitation ability in the later stage during the searching process.

Through the comparison results of the function error value, non-parametric statistical analysis and the convergence figures, we can draw a conclusion that the proposed DCDE surpasses all the eight comparison algorithms and keeps the leading position on solving optimization problems in IEEE CEC 2017, and as the dimensionality of the problem increases, the advantage of DCDE becomes more obvious.

5.5. Parametric sensitivity analysis

The mutation operator, scale factor F and crossover rate CR in the DCDE algorithm involve two parameters that need to be set in advance, i.e., m in Eq. (11) and n in Eq. (12), and they have a certain impact on the performance of the algorithm. In fact, the value of parameter m directly affects the changing speed of the weight parameter w, which in turn affects the specific performance of the designed mutation operator (shown in Eq. (10)) in the optimization process, while the value of parameter n directly affects the changing speed of the macro parameter GP, which in turn affects the specific value of the core parameters F and CR in the optimization process, resulting in both the parameters m and n indirectly influencing the DCDE's focus on global exploration work and local exploitation work in the optimization process. Besides, the population size NP is another significant parameter that will affect the performance of our proposed DCDE algorithm. According to Ref. (Yang et al., 2014), the population size in DE is problem-dependent. If population size is small, the population diversity will be reduced and the population has a weak exploration ability which increases the probability of premature convergence and stagnation. On the other hand, a large population size may decreases the probability of finding correct search directions and increases the risk of losing the exploitation ability. Based on the survey of Ref. (Piotrowski, 2017), since DE was introduced, many researchers have suggested that NP should be related to the problem dimensionality.

In order to determine the best setting of these three parameters (i.e., m, n and NP) and analyze the sensitivity of DCDE regarding these parameters, comparison experiments are carried out based on IEEE CEC 2017 benchmark functions by setting different parameter combination schemes. Specifically, both m and n are independently set to 1, 2 and 3; population size NP is set to equal to or twice to the problem dimension, i.e., $NP = 1 \cdot D$ or $NP = 2 \cdot D$.

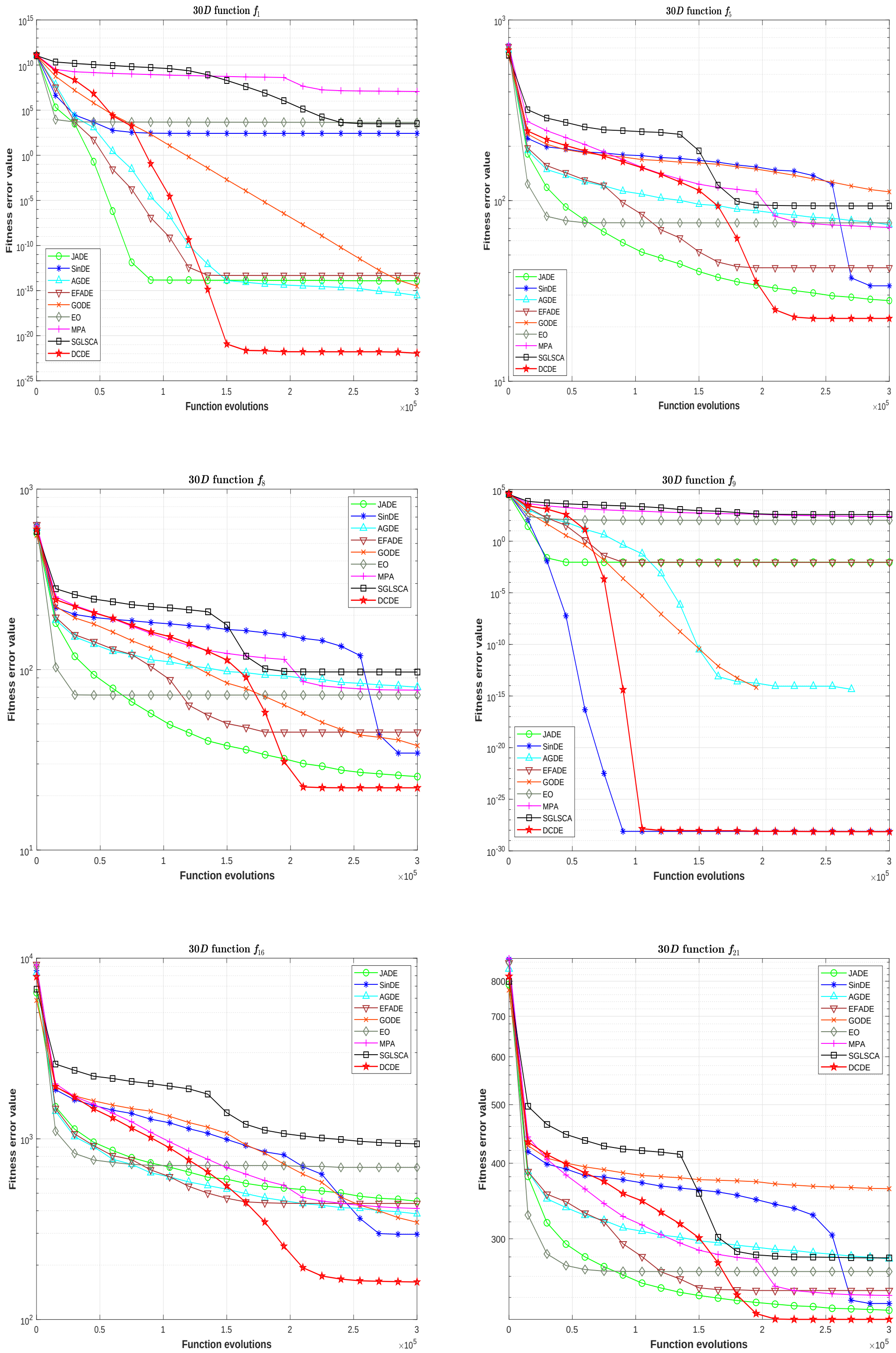


Fig. 2. The convergence characteristic of the mean fitness errors derived from proposed DCDE and other compared algorithms on 30D benchmarks functions f_1, f_5, f_8, f_9, f_{16}, f_{21}.

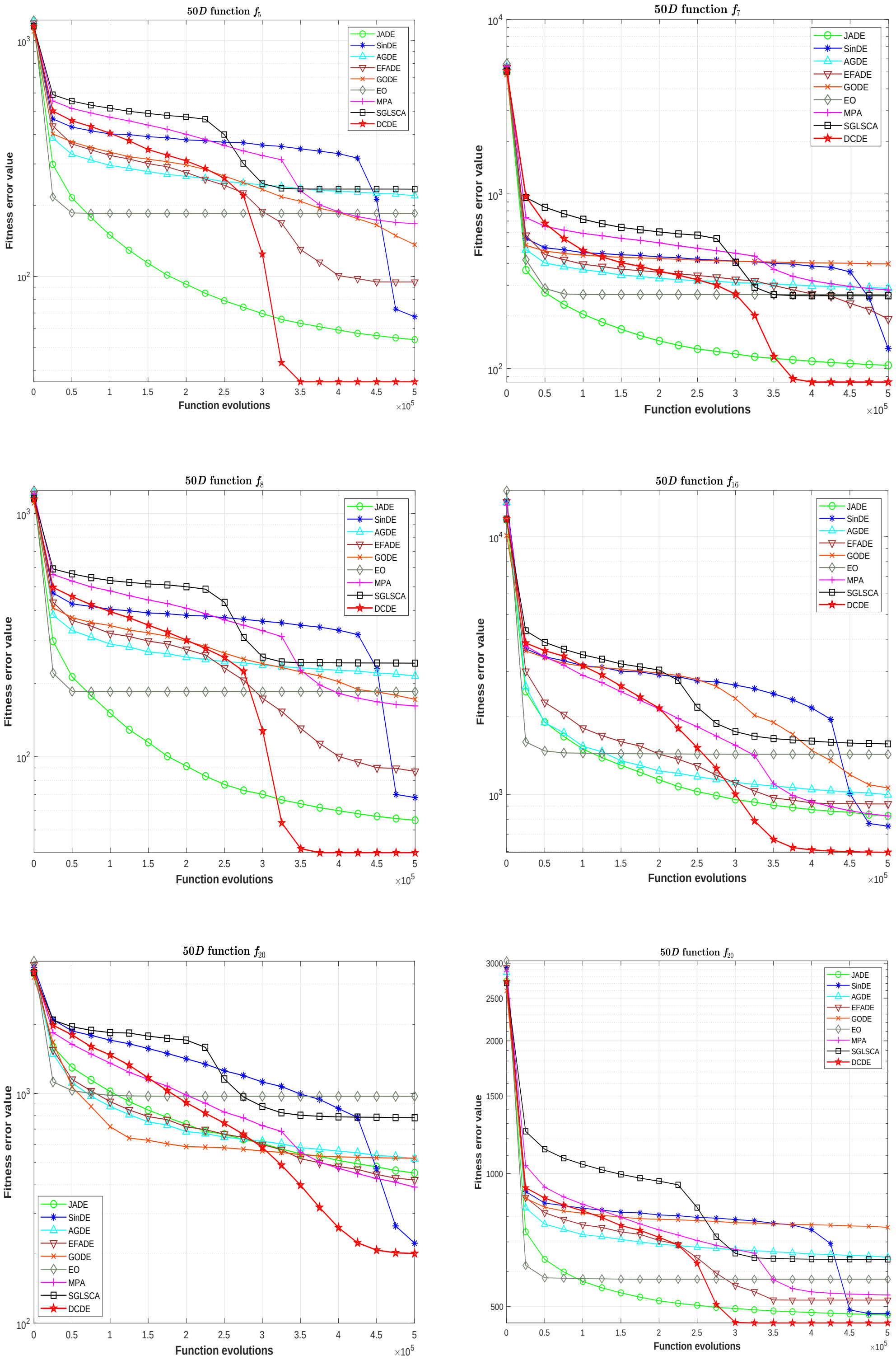


Fig. 3. The convergence characteristic of the mean fitness errors derived from proposed DCDE and other compared algorithms on $50D$ benchmarks functions f_5, f_7, f_8, f_{16}, f_{20}, f_{23}.

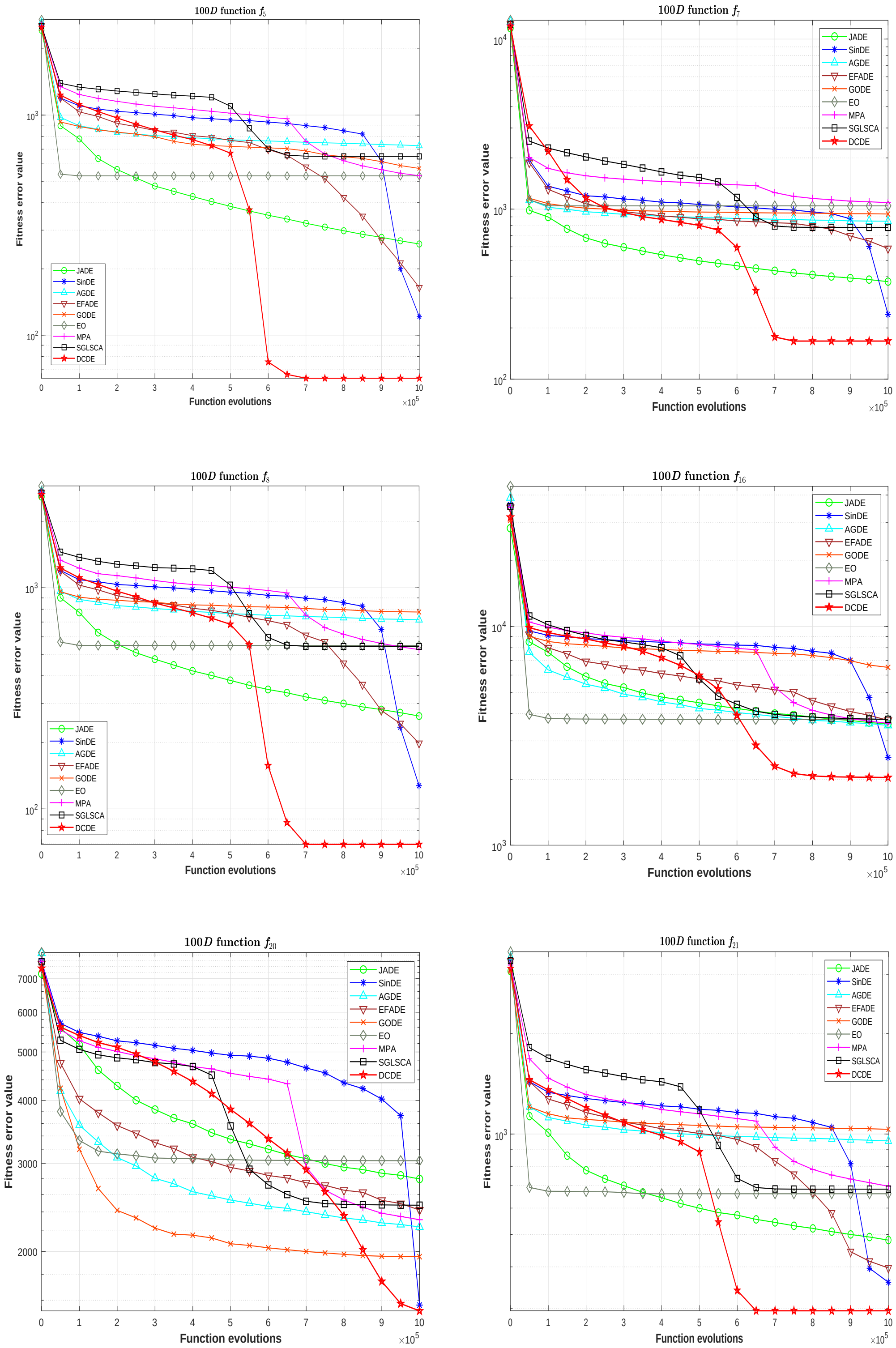


Fig. 4. The convergence characteristic of the mean fitness errors derived from proposed DCDE and other compared algorithms on $100D$ benchmarks functions f_5, f_7, f_8, f_{16}, f_{20}, f_{21}.

Table 6. Ranking results of all combination schemes by Friedman test at the 0.05 significance level for 30D, 50D and 100D functions.

Index	Combination Schemes (NP–m–n)	30D	50D	100D	Average Rank	Final Rank
1	$1 \cdot D$–1–1	12.3621	10.8793	11.0000	11.4138	13
2	$1 \cdot D$–1–2	8.7759	7.7586	**6.0000**	7.5115	4
3	$1 \cdot D$–1–3	8.6207	7.4483	7.3103	7.7931	7
4	$1 \cdot D$–2–1	12.1034	12.1379	10.6552	11.6322	14
5	$1 \cdot D$–2–2	8.4828	7.9138	8.2586	8.2184	8
6	$1 \cdot D$–2–3	10.1552	8.9138	7.6897	8.9196	11
7	$1 \cdot D$–3–1	13.5000	12.431	11.3276	12.4195	15
8	$1 \cdot D$–3–2	8.8621	8.6897	8.6379	8.7299	10
9	$1 \cdot D$–3–3	9.9828	10.1552	8.5345	9.5575	12
10	$2 \cdot D$–1–1	14.2931	13.4138	14.8276	14.1782	16
11	$2 \cdot D$–1–2	5.3103	6.2414	6.8103	6.1207	2
12	$2 \cdot D$–1–3	**4.6379**	**5.069**	6.0172	**5.2414**	**1**
13	$2 \cdot D$–2–1	14.8966	13.9828	14.7931	14.5575	17
14	$2 \cdot D$–2–2	5.9828	8.3103	8.2586	7.5172	5
15	$2 \cdot D$–2–3	6.2759	7.0517	6.0345	6.4540	3
16	$2 \cdot D$–3–1	14.3103	13.9138	15.7931	14.6724	18
17	$2 \cdot D$–3–2	6.2759	8.6379	10.1724	8.3621	9
18	$2 \cdot D$–3–3	6.1724	8.0517	8.8793	7.7011	6

Therefore, three are altogether 18 ($3 \times 3 \times 2$) combination schemes, and these schemes are discriminated by the marker "$NP - m - n$". Each scheme is incorporated into DCDE to optimize the 29 functions in IEEE CEC 2017. The non-parametric Friedman test is utilized to obtain the performance ranking for each combination scheme and the results are listed in Table 6. In this table, the best-performing combination scheme for each kind of dimension is highlighted in bold, and "Average Rank" means the average ranking value of each scheme for three different kinds of dimensions, which derives the final ranking results. From the results list in Table 6, we can see that the combination scheme ($2 \cdot D$–1–3) won first place for the 30D and 50D test problems, and second place (only a small difference from the first place) for the 100D test problem, giving it a first place overall ranking. Thus, based on the experimental results and from a comprehensive consideration, the parameter combination scheme $2 \cdot D$–1–3 is the best choice for DCDE. More specifically, setting preset parameter m in Eq. (11) to 1, preset parameter n in Eq. (12) to 3 and population size NP equaling to $2 \cdot D$ is the most appropriate parameter setting for DCDE.

6. Conclusions and Future Work

Given that appropriate mutation operators and parameter settings can effectively improve the optimization performance of the DE algorithm, we propose a novel DE variant (named DCDE) based on the basic design philosophy of balancing the global exploration ability and the local exploitation ability in the optimization process. In the DCDE algorithm, we design a base vector with dynamic change characteristics using the synergistic utility of the optimal individual and a randomly selected elite individual, and then propose a mutation operator that can take into account both global exploration work and local exploitation work. In addition, a macro parameter GP at the population level and micro parameters (ES and IS) at individual level are used to jointly regulate the scale factor F and crossover rate CR, and thus control the search focus of the DCDE algorithm at different stages of optimization process. The comparison experiments with five state-of-the-art DE variants: JADE, SinDE, AGDE, EFADE, GODE and three non-DE algorithms: EO, MPA, SGLSCA on solving $30D$, $50D$ and $100D$ test functions in IEEE CEC 2017 demonstrate that the proposed DCDE has a significantly better performance than all competitors.

In general, the DCDE algorithm proposed in this paper has a clear design philosophy, simple operation and structure, and excellent performance in continuous optimization problems with a single objective. However, the specific strategies employed in DCDE are expected to fail when dealing with discrete optimization problems, multi-objective optimization problems, or multimodal optimization problems. Our subsequent research will target modifications to the specific operations of DCDE for handling other types of optimization problems based on the basic design philosophy of balancing the global exploration ability and the local exploitation ability.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 71701187, 71704162 and 61401121, Guangxi Key Laboratory of Automatic Detecting Technology and Instruments under Grant YQ19203, Fundamental Research Funds for the Central Universities under Grant HIT NSRIF 2019083 and Sêr Cymru II COFUND Fellowship.

References

Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M., & Vrahatis, M. N. (2019). *No Free Lunch Theorem: A Review*. Springer.

Alcalá-Fdez, J., Sanchez, L., Garcia, S., del Jesus, M. J., Ventura, S., Garrell, J. M., Otero, J., Romero, C., Bacardit, J., Rivas, V. M. et al. (2009). KEEL: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, *13*, 307–318.

Alswaitti, M., Albughdadi, M., & Isa, N. A. M. (2019). Variance-based differential evolution algorithm with an optional crossover for data clustering. *Applied Soft Computing*, *80*, 1–17.

Arce, F., Zamora, E., Sossa, H., & Barrón, R. (2018). Differential evolution training algorithm for dendrite morphological neural networks. *Applied Soft Computing*, *68*, 303–313.

Bhattacharya, A., & Chattopadhyay, P. K. (2010). Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. *IEEE Transactions on Power Systems*, *25*, 1955–1964.

Biswas, P. P., Suganthan, P. N., & Amaratunga, G. A. (2017). Minimizing harmonic distortion in power system with optimal design of hybrid active power filter using differential evolution. *Applied Soft Computing*, *61*, 486–496.

Črepinšek, M., Liu, S., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, *45*, 1–33.

Das, S., Mullick, S. S., & Suganthan, P. N. (2016). Recent advances in differential evolution–an updated survey. *Swarm and Evolutionary Computation*, *27*, 1–30.

Das, S., & Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, *15*, 4–31.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*, 3–18.

Dong, C., Ng, W. W., Wang, X., Chan, P. P., & Yeung, D. S. (2014). An improved differential evolution and its application to determining feature weights in similarity-based clustering. *Neurocomputing*, *146*, 95–103.

Draa, A., Bouzoubia, S., & Boukhalfa, I. (2015). A sinusoidal differential evolution algorithm for numerical optimisation. *Applied Soft Computing*, *27*, 99–126.

Draa, A., Chettah, K., & Talbi, H. (2019). A compound sinusoidal differential evolution algorithm for continuous optimization. *Swarm and Evolutionary Computation*, *50*, 100450.

Duchanoy, C. A., Moreno-Armendriz, M. A., Urbina, L., Cruz-Villar, C. A., Calvo, H., & de J. Rubio, J. (2017). A novel recurrent neural network soft sensor via a differential evolution training algorithm for the tire contact patch. *Neurocomputing*, *235*, 71–82.

Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020a). Marine predators algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, *152*, 113377.

Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020b). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems*, *191*, 105190.

Hancer, E. (2019). Differential evolution for feature selection: a fuzzy wrapper–filter approach. *Soft Computing*, *23*, 5233–5248.

Hancer, E., Xue, B., & Zhang, M. (2018). Differential evolution for filter feature selection based on information theory and feature ranking. *Knowledge-Based Systems*, *140*, 103–119.

Islam, S. M., Das, S., Ghosh, S., Roy, S., & Suganthan, P. N. (2012). An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*, 482–500.

Jena, C., Basu, M., & Panigrahi, C. (2016). Differential evolution with gaussian mutation for combined heat and power economic dispatch. *Soft Computing*, *20*, 681–688.

Jian, J.-R., Zhan, Z.-H., & Zhang, J. (2020). Large-scale evolutionary optimization: a survey and experimental comparative study. *International Journal of Machine Learning and Cybernetics*, *11*, 729–745.

Khaparde, A., Raghuwanshi, M., & Malik, L. (2015). Distance-based analysis for base vector selection in mutation operation of differential evolution algorithm. In *Proceedings of Fourth International Conference on Soft Computing for Problem Solving* (pp. 431–442). Springer.

Li, X., & Yin, M. (2016). Modified differential evolution with self-adaptive parameters method. *Journal of Combinatorial Optimization*, *31*, 546–576.

Li, Y., Guo, H., Liu, X., Li, Y., Pan, W., Gong, B., & Pang, S. (2017). New mutation strategies of differential evolution based on clearing niche mechanism. *Soft Computing*, *21*, 5939–5974.

Liang, J., Wang, P., Guo, L., Qu, B., Yue, C., Yu, K., & Wang, Y. (2019). Multi-objective flow shop scheduling with limited buffers using hybrid self-adaptive differential evolution. *Memetic Computing*, *11*, 407–422.

Liu, Z., Ji, X., & Yang, Y. (2019). Hierarchical differential evolution algorithm combined with multi-cross operation. *Expert Systems with Applications*, *130*, 276–292.

Liu, Z., Wang, Y., Yang, S., & Cai, Z. (2016). Differential evolution with a two-stage optimization mechanism for numerical optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 3170–3177). IEEE.

Mallipeddi, R., Suganthan, P. N., Pan, Q., & Tasgetiren, M. F. (2011). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*, *11*, 1679–1696.

Mallipeddi, R., Wu, G., Lee, M., & Suganthan, P. N. (2014). Gaussian adaptation based parameter adaptation for differential evolution. In *2014 IEEE congress on evolutionary computation (CEC)* (pp. 1760–1767). IEEE.

Meng, Z., Pan, J., & Kong, L. (2018). Parameters with adaptive learning mechanism (PALM) for the enhancement of differential evolution. *Knowledge-Based Systems*, *141*, 92–112.

Meng, Z., Pan, J., & Tseng, K. (2019). PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization. *Knowledge-Based Systems*, *168*, 80–99.

Mohamed, A. W. (2017). An efficient modified differential evolution algorithm for solving constrained non-linear integer and mixed-integer global optimization problems. *International Journal of Machine Learning and Cybernetics*, *8*, 989–1007.

Mohamed, A. W., Hadi, A. A., & Jambi, K. M. (2019). Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization. *Swarm and Evolutionary Computation*, *50*, 100455.

Mohamed, A. W., & Mohamed, A. K. (2019). Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. *International Journal of Machine Learning and Cybernetics*, *10*, 253–277.

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

Omran, M. G., Salman, A., & Engelbrecht, A. P. (2005). Self-adaptive differential evolution. In *International Conference on Computational and*

Information Science (pp. 192–199). Springer.

Opara, K. R., & Arabas, J. (2019). Differential evolution: A survey of theoretical analyses. *Swarm and evolutionary computation*, *44*, 546–558.

Otani, T., Suzuki, R., & Arita, T. (2013). DE/isolated/1: a new mutation operator for multimodal optimization with differential evolution. *International Journal of Machine Learning and Cybernetics*, *4*, 99–105.

Piotrowski, A. P. (2017). Review of differential evolution population size. *Swarm and Evolutionary Computation*, *32*, 1–24.

Reddy, S. S. (2019). Optimal power flow using hybrid differential evolution and harmony search algorithm. *International Journal of Machine Learning and Cybernetics*, *10*, 1077–1091.

Sakr, W. S., El-Sehiemy, R. A., & Azmy, A. M. (2017). Adaptive differential evolution algorithm for efficient reactive power management. *Applied Soft Computing*, *53*, 336–351.

Sallam, K. M., Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2017). Landscape-based adaptive operator selection mechanism for differential evolution. *Information Sciences*, *418*, 383–404.

Storn, R., & Price, K. (1997). Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*, 341–359.

Sun, G., Lan, Y., & Zhao, R. (2019). Differential evolution with gaussian mutation and dynamic parameter adjustment. *Soft Computing*, *23*, 1615–1642.

Sun, G., Li, C., & Deng, L. (2021). An adaptive regeneration framework based on search space adjustment for differential evolution. *Neural Computing and Applications*, . doi:https://doi.org/10.1007/s00521-021-05708-1.

Sun, G., Peng, J., & Zhao, R. (2018). Differential evolution with individual-dependent and dynamic parameter adjustment. *Soft Computing*, *22*, 5747–5773.

Sun, G., Xu, G., & Jiang, N. (2020a). A simple differential evolution with time-varying strategy for continuous optimization. *Soft Computing*, *24*, 2727–2747.

Sun, G., Yang, B., Yang, Z., & Xu, G. (2020b). An adaptive differential evolution with combined strategy for global numerical optimization. *Soft Computing*, *24*, 6277–6296.

Tanabe, R., & Fukunaga, A. (2013). Success-history based parameter adaptation for differential evolution. In *2013 IEEE Congress on Evolutionary Computation* (pp. 71–78). IEEE.

Tanabe, R., & Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction. In *2014 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1658–1665). IEEE.

Tang, L., Dong, Y., & Liu, J. (2014). Differential evolution with an individual-dependent mechanism. *IEEE Transactions on Evolutionary Computation*, *19*, 560–574.

Tatsis, V. A., & Parsopoulos, K. E. (2017). Differential evolution with grid-based parameter adaptation. *Soft Computing*, *21*, 2105–2127.

Wang, C., Liu, Y., Zhang, Q., Guo, H., Liang, X., Chen, Y., Xu, M., & Wei, Y. (2019). Association rule mining based parameter adaptive strategy for differential evolution algorithms. *Expert Systems with Applications*, *123*, 54–69.

Wang, Y., Cai, Z., & Zhang, Q. (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation*, *15*, 55–66.

Wu, G., Mallipeddi, R., & Suganthan, P. (2016a). Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization. *Nanyang Technol. Univ., Singapore, Rep*, .

Wu, G., Mallipeddi, R., Suganthan, P. N., Wang, R., & Chen, H. (2016b). Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences*, *329*, 329–345.

Wu, G., Shen, X., Li, H., Chen, H., Lin, A., & Suganthan, P. N. (2018). Ensemble of differential evolution variants. *Information Sciences*, *423*, 172–186.

Yang, M., Li, C., Cai, Z., & Guan, J. (2014). Differential evolution with auto-enhanced population diversity. *IEEE Transactions on Cybernetics*, *45*, 302–315.

Yu, W., Shen, M., Chen, W., Zhan, Z., Gong, Y., Lin, Y., Liu, O., & Zhang, J. (2013). Differential evolution with two-level parameter adaptation. *IEEE Transactions on Cybernetics*, *44*, 1080–1099.

Zhang, J., & Sanderson, A. C. (2007). An approximate gaussian model of differential evolution with spherical fitness functions. In *2007 IEEE Congress on Evolutionary Computation* (pp. 2220–2228). IEEE.

Zhang, J., & Sanderson, A. C. (2009). JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, *13*, 945–958.

Zhang, T., Li, Y., & Wang, X. (2020). Gaussian prior based adaptive synthetic sampling with non-linear sample space for imbalanced learning. *Knowledge-Based Systems*, *191*, 105231.

Zhao, F., Xue, F., Zhang, Y., Ma, W., Zhang, C., & Song, H. (2018). A hybrid algorithm based on self-adaptive gravitational search algorithm and differential evolution. *Expert Systems with Applications*, *113*, 515–530.

Zhao, X., Feng, S., Hao, J., Zuo, X., & Zhang, Y. (2021). Neighborhood opposition-based differential evolution with gaussian perturbation. *Soft Computing*, *25*, 27–46.

Zhou, B., Liao, X., & Wang, K. (2019). Kalman filter and multi-stage learning-based hybrid differential evolution algorithm with particle swarm for a two-stage flow shops scheduling problem. *Soft Computing*, *23*, 13067–13083.

Zhou, W., Wang, P., Heidari, A. A., Wang, M., Zhao, X., & Chen, H. (2021). Multi-core sine cosine optimization: Methods and inclusive analysis. *Expert Systems with Applications*, *164*, 113974.

Zhu, H., He, Y., Wang, X., & Tsang, E. C. (2017). Discrete differential evolutions for the discounted {0-1} knapsack problem. *International Journal of Bio-Inspired Computation*, *10*, 219–238.