

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

# Adaptive Differential Evolution Based on Successful Experience Information

LIANZHENG CHENG<sup>1,2</sup>, YUN WANG<sup>3</sup>, CHAO WANG<sup>1</sup>, ALI WAGDY MOHAMED<sup>4,5</sup>, AND TIAOJIE XIAO<sup>6</sup>

<sup>1</sup>State Key Laboratory of Ore Deposit Geochemistry, Institute of Geochemistry, Chinese Academy of Sciences, Guiyang 550081, China

<sup>2</sup>University of Chinese Academy of Sciences, Beijing 100049, China

<sup>3</sup>School of Geophysics and Information Technology, China University of Geoscience, Beijing 100083, China

<sup>4</sup>Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza 12613, Egypt

<sup>5</sup>Wireless Intelligent Networks Center (WINC), School of Engineering and Applied Sciences, Nile University, Giza, Egypt.

<sup>6</sup>Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha 410073, China

Corresponding author: Chao Wang (E-mail: wangchao@mail.gyig.ac.cn).

This work is supported by the National Key Research and Development Program of China under Grant 2016YFC0600301, Grant 2017YFC0602500, and in part by National Natural Science Foundation of China under Grant 41904129

**ABSTRACT** As a powerful optimization algorithm for solving nonlinear, complex and tough global optimization problems, differential evolution (DE) has been widely applied in various science and engineering fields. In this paper, considering that the evolution direction of each individual is not fully exploited to guide the search process in most DE algorithms, a new DE variant (named ADEwSE), which incorporates the successful experience of evolved individuals into classic “current-to-pbest/1” mutation strategy to reduce the randomness of search direction, is proposed. Moreover, crossover matrix sorting scheme based on real crossover rate, opposition learning of crossover rate and adaptive adjustment of top  $p\%$  values are combined with the new mutation strategy to improve the global search ability. In addition, to improve the searching ability of ADEwSE further, an ADEwSE variant by introducing the linear reduction of population size is proposed. In order to verify and analyze the performance of ADEwSE, numerical experiments on a set of 29 test problems from CEC2017 benchmark for 30, 50 and 100 dimensions are executed. And the experimental results are compared with that of 21 state-of-art DE-based algorithms. Comparative analysis indicates that the ADEwSE and its improved version are competitive with these state-of-art DE variants in terms of solution quality obtained.

**INDEX TERMS** Differential evolution, successful experience, opposition learning, mutation strategy

## I. INTRODUCTION

Optimization problems exist in many real-world applications, and the key point of a proper optimizer is to obtain a satisfactory solution in a reasonable time [1]. Today, many innovative algorithms, such as Grammatical Evolution (GE) [2], Particle Swarm Optimization (PSO) [3, 4], Differential Search algorithm (DS), Artificial Bee Colony (ABC) [5], and Gaining-sharing Knowledge based algorithm (GSK) [6] etc., have been developed and successfully applied for solving nonlinear, high dimensional and complex optimization problems [7]. Differential evolution (DE) has been illustrated as a simple but efficient evolution algorithm proposed by Storn and Price [8] in 1995. Similar to all other evolution algorithms (EAs), DE employs a population-based stochastic search method to obtain acceptable results. The advantages of DE are ease of use, speediness, simple structure, and robustness. Therefore,

during the recent decades, it has drawn more and more attention in many scientific and engineering fields, such as electrical power system[9, 10], optima power flow[11, 12], neural network training[13], data mining[14], geophysical inversion [15] and image processing[16, 17], etc. Generally, DE uses mutation and recombination as search mechanisms to explore new components of potential solutions. It provides a new candidate vector by combining the components of a parent chosen from the same population with the components of the mutation vector. A candidate vector is chosen for the next generation only if it has a better fitness value than its parent vector.

Despite the DE algorithm benefits from several aforementioned attractive features, it is observed that the performance of DE is highly dependent on the configuration of mutation strategies and control parameters, such as

crossover rate ( $CR$ ), and scaling factor ( $F$ ). Actually, it's a perplexing question to choose the most appropriate mutation strategy and parameter settings since some mutation strategies are effective for global search, and some others are helpful for local exploitation, and some parameter settings can improve the convergence speed [18]. In order to obtain better results, the trial-and-error method is usually used to determine the best control parameters and mutation strategy for different problems. Actually, even for the same problem, the required parameters and strategies may vary at different stages of evolution searching. So, a considerable number of DE variants have been proposed to tune control parameters or ensemble of mutation strategies by using adaptive or self-adaptive mechanisms, including jDE [19], JADE [20], SHADE [21], CoBiDE [22], MPEDE [18], jSO [23] and so on. Note that JADE, SHADE, jSO share the same idea of maintaining a balance between exploration and exploitation abilities, and achieve it in various ways. Besides, in most of DE, the mutation direction is set by a random manner. To control the search direction more heuristically, Jia et al. [24] presented a novel and simple technique to control the search direction of jDE (jdDE). In jdDE, the search direction using the information between the trial vector and its target vector can improve the convergence rate and obtain better solution quality. Li et al. [25] formulated a new evolutionary algorithm framework by introducing the evolution path (EP) into DE. This framework offers advantages of both a distributed model (DM) and a centralized model (CM), and hence enhances the performance of DE. Lately, Zhang and Yuen [26] proposed a directional mutation (DM) operator. The proposed method can result in a larger number of generations having fitness improvement, and saving computational time than classic DE. With information sharing concept, Lee et al. [27] added the momentum of target individual with "current-to-best/1" mutation strategy, and adjusted scaling factor with fuzzy logic system. However, it is known to all that evolution direction information, such as evolution path and momentum, is rarely used to guide the search process. The reason is that to incorporate these directions into DE will increase the risk of premature convergence and is incapable of improving the exploration ability.

The motivation of this research is described as following two ideas. Firstly, the main aim is to combine the evolution direction information of individuals with descent direction of population. Considering that the "current-to-pbest/1" mutation strategy which is used widely already includes the direction information of top  $p\%$  individuals, an interesting idea is to design a novel mutation strategy by combing the evolution direction information and guided information of top  $p\%$  individuals. Secondly, as discussed by Zhang and Yuan [26], to use the evolution directions more than once will cause the algorithm to be trapped in local optima due to their greediness. Therefore, it's necessary to introduce some new parameters and modifications of crossover to improve the exploration and exploitation abilities.

In this study, a successful experience (SE) pool which includes the evolution direction of each individual is established based on the selection operator. Unlike the directional mutation algorithm [26] which maintains a difference vector pool by using the difference vector between a trial solution and its parent, the update of successful experience is related to the crossover process. It implies that the successful experience stores the evolution direction of all variables of all individuals. Obviously, a successful experience vector in SE pool may be seen as a feasible search direction which means that further fitness improvement by searching along this direction is possible. In other words, the successful experience provides past information related to the evolution search process, which can improve the ability to search the optimal solution. To utilize the successful experience information, an adaptive differential evolution based on successful experience (named ADEwSE for short) is proposed. However, merging the SE vectors into DE may lead to trap into a local optimum easily, and decrease the population diversity. Thus, for the purpose of increasing the exploration ability and keeping the diversity of the population, sorting the crossover matrix inspired by sorting the crossover rate mechanism in [28] and opposition learning of  $CR$  are proposed. In addition, as we know, DE suffers stagnation and/or premature convergence, which deteriorates significantly the performance of DE [29]. Some practical experiences have suggested that no better solution can be created by mutation strategy and crossover operation when stagnation occurs [30]. Thus, in order to relieve the problem of stagnation, a disturbance-based crossover (DX) is added, which is helpful for increasing the diversity of potential trial solutions, and improve the local exploitation ability.

In summary, the main contributions of this paper are as follows: (1) A novel mutation strategy with successful experience information is proposed. In the proposed mutation operator, the successful experience which includes the good searching direction is incorporated into classical "current-to-pbest/1" mutation strategy. (2) To strengthen the exploration of DE, a crossover matrix sorting scheme is introduced. In the proposed method, the trial vectors generated by superior individuals tend to inherit more components from their parents. (3) An opposition learning of crossover rate strategy is presented to balance the exploration and exploitation abilities dynamically. (4) A disturbance crossover (DX) is presented. The DX is executed only when an individual is stagnant. (5) An adaptive  $p$  strategy is proposed, instead of a static  $p$  value in the JADE. (6) The performance of ADEwSE is improved further by introducing linear reduction of population size strategy.

This paper is organized as follows. Section 2 introduces the classical DE algorithm. In Section 3, some recent related works are surveyed. Then the proposed algorithm is presented in detail in Section 4. In Section 5, the experimental results and comparisons with other techniques are reported and analyzed. Finally, the conclusion is given in Section 6.

## II. DIFFERENTIAL EVOLUTION

DE is a population-based heuristic search method for solving the numerical optimization problem. It has four basic processes: initialization, mutation, crossover and selection. Without loss of generality, in this work, the minimization numerical optimization problem is considered to be  $f(\mathbf{x})$ ,  $\mathbf{x} \in R^D$ , and  $D$  is the number of dimensions in the search space. DE starts with a random population of real-coded vectors representing the solution of a given problem. Every individual in population can be denoted as  $\mathbf{x}_i^G = (x_{i,1}^G, x_{i,2}^G, \dots, x_{i,D}^G)$ , where  $i = 1, 2, 3, \dots, NP$ ,  $NP$  is the population size and  $G$  is the generation times. These individuals are evolved by executing mutation, crossover and selection operation iteratively. In the following subsection, a brief summary of the original DE algorithm is provided.

### A. INITIALIZATION

Like other EAs, initial population vectors in DE are randomly generated in the search space constrained by the prescribed minimum and maximum bounds. Thus, the  $j$ th decision parameter of the  $i$ th individual is initialized via the following formula:

$$x_{ij}^0 = L_j + rand(0,1) \cdot (U_j - L_j), i \in [1, NP], j \in [1, D] \quad (1)$$

where  $rand(0,1)$  denotes a uniformly distributed real number within the range  $[0,1]$ , and  $L_j$  and  $U_j$  are the lower and upper bounds of the  $j$ th decision parameter, respectively. Therefore, in the initialization step,  $NP$  individuals are generated in the search region defined by a lower  $L$  and upper  $U$  bound, i.e.  $L = (L_1, L_2, \dots, L_D)$  and  $U = (U_1, U_2, \dots, U_D)$ .

### B. MUTATION

After the initialization, the mutation strategy is employed to generate a mutant vector  $\mathbf{v}_i^G$  for each target vector  $\mathbf{x}_i^G$  in the current population. Essentially, the formula for mutation can be regarded as an operation that combines one or multiple differential vectors with a population vector. The most widely used mutation strategies are followings [18]:

“DE/rand/1”

$$\mathbf{v}_i^G = \mathbf{x}_{r_1}^G + F(\mathbf{x}_{r_2}^G - \mathbf{x}_{r_3}^G) \quad (2)$$

“DE/rand/2”

$$\mathbf{v}_i^G = \mathbf{x}_{r_1}^G + F(\mathbf{x}_{r_2}^G - \mathbf{x}_{r_3}^G) + F(\mathbf{x}_{r_4}^G - \mathbf{x}_{r_5}^G) \quad (3)$$

“DE/best/1”

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

“DE/best/2”

$$\mathbf{v}_i^G = \mathbf{x}_{best}^G + F(\mathbf{x}_{r_1}^G - \mathbf{x}_{r_2}^G) + F(\mathbf{x}_{r_3}^G - \mathbf{x}_{r_4}^G) \quad (5)$$

“DE/rand-to-best/1”

$$\mathbf{v}_i^G = \mathbf{x}_{r_1}^G + F(\mathbf{x}_{best}^G - \mathbf{x}_{r_1}^G) + F(\mathbf{x}_{r_2}^G - \mathbf{x}_{r_3}^G) \quad (6)$$

“DE/current-to-best/1”

$$\mathbf{v}_i^G = \mathbf{x}_i^G + F(\mathbf{x}_{best}^G - \mathbf{x}_i^G) + F(\mathbf{x}_{r_1}^G - \mathbf{x}_{r_2}^G) \quad (7)$$

The parameters  $r_1, r_2, r_3, r_4, r_5$  are mutually different random integers within the range of  $[1, NP]$ , which are also different from the index  $i$ . The coefficient  $F$  is a positive controlling parameter for scaling the difference vector.  $\mathbf{x}_{best}^G$  is the best individual vector which has the best fitness value at generation  $G$ . More details can be found in [31].

### C. CROSSOVER

Following the mutation step, crossover operation is introduced to obtain trial vector  $\mathbf{u}_i^G$  by replacing certain variables of the target vector  $\mathbf{x}_i^G$  with corresponding mutant vector  $\mathbf{v}_i^G$ . In DE, there are two types of crossover scheme: exponential and binominal. We here elaborate the binominal crossover. In the binominal crossover, at least one component of the trial vector is inherited from the mutant vector. Crossover operation equation can be defined as follows:

$$u_{ij}^G = \begin{cases} v_{ij}^G, & \text{if } rand(0,1) \leq CR \text{ or } j = jrand \\ x_{ij}^G, & \text{otherwise} \end{cases} \quad (8)$$

where  $CR \in (0,1)$  is the crossover rate that controls which or how many components are inherited from the mutant vector.  $jrand$  is a randomly selected integer in the range  $[1, D]$ . Let  $\mathbf{b}_i^G$  be a binary string generated for each target vector  $\mathbf{x}_i^G$  by the following equations [32]:

$$b_{ij}^G = \begin{cases} 1, & \text{if } rand(0,1) \leq CR \text{ or } j = jrand \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Then, the binominal crossover in Eq. 8 can be modified as  $u_{ij}^G = b_{ij}^G \cdot v_{ij}^G + (1 - b_{ij}^G) \cdot x_{ij}^G$  (10) Based on Eq. 9 and Eq. 10, it's no doubt that the binary string is stochastically related to  $CR$ . According to Eq. 10, the components of trial vector are directly related to its binary string.

### D. SELECTION

In DE, a greedy mechanism is employed to select a better one between the trial vector  $\mathbf{u}_i^G$  and the target vector  $\mathbf{x}_i^G$  according to their fitness values. The selection scheme is given by:

$$\mathbf{x}_i^{G+1} = \begin{cases} \mathbf{u}_i^G, & \text{if } f(\mathbf{u}_i^G) \leq f(\mathbf{x}_i^G) \\ \mathbf{x}_i^G, & \text{otherwise} \end{cases} \quad (11)$$

In Algorithm 1, a detailed description of classic DE which utilizes “DE/rand/1” mutation operator is given.

## III. RELATED WORKS

In fact, the performance of the conventional DE algorithm mainly depends on the chosen mutation, crossover strategies and the associated control parameters. Many researchers have made a lot of contributions to the improvement of DE performance by proposing new technologies [33]. A brief overview of these techniques is discussed in the following subsection.

### A. ADAPTATION OF CONTROL PARAMETERS

The three main parameters of DE are the scaling factor  $F$ , the crossover rate  $CR$  and the population size  $NP$ , which need to be set during the initialization phase. To obtain an optimal result, it's a vital step to find out the proper setting of these control parameters. Storn and Price [34] suggested that the population size between  $5D$  and  $10D$  and the initial value of scaling factor as 0.5 are sufficient for obtaining a satisfactory solution. Later, Gämperle [35] recommended that choosing the population size within the range of  $[3D, 8D]$ , the scaling factor as 0.6 and crossover rate in  $[0.3, 0.9]$  is a proper choice. However, Ronkkonen [36] concluded that taking  $F$  as 0.9 initially is suitable for making a compromise between convergence speed and convergence rate. For the crossover rate, they suggested that  $CR$  with a value between  $[0, 0.2]$  is proper for separable objective functions, and a value of  $CR$  between  $[0.9, 1]$  is proper when the objective functions are not separable and multimodal. In addition, Zielinski [37] suggested that setting  $F$  and  $CR \geq 0.6$  in many cases is helpful for obtaining good results. From the above

**Algorithm 1** The DE algorithm with "DE/rand/1/bin"

```

01: Begin
02:  $G = 1$ 
03: Generate a random initial population  $\mathbf{x}_i^G, i \in [1, NP]$ 
04: Evaluate  $f(\mathbf{x}_i^G) \forall i, i = 1, 2, 3, \dots, NP$ 
05: While the halting criterion is not satisfied Do
06:   For  $i = 1$  to  $NP$  Do
07:     Select uniform randomly  $r1 \neq r2 \neq r3 \neq i \in [1, NP]$ 
08:      $jrand = randint(1, D)$ 
09:     For  $j = 1$  to  $D$  Do
10:       If  $rand(0,1) \leq CR$  or  $j = jrand$  Then
11:          $u_{i,j}^G = x_{r1,j}^G + F \cdot (x_{r2,j}^G - x_{r3,j}^G)$ 
12:       Else
13:          $u_{i,j}^G = x_{i,j}^G$ 
14:       End If
15:     End For
16:     End For
17:     For  $i = 1$  to  $NP$  Do
18:       Evaluate the offspring  $\mathbf{u}_i^G$ 
19:       If  $f(\mathbf{u}_i^G) \leq f(\mathbf{x}_i^G)$  Then
20:          $\mathbf{x}_i^{G+1} = \mathbf{u}_i^G$ 
21:       Else
22:          $\mathbf{x}_i^{G+1} = \mathbf{x}_i^G$ 
23:       End If
24:     End For
25:      $G = G + 1$ 
26:   End While
27: End

```

ve, it can be observed that various suggestions for parameter settings have been proposed according to the manual parameter tuning method. Anyway, these empirical parameter settings are quite different from each other, and it implies that choosing the most proper parameter settings is usually difficult when the properties of optimization are unknown, and

sufficient experimental justifications are lacked [33]. Thus, some adaptive or self-adaptive techniques are designed to produce the control parameters to avoid manually tuning the control parameters.

Liu and Lampinen proposed fuzzy adaptive differential evolution with a fuzzy logic controller in [38] which  $F$  and  $CR$  are adjusted based on the relative fitness value and individuals of successive generations. Likewise, Brest [23] proposed an efficient self-adaptation scheme (jDE) by encoding  $F$  and  $CR$  into each individual, and adapted them by means of evolution. Qin and Suganthan [39] introduced a memory-based self-adaptive differential evolution (SaDE) algorithm. In SaDE, the authors allow  $F$  to take different values in the range  $(0, 2]$  with a normal distribution  $(0.5, 0.3)$  for each individual. Meanwhile, the new values of  $CR$  are produced based on the successful values of the crossover rate stored in a memory bank. Zhang and Sanderson [24] presented an adaptive scheme to update the control parameters to proper values in JADE. For each target vector, the scaling factor  $F$  and crossover rate  $CR$  are drawn from a Cauchy distribution  $C(\mu_F, 0.1)$  and a normal distribution  $N(\mu_{CR}, 0.1)$  respectively where  $\mu_F$  and  $\mu_{CR}$  are the respective mean values of distribution. Instead of sampling  $F$  and  $CR$  values from the most recent successful mutation information in JADE. Tannabe and Fukunaga [25] used historical memory archives  $M_F$  and  $M_{CR}$  to generate new  $F, CR$  pairs by directly sampling the parameter space. Wang and Li [15] proposed a bimodal distribution parameter setting scheme to control parameters of mutation and crossover operators, and balance the exploration and exploitation ability of DE. Yu and Shen [40] introduced a two-level parameter adaptation scheme for DE: adjusting dynamically the population-level scale factor  $F_p$  and crossover rate  $CR_p$  according to the exploration and exploitation statuses, and adapting the individual-level control parameters  $F_i$  and  $CR_i$  based on the fitness value of the individual as well as the distance to the global best individual in the population. Mohamed et al. [33] presented a novel and effective adaptation scheme, in which crossover rate is adaptively selected from pre-determined two sets based on their experiences of generating promising solutions in the process of evolution, and scale factor is independently generated according to a uniform distribution in  $(0.1, 1)$  at each generation. Zhou et al. [28] modified JADE by sorting crossover rate strategy. The individuals with small fitness values in the population are assigned with small  $CR$  values. Cui et al. [41] proposed a new self-adaptive adjustment for differential evolution, called ADEDE. In the ADEDE, a parameter population is established for the solution population and updated from generation to generation under the basic principle that a good parameter individual is more likely to survive at high risk, while the bad parameter individuals should have a high probability to learn from the good parameter individuals. Sun [42] adjusted the control parameters by applying the fitness information of each

individual and a dynamic fluctuation rule in IDDE algorithm and obtained a better balance between the exploration and exploitation ability.

Apart from  $F$  and  $CR$ , adaptive adjustment of population size  $NP$  has also received much attention in recent years since the population size significantly impacts the convergence rate of evolution algorithm. The modified jDE algorithm, called dynNP-DE [43], is initialized with a large population size ( $10D$ ) and after each 25% of allowed function calls  $NP$  is reduced by half, to  $1.25D$  in the last quarter of the run. Tirronen and Neri [44] introduced a population adaptation mechanism based on fitness diversity which is calculated by some proposed measures. Zhu et al. [45] in their ATPS-DE algorithm introduced a new  $NP$  adaptation strategy in which a status monitor counts the number of the recent consecutive generations with or without improvement of the best solution. Tanabe and Fukunaga [46] further improved SHADE with a linear population size reduction, which continuously reduces the population size according to a linear function. A detailed overview of population size adaptation can be found in [47].

### B. MODIFICATIONS OF THE MUTATION STRATEGY, CROSSOVER SCHEME AND SELECTION OPERATOR

To improve the performance of the DE algorithm, researchers have made some attempts to design new kinds of mutation strategy. Das et al. [48] proposed a physical neighbor-based mutation operator, which contains two parts: local neighborhood-based mutation and global neighborhood-based mutation, to balance the exploitation and exploration abilities of DE. Zhang and Sanderson [24] improved the optimization performance of DE by implementing a new mutation strategy “current-to-pbest/1” with the optional external archive. Due to the external archive and the manner of updating control parameters, JADE showed significant performance improvement for relatively high-dimensional problems. Mohamed [49] presented a new triangular mutation rule based on the convex combination vector of the triangular to enhance the local search tendency, to improve the global exploration ability and to accelerate the DE algorithm. Mohamed [33] introduced an adaptive guided mutation rule where the difference vector is formed by two randomly chosen vectors of the top and bottom  $100p\%$  individuals in the current population to maintain the balance between the exploration and exploitation abilities during the process of evolution. Islam et al. [50] proposed a modified DE with p-best crossover, called MDE\_pBX. The novel mutation strategy selects the best individual from a dynamic group of  $100 \times q\%$  randomly in the current population. Inspired by collective intelligence, Zheng et al. [51] proposed a new mutation strategy, referred to as “current-to-ci.mbest/1”, to achieve better search capability. A collective vector  $\mathbf{x}_{ci\_mbest}^G$  is a linear combination of the top-ranking  $m$  ( $m$  is a random integer for  $\mathbf{x}_i^G$  and  $m \in [1, i]$ ) population vectors in current population with fitness values better than or equal to  $\mathbf{x}_i^G$ . He and Zhou [52] proposed a new mutation scheme named the

“current-to-better/1” by embedding CMA-ES into the “current-to-pbest/1” strategy. This scheme strengthens both exploration and exploitation of DE by guiding the search with a Gaussian distribution. Peng et al. [53] proposed a random neighborhood-based mutation strategy “neighbor/1”, and adopted the best one in the neighbors as the base vector. According to the experimental results, neighborhood information is capable of making a proper balance between global and local search in the searching process of DE.

In addition to the modification of mutation and the adjustment of control parameter settings, enhancements in crossover has also been investigated. To enhance the searching capability of DE, the orthogonal crossover was introduced by Wang et al. [54], and adapted with a quantization technique proposed by Leung and Wang [55] who used it with genetic algorithm (GA). Besides, considering that binomial crossover widely used in DE is not rotationally invariant, Guo and Yang [56] suggested that a rotationally invariant crossover is more likely to generate better trial vector. In addition, Wang et al. [22] presented a covariance matrix learning strategy to relieve the dependence of DE on the coordinate system, and improve the capability of DE to solve problems with high variable correlation. Another modification of crossover proposed by Islam et al [50] was incorporating a greedy parent selection strategy with binomial crossover strategy, named p-Best crossover. The p-Best crossover is beneficial to improve the performance of DE by a large magnitude.

Concerning the selection operator, Hoang [57] introduced a new probabilistic similarity-based method to improve the DE’s selection process, and preserve the population diversity. Wang and Gao [58] designed a local selection operator by decomposing the high dimensional problem into some subcomponents and assigning a local fitness function to evaluate each subcomponent. Recently, Addawe [59] presented a mathematical analysis of DE-based algorithm with SA-like selection operator. The SA-like selection operator is capable of enhancing the performance of conventional differential evolution algorithm by increasing the variance of the population to avoid obtaining a local optimum.

### IV . ADAPTIVE DIFFERENTIAL EVOLUTION WITH SUCCESSFUL EXPERIENCE

In this section, the proposed ADEwSE algorithm, a JADE variant, which embedded successful experience vector into the “current-to-pbest/1” mutation rule. The main innovations of this work are presented as follows in detail.

1) In classic differential evolution mutation strategies, each difference vector for perturbation is generated by a few individuals in the population. However, this process does not take the history evolution direction information into consideration. In this section, a novel mutation scheme by embedding a successful experience vector with “current-to-pbest/1” mutation strategy is designed to improve the search ability of DE.

2) Sorting crossover matrix technique is added. The principle of sorting crossover matrix is similar to the mechanism of sorting crossover rate. Instead of assigning better individuals with smaller  $CR$  values, the crossover matrix is sorted according to real crossover rate, then the better individual will be assigned a binary string which has smaller real crossover rate value.

3) Opposition learning of crossover rate scheme is proposed to dynamically change the crossover rate, help switch the exploration and exploitation phase of DE adaptively, and maintain the diversity of population.

4) During the evolution search process, some target individuals in the DE population may stop generating better solutions. To solve this problem, a disturbance-based crossover is applied to the mutant vector  $\mathbf{v}_{i,G}$  and disturbance vector  $\mathbf{x}_{d,i,G}$ .

5) At each generation, each individual  $\mathbf{x}_i$  has an associated  $p_i$  generated randomly from a normal distribution.

#### A. "CURRENT-TO-PBEST/1" MUTATION STRATEGY

Since the DE/current-to-best/1 strategy Eq. 7 incorporates the best solution information in the search process, it can guide the mutant vectors towards the best member of the population. However, it has been shown that this strategy may lead to premature convergence and poor performance on multimodal problems. Thus, Zhang and Sanderson [24] proposed a "current-to-pbest" mutation strategy, which can be described as follows:

$$\mathbf{v}_i^G = \mathbf{x}_i^G + F_i(\mathbf{x}_{pbest}^G - \mathbf{x}_i^G) + F_i(\mathbf{x}_{r1}^G - \mathbf{x}_{r2}^G) \quad (12)$$

where  $\mathbf{x}_{pbest}^G$  is randomly selected among the top  $100 \times p\%$  ( $p \in [0,1]$ ) individuals in the  $G$ th generation.  $F_i$  is a scale factor associated with individual  $\mathbf{x}_i$ . The greediness of "current-to-pbest/1" is controlled by parameter  $p$  which can balance exploitation and exploration abilities in the process of evolution.

In the original JADE, the parameter  $p$ , which is used to control the greediness of the "current-to-pbest/1" mutation strategy, is static during the whole evolution process and set manually. To improve the exploration ability of SHADE, each individual  $\mathbf{x}_i$  has an associated  $p_i$ , which is selected randomly from a uniform distribution. In this paper, the  $p_i$  value is set adaptively for each individual  $\mathbf{x}_i$  based on a Gaussian perturbation scheme. The formula of generating of  $p_i$  is:

$$p_i = randn_i(\mu_p, 0.1) \quad (13)$$

where mean value  $\mu_p$  is initialized to be 0.5. In case a value for  $p_i$  outside of  $[2/NP, 0.5]$  is generated, it is truncated to the limit value. At the end of each generation,  $\mu_p$  is updated as follows:

$$\mu_p = (1 - c_p)\mu_p + c_p \cdot \text{mean}_A(S_p) \quad (14)$$

where  $S_p$  is the set of successful  $p$  values in the current generation;  $c_p$  is a learning rate in the range of  $(0,1)$ ;  $\text{mean}_A(x)$  means arithmetic mean. According to Eq. 13, the greediness is adjusted by the parameter  $\mu_p$  instead of  $p$

value. Thus,  $\mu_p$  is set to be large at the beginning of the evolutionary process to maintain the diversity of the population. It is noteworthy that the update formula for  $\mu_p$  in Eq. 14 uses an arithmetic mean, this biases  $\mu_p$  to converge to a small value. In other words, as the individuals converge, the parameter  $\mu_p$  will decrease to improve the local exploitation ability.

#### B. DISTURBANCE BASED CROSSOVER

To give differential evolution algorithm a chance of escaping from false optima and improve the efficiency of search, a disturbance-based crossover (DX) technique is incorporated into ADEwSE when a target vector is stagnant.  $ST$  denotes a stagnation memory vector which records the consecutive unsuccessful updates for each vector  $\mathbf{x}_i^G$ . The contents of  $ST$  are updated as follows:

$$ST_i^{G+1} = \begin{cases} 0, & \text{if } f(\mathbf{u}_i^G) \leq f(\mathbf{x}_i^G) \\ ST_i^G + 1, & \text{otherwise} \end{cases} \quad (15)$$

When the  $i$ th target vector is stagnant, i.e.,  $ST_i^G \geq T$ , where  $T$  is a threshold, the DX is realized as follows:

$$\mathbf{u}_{i,j}^G = \begin{cases} \mathbf{v}_{i,j}^G, & \text{if } rand_j(0,1) < CR_i \text{ or } j = jrand \\ \mathbf{x}_{d,j}^G, & \text{otherwise} \end{cases} \quad (16)$$

where  $\mathbf{x}_{d,j}^G$  is the  $j$ th component of disturbance vector  $\mathbf{x}_d^G$  which is defined as:

$$\begin{aligned} \mathbf{x}_d^G &= \mathbf{x}_{r_p}^G + \delta F \cdot (\mathbf{x}_{r_p}^G - \mathbf{x}_i^G) \\ \delta F &= -0.1 + 0.2 \cdot rand(0,1) \end{aligned} \quad (17)$$

where  $r_p \neq i$  is selected randomly from the better individuals than target vector  $i$ ;  $\delta F$  generated by a uniform distribution in the range of  $(-0.1, 0.1)$  can control the disturbance direction and size. Note that, for the global best individual, the disturbance-based crossover would not be conducted. The binary form of DX can be written as:

$$\mathbf{u}_{ij}^G = b_{ij}^G \cdot \mathbf{v}_{ij}^G + (1 - b_{ij}^G) \cdot \mathbf{x}_{d,ij}^G \quad (18)$$

In Section V, numerical experiments will be presented to show a comprehensive study of the effectiveness of DX. According to Eq. 15 – 18, it can be seen that the DX crossover operation introduces the information of better individuals, so the exploitation ability can be enhanced.

#### C. A NOVEL MUTATION STRATEGY BASED ON SUCCESSFUL EXPERIENCE INFORMATION

In the selection phase of differential evolution, the objective function values of all trial vectors are compared to corresponding target vectors in the current population. Therefore, we can give a formal definition of a *successful trial vector* at a generation.

**Definition 1.** (*successful trial vector*): without loss of generality, in a minimization problem, if the trial vector  $\mathbf{u}_i^G$  produced by its target vector  $\mathbf{x}_i^G$  survives in the selection step according to Eq. 11, then  $\mathbf{u}_i^G$  is a *successful trial vector*.

**Definition 2.** (*successful experience vector*): suppose for one index  $i$ , if  $\mathbf{u}_i^G$  is a successful trial vector, the *successful experience* (SE) vector  $\delta \mathbf{s}_i^{G+1}$  is defined as:

$$\delta s_{i,j}^{G+1} = \begin{cases} u_{i,j}^G - x_{i,j}^G, & \text{if } b_{ij}^G = 1, f(\mathbf{u}_i^G) \leq f(\mathbf{x}_i^G) \\ \delta s_{i,j}^G, & \text{otherwise} \end{cases} \quad (19)$$

According to definition 2, the *successful experience* vector  $\delta s_i^G$  is a difference vector determined by trial vector  $\mathbf{u}_i^G$  and its parent. It is noted that the *experience direction vector* is not updated when the target vector  $\mathbf{x}_i^G$  fails to generate a better offspring. Besides, from the definitions above, the *successful experience* vector of each individual records the historical evolution direction information, thus it can improve the local exploitation of differential evolution. Now, a successful experience memory  $\delta s$  can be established for all individuals, and updated according to Eq.19. The manner of updating the successful experience memory is presented in Algorithm 2.

**Algorithm 2** The description of updating successful experience memory

```

01: For  $i = 1$  to  $NP$  Do
02:   If  $f(\mathbf{u}_i^G) \leq f(\mathbf{x}_i^G)$  Then
03:     Calculate the successful experience vector  $\delta s_i^G$ 
        according to Eq. 18
04:   End If
05: End For
    
```

However, from the Algorithm 2, no trial vector could be utilized to calculate a successful experience vector when  $G = 0$ . Thus, at the beginning of the search, a randomly selected individual that has different index with current individual  $i$  is used to initialize the successful experience vector memory based on Algorithm 3.

**Algorithm 3** The method of initializing  $\delta s$

```

01: For  $i = 1$  to  $NP$  Do
02:   Select uniform randomly  $r1 \neq i \in [1, NP]$ 
03:   If  $f(\mathbf{x}_{r1}^G) \leq f(\mathbf{x}_i^G)$  Then
04:      $\delta s_i^G = \mathbf{x}_{r1}^G - \mathbf{x}_i^G$ 
05:   Else
06:      $\delta s_i^G = \mathbf{x}_i^G - \mathbf{x}_{r1}^G$ 
07:   End If
08: End For
    
```

Obviously, although searching solution along these experience directions may lead to fitness improvement with a higher probability compared with the difference vectors generated by randomly selected individuals, the risk of trapping into local best points is also increased. Hence, a novel mutation strategy is introduced denoted “current -to-pbest with SE /2”:

$$\mathbf{v}_i^G = \mathbf{x}_i^G + F_i(\mathbf{x}_{pbest}^G - \mathbf{x}_i^G) + F_i(\mathbf{x}_{r1}^G - \mathbf{x}_{r2}^G) + K_i \Lambda_i \cdot \delta s_{rd}^G \quad (20)$$

where  $\delta s_{rd}^G$  is a successful experience vector selected randomly from experience memory;  $rd \in [1, NP]$  is the index of chosen successful experience vector;  $r1 r2$  are selected randomly from the current population;  $K_i \in (0,1)$  is a inertial wight factor used to control the effect of successful experience vector;  $\Lambda_i \in [0, 1]$  is a new scale factor that adjusts the size and direction of successful experience vector. one can observe from Eq. 20 that the incorporation of the successful experience information in the mutation strategy has two benefits. Firstly, the successful experience vector is

calculated according to the successful search direction of an individual, a better solution would be found with higher probability following the direction indicated by  $\delta s_{rd}^G$ . Secondly, the target vectors are not always attracted toward the pbest solutions found so far which helps avoid premature convergence at local optima.

#### D. PARAMETER ADAPTION SCHEMES

In JADE,  $CR_i$  is generated at the beginning of each generation according to the following equation:

$$CR_i = randn_i(\mu_{CR}, 0.1) \quad (21)$$

where  $randn_i(\mu_{CR}, 0.1)$  returns a random value with a normal distribution  $(\mu_{CR}, 0.1)$ . As long as the values generated by Eq. 21 are out of the range  $[0,1]$ , a repaired scheme would be activated. if  $CR_i > 1$  or  $CR_i < 0$ , it is truncated to  $[0,1]$ . At the end of each generation, the mean value  $\mu_{CR}$  is updated as follows:

$$\mu_{CR} = (1 - c)\mu_{CR} + c \cdot mean_A(S_{CR}) \quad (22)$$

where  $S_{CR}$  is the set of successful  $CR$  values in the current generation;  $mean_A(x)$  is the arithmetic mean; and the learning rate  $c$  is a positive constant in the range  $(0,1)$ .

The generation of  $F_i$  is similar to  $CR_i$  with the following manner,

$$\begin{cases} F_i = randc_i(\mu_F, 0.1) \\ \mu_F = (1 - c)\mu_F + c \cdot mean_L(S_F) \\ mean_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F} \end{cases} \quad (23)$$

where the  $mean_L$  is Lehmer mean;  $S_F$  is the set of any  $F_i$  that helps the mutation strategy to generate better offspring;  $randc_i(\mu_F, 0.1)$  means Cauchy distribution with location parameter  $\mu_F$  and scale parameter 0.1. After generation of  $F_i$ , If  $F_i \leq 0$ , it is regenerated by the execution of Eq.23 until an effective value is obtained; if  $F_i > 1$ , it is truncated to 1.0.

In Eq. 20, an inertial weight parameter  $K$  is introduced to adaptively control the effect of *successful experience* vector, and defined according to following the equations:

$$\begin{cases} K_i = 0.95^{mean_A(ST)} B_i \\ B_i = randn_i(\mu_B, 0.1) \\ \mu_B = (1 - c)\mu_B + c \cdot mean_{pow}(S_B) \\ mean_{pow}(S_B) = \left( \frac{1}{|S_B|} \sum_{B \in S_B} B^n \right)^{\frac{1}{n}} \end{cases} \quad (24)$$

where  $randn_i(\mu_B, 0.1)$  generates  $B_i$  by applying a normal distribution with a mean value  $\mu_B$  and a standard deviation 0.1;  $S_B$  is the set of successful B values, and  $mean_{pow}$  is the power mean with  $n=1.5$ ;  $|S_B|$  denotes the cardinality of the set  $S_B$ . After generation of  $B_i$ , if  $B_i < 0$  or  $B_i > 1$ ,  $B_i$  will be regenerated. To keep the exploration ability at the beginning of searching process,  $\mu_B$  is set to 0. From Eq. 24, the weight of successful experience vector  $K$  would be reduced if certain individuals stagnate in the population. In addition, to ensure that better individuals have more chance to explore its neighborhood domain, the B values should be

sorted, then smaller B values are assigned to those individuals with better fitness values.

Apart from parameter  $K$ , a new scale factor  $\Lambda$  is used to control the size and direction of successful experience vector. For each individual,  $\Lambda_i$  is generated by

$$\begin{cases} \Lambda_i = \text{sgn}(\Gamma_i - \text{rand}(0, 1)) \cdot A_i \\ \Gamma_i = \text{randn}_i(\mu_\Gamma, 0.1) \\ A_i = \text{randc}_i(\mu_A, 0.1) \\ \mu_\Gamma = (1 - c)\mu_\Gamma + c \cdot \text{mean}_{\text{pow}}(S_\Gamma) \\ \mu_A = (1 - c)\mu_A + c \cdot \text{mean}_L(S_A) \\ \text{mean}_{\text{pow}}(S_\Gamma) = \left( \frac{1}{|S_\Gamma|} \sum_{\Gamma \in S_\Gamma} \Gamma^n \right)^{\frac{1}{n}} \\ \text{mean}_L(S_A) = \frac{\sum_{A \in S_A} A^2}{\sum_{A \in S_A} A} \end{cases} \quad (25)$$

where  $\Gamma_i$  and  $A_i$  are direction controller and scale factor, respectively;  $\text{sgn}(x)$  is a signum function which is often defined simply as 1 for  $x > 0$  and  $-1$  for  $x < 0$ .  $|S_\Gamma|$  is the cardinality of the set  $S_\Gamma$ . If  $\Lambda_i < 0$  or  $\Lambda_i > 1$ ,  $\Lambda_i$  is truncated into an interval  $[0, 1]$ . For parameter  $A_i$ , If  $A_i > 1$ , it is truncated to 1; if  $A_i \leq 0$ , it is regenerated by using Eq. 25. At the early stage of searching process, if  $\mu_\Gamma$  is close to 1, then  $\Gamma$  generated from Gaussian distribution is more than 0.5 in most cases. It means that  $\text{sgn}(\Gamma - \text{rand}(0,1))$  is equal to 1 in the majority of cases, then the mutant vector  $\mathbf{v}_i^G$  tend to evolve along with the directions of successful experience which have generated better offspring in the last generation. In our view, a larger initial value of  $\mu_\Gamma$  is not conducive to balance the exploration and exploitation abilities. In this work,  $\mu_\Gamma$  is set to 0.5 at the initial stage of algorithm. With regard to parameter  $\Lambda$ , it's controlled by parameter  $\mu_A$ . To weaken the risk of premature convergence, it's necessary to decrease the scale factor of successful experience vector, then improve the randomness of algorithm. Hence, similar to direction controller  $\Gamma$ , it's proper to set the location parameter  $\mu_A=0$  at  $G = 1$ .

### E. SORTING CROSSOVER MATRIX SCHEME

In differential evolution, the crossover rate controls how many components of the newly generated trial vector are from the mutant vector [21]. In the crossover operation, retaining the components of better individuals into its trial vectors will increase the probability of the generation of better solutions. Zhou et al. [28] introduced a modification to JADE by sorting the crossover rate. In their research work,  $CR$  set and population are sorted according to their values, a smaller  $CR$  value is then reassigned to an individual with smaller fitness value. Based on the experimental results conducted, sorting crossover rate can enhance the global search ability of JADE. In this section, we introduce another sorting strategy based on the real crossover rate.

At each generation, a crossover matrix  $\mathbf{b}^G$  is generated by using Eq. 10

. So, the real crossover rate is calculated as:

$$CR_i^G = \frac{\sum_{j=1}^D b_{i,j}^G}{D} \quad (26)$$

In the proposed sorting crossover matrix scheme,  $\mathbf{b}$ ,  $CR$  are sorted according to the  $CR'$  values, then the population is sorted in view of fitness values. Finally, the  $\mathbf{b}$ ,  $CR$  with smaller  $CR'$  are reassigned to an individual with better fitness value. The procedure of sorting crossover matrix technique is shown in Algorithm 4.

#### Algorithm 4 Procedure of crossover matrix sorting

- 01: Generate  $CR_i$  for each target vector  $\mathbf{x}_i$  by using Eq. 9
- 02: Get crossover matrix  $\mathbf{b}$  according to Eq. 9
- 03: Calculate the real crossover rate  $CR'_i$  using Eq. 26
- 04: Sort  $CR$  values and  $\mathbf{b}$  matrix according to  $CR'$  values
- 05: Reassign the crossover matrix to each individual on the basis of their fitness values

### F. OPPOSITION LEARNING OF CROSSOVER RATE

The crossover rate practically plays a vital role for keeping population diversity [18]. In fact, if  $CR$  is small, the components of the trial individual inherit more information from the parent vector than its mutant vector. It implies that small  $CR$  values increase the possibility of stagnation that may weaken the exploration ability of the algorithm. On the other hand, larger  $CR$  values will increase the population diversity. Nevertheless, it will reduce the stability of search in the evolution process. According to above considerations, and inspired by opposition learning mechanism, the opposition learning of  $CR$  is proposed in this section, which can switch the larger  $CR$  and smaller ones dynamically and adaptively.

During the iteration, if an offspring can replace its parent, it means that some better gene information has been obtained. Now, if the  $CR_i$  is high, a small  $CR_i$  at the next generation is helpful to maintain the better genes to survive and more likely to generate better offspring. On the contrary, if the  $CR_i$  is small, a large  $CR_i$  will be used, then, it may increase the population diversity. According to the discussion above, opposition learning of crossover rate (OLCR) technique is proposed. At the beginning of generation, we set OLCR value  $\tilde{CR}_i = CR_i$ , then  $\tilde{CR}_i$  is updated according to the following equations:

$$\begin{cases} \tilde{CR}_i^G = \begin{cases} \max(0.02, CR_r), & \text{if } f(\mathbf{u}_i^{G-1}) \leq f(\mathbf{x}_i^{G-1}) \\ CR_i^G, & \text{otherwise} \end{cases} \\ CR_r = 1 - \tilde{CR}_i^{G-1} \end{cases} \quad (27)$$

According to the Eq. 27, in the case of that a target vector  $\mathbf{x}_i^G$  is replaced by its trial vector  $\mathbf{u}_i^G$ , the crossover process in next generation will be controlled by opposition crossover rate  $\tilde{CR}_i$ . This new crossover operation is defined by the following equations:

$$b_{i,j}^G = \begin{cases} 1, & \text{if } \text{rand}_j \leq \tilde{CR}_i^G \text{ or } j = j_{\text{rand}} \\ 0, & \text{otherwise} \end{cases} \quad (28)$$

It should be noted that OLCR technique modifies the crossover matrix only if a better solution is obtained by the evolution search. In addition, the  $\tilde{CR}$  values are identical to



$CR$  values at the beginning of the searching, and then updated the contents of OLCR using Eq. 28. The detailed pseudo-code of OLCR is shown in Algorithm 5.

**Algorithm 5** Opposition learning of crossover rate (OLCR) scheme

01: Calculate the  $\widetilde{CR}_i^G$  using Eq. 27  
 02: Regenerate the binary string  $\mathbf{b}_i^G$  if a trial vector is better than its parent in the last generation

**G. OVERALL IMPLEMENTATION**

By incorporating the successful experience vector into the “current-to-pbest/1” mutation strategy, and the self-adaptive control parameter setting method introduced above, we can now present the entire ADEwSE framework as follows.

(1) Initialization:

Set generation time  $G = 0$ , randomly generate an initial population in the feasible solution region, and initialize successful experience vector memory according to Algorithm 3. The other relevant parameters are given as follows:  $NP=0.5$ , set  $\mu_{CR}=0.5$  (the initial mean value of normal distribution for crossover rate  $CR$ ),  $\mu_F=0.5$  (the initial location parameter of Cauchy distribution for scale factor  $F$ ),  $\mu_A=0$  (the initial location parameter of Cauchy distribution for scale factor  $\alpha$ ),  $\mu_B=0$  (the initial mean value of normal distribution for weight parameter  $\beta$ ),  $\mu_p=0.5$  (the initial mean value of normal distribution for the proportion of the  $p$ best individual).

(2) Mutation

For every individual  $\mathbf{x}_i^G$ , a mutant vector is generated by Eq. 20. In the early stage of evolution process, the mean value  $\mu_B$  and location parameter  $\mu_A$  go close to zero, then a small parameter  $\Lambda$  is generated. It means that a small step along the direction of successful experience is conducted. In this way the diversity of the population could be maintained. In addition, a larger  $p$  value is beneficial to enlarge the search region, and improve the exploration ability at the beginning.

(3) Crossover

For each target vector  $\mathbf{x}_i^G$  and its mutant vector  $\mathbf{v}_i^G$ , a trial vector  $\mathbf{u}_i^G$  is generated by a binary crossover operation expressed as Eq. 10 in Section II. The values of  $CR$  and crossover matrix  $\mathbf{b}$  are sorted according to real crossover rate  $CR'$ . Each individual in the population is assigned a binary string in matrix  $\mathbf{b}$  based on its fitness value. In addition, we repair the crossover matrix  $\mathbf{b}$  with the opposition learning technique if a successful search is obtained in the last generation. Finally, for individual  $i$ , if  $ST_i^G > T$ , DX will be performed.

(4) Selection

As classic DE, the fitness value of the trial vector  $\mathbf{u}_i^G$  is compared with the fitness of its target vector  $\mathbf{x}_i^G$ , then the superior one will be chosen as  $\mathbf{x}_i^{G+1}$  to enter the next generation. The pseudo-code of ADEwSE is presented in Algorithm 6. In addition, an improved ADEwSE by introducing population size reduction scheme is presented in section V.

**Algorithm 6** Adaptive differential evolution with successful experience (ADEwSE)

01:  $G=1$ ; *Archive*  $A = \emptyset$ ;  $NFES=0$ ;  
 02: Generate initial population randomly;  
 03: Set  $NP=100$ ,  $\mu_{CR}=0.5$ ,  $\mu_F=0.5$ ,  $\mu_p=0.5$ ,  $\mu_A=0.0$ ,  $\mu_B=0.0$   $c_p=0.05$ ,  $c=0.1$ , initialize successful experience vectors based on Algorithm 3, and let  $ST$  be a zero vector with  $NP$  entries.  
 04: **While** the termination condition is not reached **do**  
 05:   **For**  $i = 1$  to  $NP$  **do**  
 06:      $CR_i^G = randn_i(\mu_{CR}, 0.1)$ ;  
 07:      $F_i^G = randc_i(\mu_F, 0.1)$ ;  
 08:      $A_i^G = randc_i(\mu_A, 0.1)$   
 09:      $B_i^G = randn_i(\mu_B, 0.1)$   
 10:      $p_i^G = randn_i(\mu_p, 0.1)$   
 11:   **End For**  
 12: Sort crossover string  $\mathbf{b}_i$  according to real crossover rate values, and sort  $B_i$  based on their values  
 13. Apply opposition learning of  $CR$  by **Algorithm 5**  
 14: **For**  $i = 1$  to  $NP$  **do**  
 15:   Generate a mutant vector  $\mathbf{v}_i^G$  according to Eq. 20  
 16:   **If**  $ST_i^G \leq T$   
 17:     Generate a trial vector  $\mathbf{u}_i^G$  according to Eq. 10;  
 18:   **Else**  
 19:     Generate a trial vector  $\mathbf{u}_i^G$  according to Eq. 18;  
 20:   **End If**  
 21: **End For**  
 22: Evaluate trial vectors  $\mathbf{u}$   
 23: Update successful experience vector as stated in **algorithm 2**.  
 24: Update population according to the evaluation of  $\mathbf{u}$ ;  
 25: Update Archive set  $A$ ; update  $\mu_{CR}$ ,  $\mu_F$ ,  $\mu_p$ ,  $\mu_A$ ,  $\mu_B$ ,  $NFES$ , and  $ST$ .  
 26:  $G = G + 1$ ;  
 27: **End While**

Considering that a lot of new control parameters and symbols are introduced in this paper, a simple description is shown in Table 1.

**TABLE 1. Nomenclature in ADEwSE**

$\mathbf{b}$	Binary string
$\widetilde{CR}$	The value of opposition learning rate of crossover rate
$K$	Inertial weight
$p$	Top % $p$
$A$	Scale factor of successful experience vector
$B$	Inertial weight controller
$\Gamma$	Direction controller
$T$	Threshold of stagnation
$\delta F$	Scale factor of disturbance vector
$\delta s$	Successful experience memory
$\delta s_i$	Successful experience vector of individual $i$
$ST$	Record the consecutive unsuccessful updates for each individual $\mathbf{x}_i$

**V. EXPERIMENTAL RESULTS AND COMPARISONS**

In this section, the computational results of the proposed algorithm are presented and discussed along with the comparison with other advanced DE and other “current-to-

**TABLE 2. Experimental results of ADEwSE, jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, and EDE on 30D**

		jDE	JADE	SHADE	CoBiDE	MPEDE	AGDE	EFADE	CIPDE	EDE	ADEwSE
$f_1$	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
$f_3$	Mean	4.89E-04	8.49E+03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.82E+02	0.00E+00	0.00E+00
	SD	1.91E-03	1.55E+04	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.75E+03	0.00E+00	0.00E+00
$f_4$	Mean	5.53E+01	4.71E+01	3.99E+01	3.78E+01	5.60E+01	3.42E+01	3.39E+01	4.07E+01	1.87E+01	4.01E+01
	SD	1.66E+01	2.49E+01	2.82E+01	2.84E+01	1.39E+01	3.04E+01	2.95E+01	2.85E+01	2.72E+01	2.84E+01
$f_5$	Mean	4.12E+01	2.71E+01	1.61E+01	4.00E+01	2.63E+01	7.31E+01	3.76E+01	2.05E+01	1.53E+01	9.78E+00
	SD	5.19E+00	3.47E+00	2.71E+00	1.22E+01	7.50E+00	7.27E+00	8.99E+00	7.23E+00	2.69E+00	1.90E+00
$f_6$	Mean	0.00E+00	0.00E+00	2.47E-07	1.46E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.96E-08
	SD	0.00E+00	0.00E+00	8.22E-07	3.27E-08	2.71E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.02E-07
$f_7$	Mean	7.87E+01	5.36E+01	4.60E+01	7.18E+01	5.50E+01	1.14E+02	8.02E+01	4.89E+01	4.60E+01	3.83E+01
	SD	5.64E+00	3.61E+00	2.28E+00	1.14E+01	7.53E+00	1.02E+01	2.32E+01	7.75E+00	2.63E+00	1.34E+00
$f_8$	Mean	4.36E+01	2.53E+01	1.71E+01	3.97E+01	2.97E+01	7.87E+01	4.22E+01	2.03E+01	1.71E+01	9.44E+00
	SD	5.70E+00	3.89E+00	2.60E+00	1.16E+01	7.11E+00	6.91E+00	1.11E+01	7.21E+00	2.39E+00	1.95E+00
$f_9$	Mean	0.00E+00	5.27E-03	0.00E+00	0.00E+00	1.07E-02	8.91E-03	8.91E-03	1.76E-03	0.00E+00	8.91E-03
	SD	0.00E+00	2.78E-02	0.00E+00	0.00E+00	6.46E-02	6.36E-02	6.36E-02	1.25E-02	0.00E+00	6.36E-02
$f_{10}$	Mean	2.72E+03	1.93E+03	1.71E+03	1.95E+03	2.77E+03	3.80E+03	3.43E+03	1.61E+03	1.70E+03	1.63E+03
	SD	2.84E+02	1.81E+02	1.90E+02	5.26E+02	4.16E+02	3.34E+02	8.88E+02	3.83E+02	2.19E+02	3.58E+02
$f_{11}$	Mean	2.27E+01	2.73E+01	2.02E+01	1.98E+01	2.25E+01	2.28E+01	1.48E+01	3.21E+01	1.96E+01	1.16E+01
	SD	2.24E+01	2.21E+01	2.04E+01	1.96E+01	8.47E+00	2.50E+01	1.22E+01	2.57E+01	2.14E+01	1.73E+01
$f_{12}$	Mean	9.68E+03	1.19E+03	1.28E+03	3.97E+03	1.00E+03	7.20E+03	8.88E+03	4.73E+03	1.57E+03	3.22E+02
	SD	6.59E+03	4.20E+02	4.22E+02	4.79E+03	4.18E+02	1.00E+04	6.99E+03	3.26E+03	1.25E+03	1.77E+02
$f_{13}$	Mean	4.23E+01	4.44E+02	3.91E+01	2.33E+01	2.06E+01	3.16E+01	2.85E+01	5.64E+01	5.50E+01	1.52E+01
	SD	8.47E+01	2.80E+03	3.87E+01	8.38E+00	9.76E+00	2.34E+01	1.92E+01	5.13E+01	3.06E+01	6.51E+00
$f_{14}$	Mean	2.67E+01	8.26E+03	2.71E+01	1.08E+01	1.34E+01	2.94E+01	1.13E+01	3.82E+02	2.62E+01	1.39E+01
	SD	9.44E+00	1.37E+04	7.20E+00	6.47E+00	9.72E+00	9.34E+00	4.27E+00	9.81E+02	2.93E+00	1.00E+01
$f_{15}$	Mean	1.10E+01	1.36E+03	1.32E+01	7.55E+00	8.63E+00	9.42E+00	8.34E+00	2.55E+02	1.04E+01	2.33E+00
	SD	4.96E+00	3.58E+03	7.81E+00	3.85E+00	3.30E+00	8.40E+00	2.92E+00	9.84E+02	5.84E+00	1.26E+00
$f_{16}$	Mean	4.26E+02	4.50E+02	3.01E+02	3.74E+02	3.29E+02	4.20E+02	4.15E+02	3.80E+02	3.04E+02	1.50E+02
	SD	1.65E+02	1.53E+02	1.29E+02	1.84E+02	1.84E+02	1.47E+02	1.77E+02	1.82E+02	1.07E+02	1.20E+02
$f_{17}$	Mean	8.74E+01	7.36E+01	4.47E+01	3.91E+01	4.99E+01	8.41E+01	5.05E+01	4.33E+01	4.48E+01	3.46E+01
	SD	1.89E+01	2.19E+01	9.30E+00	1.20E+01	1.40E+01	2.84E+01	2.68E+01	3.76E+01	1.03E+01	1.04E+01
$f_{18}$	Mean	7.04E+01	2.97E+04	7.81E+01	1.96E+01	2.20E+01	2.58E+02	9.32E+01	6.11E+03	5.23E+01	2.09E+01
	SD	1.43E+02	6.94E+04	4.87E+01	8.78E+00	8.13E+00	3.37E+02	1.12E+02	1.91E+04	3.99E+01	7.23E-01
$f_{19}$	Mean	1.11E+01	1.53E+03	9.68E+00	4.64E+00	7.30E+00	1.36E+01	6.15E+00	2.36E+02	1.24E+01	2.99E+00
	SD	3.31E+00	5.00E+03	4.02E+00	1.43E+00	2.06E+00	2.96E+00	2.21E+00	1.24E+03	3.04E+00	1.17E+00
$f_{20}$	Mean	7.85E+01	1.42E+02	6.76E+01	4.78E+01	6.72E+01	1.05E+02	7.55E+01	9.06E+01	6.28E+01	2.31E+01
	SD	3.43E+01	5.43E+01	4.57E+01	6.00E+01	5.14E+01	5.76E+01	6.50E+01	6.87E+01	3.98E+01	9.91E+00
$f_{21}$	Mean	2.43E+02	2.27E+02	2.18E+02	2.42E+02	2.28E+02	2.78E+02	2.43E+02	2.20E+02	2.18E+02	2.10E+02
	SD	6.15E+00	4.89E+00	2.97E+00	1.04E+01	8.02E+00	8.82E+00	1.04E+01	8.32E+00	3.50E+00	2.46E+00
$f_{22}$	Mean	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02
	SD	0.00E+00	2.57E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
$f_{23}$	Mean	3.87E+02	3.73E+02	3.64E+02	3.88E+02	3.77E+02	4.21E+02	3.89E+02	3.67E+02	3.62E+02	3.53E+02
	SD	6.77E+00	6.01E+00	5.32E+00	1.11E+01	9.45E+00	7.69E+00	1.00E+01	1.02E+01	4.48E+00	4.88E+00
$f_{24}$	Mean	4.58E+02	4.40E+02	4.33E+02	4.58E+02	4.44E+02	4.97E+02	4.68E+02	4.35E+02	4.32E+02	4.28E+02
	SD	7.25E+00	5.40E+00	4.04E+00	1.13E+01	8.72E+00	1.63E+01	1.40E+01	6.13E+00	2.84E+00	2.94E+00
$f_{25}$	Mean	3.87E+02	3.87E+02	3.87E+02	3.87E+02	3.87E+02	3.87E+02	3.87E+02	3.87E+02	3.87E+02	3.87E+02
	SD	1.41E-01	5.37E-01	1.51E-01	4.13E-02	7.32E-02	4.84E-01	4.42E-02	2.52E-01	3.00E-01	2.07E-02
$f_{26}$	Mean	1.28E+03	1.18E+03	1.08E+03	1.33E+03	1.18E+03	1.60E+03	1.12E+03	1.16E+03	1.07E+03	9.57E+02
	SD	1.59E+02	7.23E+01	6.39E+01	2.89E+02	1.56E+02	9.57E+01	4.99E+02	9.19E+01	6.05E+01	6.48E+01
$f_{27}$	Mean	5.00E+02	5.06E+02	5.04E+02	4.94E+02	5.01E+02	5.02E+02	4.99E+02	5.05E+02	5.03E+02	5.01E+02
	SD	4.94E+00	7.47E+00	6.90E+00	9.55E+00	6.84E+00	6.90E+00	7.45E+00	6.31E+00	5.62E+00	7.72E+00
$f_{28}$	Mean	3.32E+02	3.49E+02	3.25E+02	3.31E+02	3.23E+02	3.29E+02	3.29E+02	3.44E+02	3.12E+02	3.21E+02
	SD	4.99E+01	5.79E+01	4.58E+01	5.07E+01	4.50E+01	4.76E+01	4.73E+01	5.70E+01	3.36E+01	4.28E+01
$f_{29}$	Mean	5.01E+02	4.85E+02	4.54E+02	4.20E+02	4.46E+02	5.22E+02	4.35E+02	4.39E+02	4.63E+02	4.28E+02
	SD	3.19E+01	3.17E+01	1.90E+01	3.29E+01	3.10E+01	4.85E+01	5.15E+01	2.73E+01	1.60E+01	1.43E+01
$f_{30}$	Mean	2.18E+03	2.19E+03	2.10E+03	2.08E+03	2.03E+03	2.39E+03	2.29E+03	2.17E+03	2.09E+03	1.97E+03
	SD	1.54E+02	4.90E+02	1.36E+02	1.15E+02	1.05E+02	3.74E+02	3.40E+02	1.74E+02	1.26E+02	2.60E+01
	Nob	5	4	5	8	4	5	5	5	7	21

pbest/1” based adaptive differential evolution algorithms.

**A. ALGORITHM COMPLEXITY ANALYSIS**

In general differential evolution algorithms including ADEwSE, the computation complexity in mutation and crossover is  $O(NP \cdot D)$ . Like JADE, the complexity of finding all pbest solutions in ADEwSE is  $O(NP \cdot \log(NP))$ . Besides, there are  $O(NP)$  operations in both selection and parameter adaptation. Furthermore, supposing that the fitness values of all trial vectors are better than or equal to the fitness values of parents in each generation, the update of successful experience vector memory costs  $O(NP \cdot D)$  floating-point arithmetic operations. Additionally, opposition learning of crossover rate and sorting crossover matrix cost  $O(NP \cdot D + NP \cdot \log(NP) + NP)$  logical operations. The total complexity of ADEwSE is therefore  $O(NG \cdot NP \cdot (3D + 2 \cdot \log(NP) + 2))$ , where  $NG$  is a fixed number of generations. After omitting the lower order terms, the magnitude of com-

putation complexity of ADEwSE is  $O(NG \cdot NP \cdot D)$ , which is the same as classic differential evolution, and does not require excessive computational cost.

In addition, compared with classic DE algorithms, ADEwSE takes more RAM for storing the successful experience vectors. If  $NP$  and  $D$  are determined, the ADEwSE will consume extra  $NP \cdot D$  double-precision to store successful experience vectors.

**B. EXPERIMENTS SETUP, PARAMETER SETTINGS AND INVOLVED ALGORITHMS**

To test the performance and the viability of the proposed ADEwSE algorithm, the ADEwSE is applied to solve the optimization benchmark functions. These functions are presented for CEC2017 competition, and a detailed description of tested functions can be found in [60]. According to their character, these functions can be classified into four classes:

- (1) unimodal functions:  $f_{01}, f_{03}$

- (2) basic simple multimodal functions:  $f_{04} - f_{10}$
- (3) hybrid functions:  $f_{11} - f_{20}$
- (4) composition functions:  $f_{21} - f_{30}$

For persuasive comparisons, experiments of the compared algorithms were conducted on the test suit. In this study, the solution error measure  $f(x) - f(x^*)$ , in which  $x$  is the best result obtained by algorithms in one run and  $x^*$  is the global optimum

of each benchmark function, is adopted. As suggested in [61], error values and standard deviations smaller than  $1E-8$  are considered as zero. The dimensions ( $D$ ) of function are 30, 50 and 100, respectively. The maximal number of function evaluations (FE) is set to 10,000D in all compared algorithms and each algorithm performs 51 independent runs on the same computer.

**TABLE 3. Experimental results of ADEwSE, jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, and EDE on 50D**

		jDE	JADE	SHADE	CoBiDE	MPEDE	AGDE	EFADE	CIPDE	EDE	ADEwSE
$f_1$	Mean	1.29E-07	<b>0.00E+00</b>	<b>0.00E+00</b>	9.12E-04	<b>0.00E+00</b>	7.80E-06	1.06E+00	<b>0.00E+00</b>	1.17E+03	<b>0.00E+00</b>
	SD	2.14E-07	<b>0.00E+00</b>	<b>0.00E+00</b>	2.68E-03	<b>0.00E+00</b>	3.05E-05	2.30E+00	<b>0.00E+00</b>	1.80E+03	<b>0.00E+00</b>
$f_5$	Mean	2.31E+02	3.38E+04	<b>0.00E+00</b>	4.37E-05	1.02E-04	2.16E-07	4.02E-05	8.03E+03	<b>0.00E+00</b>	<b>0.00E+00</b>
	SD	3.51E+02	4.55E+04	<b>0.00E+00</b>	5.50E-05	4.53E-04	3.92E-07	1.11E-04	1.15E+04	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_4$	Mean	6.78E+01	<b>4.03E+01</b>	5.43E+01	4.67E+01	5.12E+01	4.43E+01	4.09E+01	4.81E+01	8.17E+01	5.03E+01
	SD	4.82E+01	<b>4.60E+01</b>	4.54E+01	3.34E+01	4.76E+01	4.09E+01	2.69E+01	4.81E+01	4.53E+01	4.42E+01
$f_5$	Mean	9.14E+01	5.51E+01	3.56E+01	8.31E+01	5.78E+01	2.19E+02	9.02E+01	4.39E+01	3.65E+01	<b>2.47E+01</b>
	SD	9.93E+00	7.00E+00	6.27E+00	2.00E+01	1.51E+01	1.21E+01	1.85E+01	1.05E+01	5.12E+00	<b>3.95E+00</b>
$f_6$	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	8.28E-05	2.07E-03	1.07E-03	<b>0.00E+00</b>	<b>0.00E+00</b>	5.09E-08	5.39E-05	2.62E-06
	SD	<b>0.00E+00</b>	<b>0.00E+00</b>	1.81E-04	6.92E-04	4.02E-03	<b>0.00E+00</b>	<b>0.00E+00</b>	2.28E-07	7.23E-05	3.48E-06
$f_7$	Mean	1.49E+02	9.91E+01	8.32E+01	1.56E+02	1.07E+02	2.86E+02	1.60E+02	8.78E+01	7.91E+01	<b>6.89E+01</b>
	SD	1.21E+01	7.17E+00	4.40E+00	4.69E+01	1.24E+01	1.36E+01	5.68E+01	1.26E+01	3.96E+00	<b>3.26E+00</b>
$f_8$	Mean	8.81E+01	5.31E+01	3.64E+01	7.73E+01	5.23E+01	2.21E+02	9.38E+01	4.64E+01	3.64E+01	<b>2.51E+01</b>
	SD	1.02E+01	7.08E+00	5.74E+00	1.48E+01	1.05E+01	1.24E+01	2.16E+01	1.29E+01	4.67E+00	<b>4.27E+00</b>
$f_9$	Mean	6.22E-02	9.98E-01	6.71E-01	<b>0.00E+00</b>	1.16E+00	1.76E-03	2.46E-01	1.52E+00	3.97E-01	9.25E-02
	SD	1.52E-01	7.76E-01	7.96E-01	<b>0.00E+00</b>	1.47E+00	1.25E-02	5.74E-01	1.15E+00	4.38E-01	1.94E-01
$f_{10}$	Mean	4.99E+03	3.75E+03	3.32E+03	3.93E+03	4.75E+03	8.52E+03	7.88E+03	3.39E+03	3.34E+03	<b>2.78E+03</b>
	SD	3.83E+02	2.95E+02	3.04E+02	6.84E+02	6.30E+02	3.76E+02	1.51E+03	5.36E+02	3.18E+02	<b>6.82E+02</b>
$f_{11}$	Mean	5.21E+01	1.36E+02	1.02E+02	4.59E+01	1.03E+02	4.80E+01	4.65E+01	1.15E+02	1.02E+02	<b>4.49E+01</b>
	SD	1.27E+01	4.18E+01	2.37E+01	8.65E+00	2.38E+01	1.10E+01	7.30E+00	4.32E+01	2.48E+01	<b>1.32E+01</b>
$f_{12}$	Mean	4.69E+04	4.61E+03	4.91E+03	6.69E+04	1.01E+04	6.74E+04	3.95E+04	4.45E+03	8.17E+03	<b>2.89E+03</b>
	SD	2.88E+04	3.15E+03	2.79E+03	4.74E+04	7.02E+03	3.62E+04	2.66E+04	1.98E+03	6.84E+03	<b>2.35E+03</b>
$f_{13}$	Mean	1.95E+03	2.59E+02	2.65E+02	7.33E+01	8.92E+01	2.71E+03	2.32E+03	3.87E+02	4.71E+02	<b>6.13E+01</b>
	SD	3.07E+03	1.54E+02	1.41E+02	3.81E+01	3.34E+01	4.31E+03	3.11E+03	3.69E+02	3.93E+02	<b>2.89E+01</b>
$f_{14}$	Mean	6.22E+01	2.52E+04	2.11E+02	3.82E+01	6.00E+01	2.64E+02	4.34E+01	1.46E+03	1.69E+02	<b>2.91E+01</b>
	SD	1.88E+01	6.69E+04	7.07E+01	1.08E+01	1.35E+01	7.04E+02	1.36E+01	6.21E+03	4.69E+01	<b>7.03E+00</b>
$f_{15}$	Mean	8.58E+01	3.70E+02	2.36E+02	3.59E+01	7.59E+01	5.08E+02	1.06E+02	2.34E+02	2.20E+02	<b>3.31E+01</b>
	SD	1.38E+02	4.54E+02	9.64E+01	7.80E+00	3.97E+01	2.17E+03	9.85E+01	1.18E+02	1.23E+02	<b>5.99E+00</b>
$f_{16}$	Mean	9.57E+02	8.89E+02	7.35E+02	8.32E+02	9.25E+02	9.72E+02	9.36E+02	6.61E+02	6.85E+02	<b>4.44E+02</b>
	SD	2.10E+02	1.98E+02	1.57E+02	2.33E+02	2.64E+02	2.13E+02	2.67E+02	2.08E+02	1.62E+02	<b>1.72E+02</b>
$f_{17}$	Mean	7.07E+02	6.10E+02	5.12E+02	4.63E+02	5.60E+02	7.37E+02	5.69E+02	5.65E+02	4.84E+02	<b>3.27E+02</b>
	SD	1.11E+02	1.58E+02	1.25E+02	1.85E+02	1.78E+02	1.41E+02	1.69E+02	1.84E+02	1.24E+02	<b>1.12E+02</b>
$f_{18}$	Mean	1.43E+03	4.48E+03	1.57E+02	3.38E+01	1.27E+02	2.78E+03	2.45E+03	2.17E+02	1.49E+02	<b>3.30E+01</b>
	SD	8.63E+02	3.06E+04	7.44E+01	5.78E+00	1.02E+02	2.64E+03	2.75E+03	2.02E+02	1.08E+02	<b>6.62E+00</b>
$f_{19}$	Mean	2.84E+01	4.51E+02	1.29E+02	1.67E+01	4.11E+01	5.23E+01	2.15E+01	1.23E+02	1.00E+02	<b>1.58E+01</b>
	SD	1.06E+01	2.23E+03	4.65E+01	2.87E+00	2.11E+01	1.15E+02	1.27E+01	3.56E+01	3.81E+01	<b>3.67E+00</b>
$f_{20}$	Mean	4.75E+02	5.50E+02	3.17E+02	3.30E+02	3.48E+02	4.98E+02	3.89E+02	4.55E+02	2.92E+02	<b>1.37E+02</b>
	SD	1.22E+02	1.18E+02	1.13E+02	1.46E+02	1.67E+02	1.51E+02	1.33E+02	2.02E+02	1.12E+02	<b>1.10E+02</b>
$f_{21}$	Mean	2.91E+02	2.54E+02	2.37E+02	2.82E+02	2.56E+02	4.26E+02	2.94E+02	2.43E+02	2.35E+02	<b>2.24E+02</b>
	SD	1.16E+01	8.06E+00	4.98E+00	1.76E+01	1.28E+01	1.25E+01	1.96E+01	1.27E+01	4.84E+00	<b>2.72E+00</b>
$f_{22}$	Mean	3.18E+03	<b>2.65E+03</b>	2.74E+03	3.42E+03	2.78E+03	7.17E+03	6.25E+03	3.11E+03	3.05E+03	2.79E+03
	SD	2.84E+03	<b>2.05E+03</b>	1.82E+03	1.99E+03	2.63E+03	3.44E+03	3.80E+03	1.85E+03	1.74E+03	1.38E+03
$f_{23}$	Mean	5.10E+02	4.79E+02	4.57E+02	5.05E+02	4.81E+02	6.47E+02	5.14E+02	4.72E+02	4.54E+02	<b>4.44E+02</b>
	SD	1.24E+01	1.16E+01	7.70E+00	1.67E+01	1.42E+01	1.35E+01	1.73E+01	1.61E+01	7.92E+00	<b>8.00E+00</b>
$f_{24}$	Mean	5.73E+02	5.41E+02	5.27E+02	5.71E+02	5.40E+02	6.99E+02	5.81E+02	5.37E+02	5.22E+02	<b>5.18E+02</b>
	SD	1.39E+01	8.99E+00	7.32E+00	1.80E+01	1.51E+01	4.56E+01	2.09E+01	1.19E+01	7.31E+00	<b>5.48E+00</b>
$f_{25}$	Mean	5.13E+02	5.23E+02	5.28E+02	4.96E+02	5.17E+02	<b>4.94E+02</b>	5.36E+02	5.50E+02	5.28E+02	<b>5.28E+02</b>
	SD	3.59E+01	3.34E+01	3.15E+01	2.60E+01	2.85E+01	3.70E+01	<b>2.54E+01</b>	3.78E+01	2.75E+01	3.29E+01
$f_{26}$	Mean	1.93E+03	1.63E+03	1.44E+03	1.96E+03	1.60E+03	3.11E+03	2.11E+03	1.53E+03	1.38E+03	<b>1.27E+03</b>
	SD	1.31E+02	9.74E+01	1.01E+02	2.22E+02	1.44E+02	1.65E+02	2.07E+02	1.59E+02	8.75E+01	<b>1.05E+02</b>
$f_{27}$	Mean	5.32E+02	5.57E+02	5.39E+02	<b>5.12E+02</b>	5.49E+02	5.15E+02	5.30E+02	5.80E+02	5.51E+02	5.18E+02
	SD	1.72E+01	3.03E+01	1.78E+01	<b>1.36E+01</b>	2.39E+01	1.49E+01	1.11E+01	3.34E+01	2.22E+01	1.40E+01
$f_{28}$	Mean	4.82E+02	4.92E+02	4.97E+02	4.64E+02	4.90E+02	4.79E+02	<b>4.61E+02</b>	4.96E+02	4.99E+02	4.90E+02
	SD	2.36E+01	2.12E+01	1.81E+01	1.43E+01	2.28E+01	2.29E+01	<b>9.58E+00</b>	1.80E+01	1.54E+01	2.19E+01
$f_{29}$	Mean	4.99E+02	4.75E+02	4.73E+02	4.69E+02	4.35E+02	6.37E+02	5.04E+02	4.24E+02	4.47E+02	<b>3.62E+02</b>
	SD	7.62E+01	8.28E+01	7.77E+01	1.13E+02	1.36E+02	1.06E+02	1.39E+02	1.17E+02	7.38E+01	<b>3.98E+01</b>
$f_{30}$	Mean	5.98E+05	6.76E+05	6.25E+05	<b>5.90E+05</b>	6.52E+05	6.32E+05	6.11E+05	6.43E+05	6.14E+05	6.05E+05
	SD	2.39E+04	8.69E+04	4.95E+04	<b>2.06E+04</b>	7.47E+04	7.73E+04	3.76E+04	6.42E+04	4.25E+04	2.82E+04
	Nob	1	3	2	3	1	1	3	1	1	<b>21</b>

To perform a comprehensive evaluation, ADEwSE is compared to fourteen DE-based algorithms, namely: the differential evolution with self-adapting control parameters (jDE) [19], adaptive differential evolution with the optional external archive (JADE) [20], success-history based parameter adaptation for differential evolution (SHADE) [21], differential evolution based on covariance matrix learning and bimodal distribution parameter setting (CoBiDE) [22], differential evolution with multi-population based ensemble

of mutation strategies (MPEDE) [18], adaptive guided differential evolution algorithm with novel mutation for numerical optimization (AGDE) [33], real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation (EFADE) [62], novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization (EDE) [29], differential evolution with ranking based mutation operators (JADErank) [63], Repairing the crossover rate in

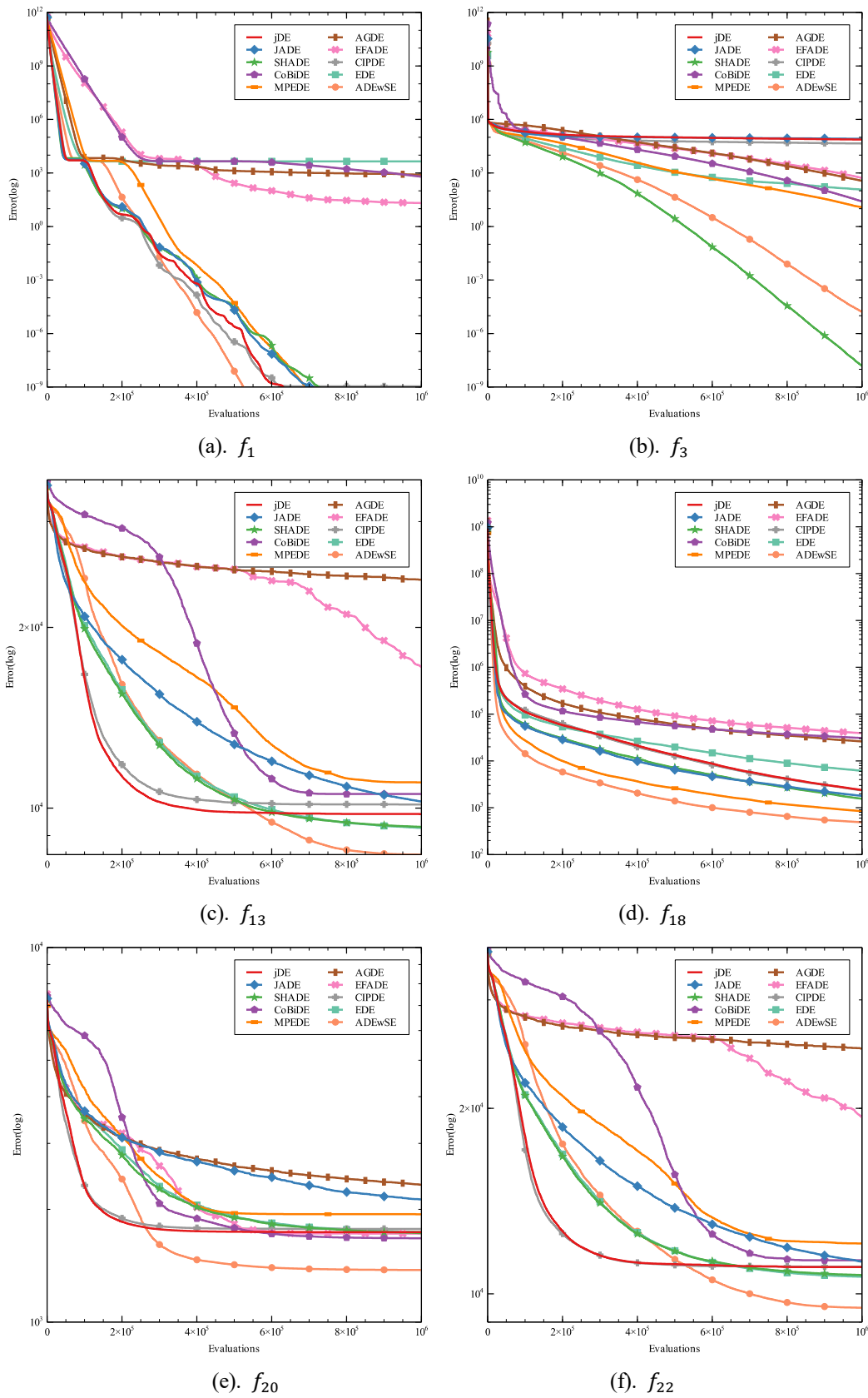


FIGURE 1. Convergence graph (mean values) of ten algorithms on  $f_1, f_3, f_{13}, f_{18}, f_{20}, f_{22}$  with 100D

**TABLE 4. Experimental results of ADEwSE, jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, and EDE on 100D**

		jDE	JADE	SHADE	CoBiDE	MPEDE	AGDE	EFADE	CIPDE	EDE	ADEwSE
$f_1$	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	6.01E+02	<b>0.00E+00</b>	8.08E+02	2.05E+01	<b>0.00E+00</b>	4.49E+03	<b>0.00E+00</b>
	SD	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	8.23E+02	<b>0.00E+00</b>	1.85E+03	3.37E+01	<b>0.00E+00</b>	3.28E+03	<b>0.00E+00</b>
$f_3$	Mean	5.14E+04	8.25E+04	<b>1.56E-08</b>	2.37E+01	1.20E+01	3.61E+02	5.26E+02	7.46E+04	1.16E+02	1.62E-05
	SD	4.26E+04	1.51E+05	<b>2.65E-08</b>	1.90E+01	3.70E+01	4.60E+02	3.87E+02	7.52E+04	8.29E+02	5.21E-05
$f_4$	Mean	2.06E+02	7.36E+01	7.18E+01	2.14E+02	6.41E+01	1.96E+02	2.05E+02	1.03E+02	1.81E+02	<b>1.68E+01</b>
	SD	1.71E+01	6.60E+01	6.50E+01	2.29E+01	6.50E+01	2.59E+01	1.37E+01	4.92E+01	4.13E+01	<b>4.69E+01</b>
$f_5$	Mean	2.41E+02	1.48E+02	1.19E+02	1.80E+02	1.54E+02	7.24E+02	1.84E+02	1.30E+02	1.08E+02	<b>1.04E+02</b>
	SD	2.21E+01	1.81E+01	1.39E+01	3.40E+01	3.06E+01	2.55E+01	3.28E+01	1.97E+01	1.35E+01	<b>1.43E+01</b>
$f_6$	Mean	1.11E-08	9.91E-04	9.61E-02	6.74E-05	1.39E-01	<b>0.00E+00</b>	<b>0.00E+00</b>	4.31E-03	8.53E-02	2.46E-02
	SD	5.76E-08	2.73E-03	6.80E-02	3.96E-05	9.95E-02	<b>0.00E+00</b>	<b>0.00E+00</b>	1.20E-02	4.19E-02	2.16E-02
$f_7$	Mean	3.56E+02	2.69E+02	2.22E+02	2.75E+02	3.01E+02	8.48E+02	2.72E+02	2.38E+02	<b>2.02E+02</b>	2.15E+02
	SD	3.09E+01	2.41E+01	1.71E+01	3.20E+01	3.50E+01	3.00E+01	3.30E+01	2.33E+01	<b>1.43E+01</b>	1.83E+01
$f_8$	Mean	2.39E+02	1.45E+02	1.13E+02	1.95E+02	1.50E+02	7.16E+02	1.79E+02	1.30E+02	1.05E+02	<b>1.03E+02</b>
	SD	2.84E+01	1.84E+01	1.13E+01	3.42E+01	2.61E+01	3.46E+01	3.71E+01	2.21E+01	1.04E+01	<b>1.59E+01</b>
$f_9$	Mean	<b>4.21E+00</b>	8.90E+01	4.74E+01	1.65E+00	3.87E+01	1.13E+00	2.46E+01	7.79E+01	4.30E+01	1.43E+02
	SD	<b>3.87E+00</b>	1.03E+02	3.29E+01	3.56E+00	3.23E+01	1.25E+00	2.65E+01	4.57E+01	2.04E+01	1.47E+02
$f_{10}$	Mean	1.31E+04	1.03E+04	9.31E+03	1.06E+04	1.10E+04	2.40E+04	1.72E+04	9.77E+03	9.27E+03	<b>8.37E+03</b>
	SD	6.59E+02	4.84E+02	5.08E+02	1.32E+03	1.12E+03	5.59E+02	6.45E+03	1.14E+03	4.63E+02	<b>1.09E+03</b>
$f_{11}$	Mean	3.46E+02	4.77E+03	1.09E+03	2.14E+02	8.71E+02	1.63E+02	<b>1.41E+02</b>	6.77E+02	5.24E+02	5.22E+02
	SD	7.89E+01	3.76E+03	2.54E+02	4.89E+01	2.78E+02	5.76E+01	<b>4.99E+01</b>	2.93E+02	1.63E+02	9.36E+01
$f_{12}$	Mean	1.94E+05	1.86E+04	<b>1.67E+04</b>	3.58E+05	2.92E+04	2.86E+05	2.37E+05	1.92E+04	1.50E+05	1.82E+04
	SD	8.55E+04	9.71E+03	<b>7.72E+03</b>	1.25E+05	1.26E+04	1.14E+05	9.80E+04	9.40E+03	7.26E+04	8.54E+03
$f_{13}$	Mean	2.92E+03	2.79E+03	1.36E+03	3.15E+03	7.75E+02	3.45E+03	2.20E+03	3.20E+03	2.41E+03	<b>3.75E+02</b>
	SD	2.71E+03	2.84E+03	2.68E+03	2.68E+03	9.43E+02	3.00E+03	1.62E+03	3.37E+03	3.37E+03	<b>5.43E+02</b>
$f_{14}$	Mean	1.23E+04	5.89E+02	5.72E+02	1.37E+03	4.62E+02	8.10E+03	9.87E+03	3.92E+02	5.33E+02	<b>2.18E+02</b>
	SD	1.00E+04	2.10E+02	2.87E+02	1.87E+03	1.09E+02	8.70E+03	6.14E+03	1.44E+02	3.46E+02	<b>4.73E+01</b>
$f_{15}$	Mean	1.13E+03	3.74E+02	3.35E+02	2.16E+03	3.37E+02	2.01E+03	1.47E+03	3.83E+02	9.85E+02	<b>2.75E+02</b>
	SD	1.43E+03	8.57E+01	9.31E+01	2.87E+03	9.18E+01	2.06E+03	2.13E+03	3.99E+02	1.11E+03	<b>8.03E+01</b>
$f_{16}$	Mean	2.90E+03	2.64E+03	2.33E+03	2.74E+03	2.69E+03	3.47E+03	2.88E+03	2.50E+03	2.54E+03	<b>2.39E+03</b>
	SD	2.88E+02	3.06E+02	2.48E+02	4.66E+02	5.98E+02	3.65E+02	5.54E+02	5.35E+02	2.76E+02	<b>3.83E+02</b>
$f_{17}$	Mean	2.19E+03	1.89E+03	1.88E+03	1.72E+03	1.98E+03	2.41E+03	1.79E+03	1.79E+03	1.75E+03	<b>3.34E+03</b>
	SD	2.40E+02	2.41E+02	2.61E+02	3.44E+02	4.60E+02	2.44E+02	3.22E+02	4.15E+02	4.15E+02	<b>1.29E+02</b>
$f_{18}$	Mean	4.88E+04	1.80E+03	1.56E+03	3.07E+04	8.43E+02	2.61E+04	3.92E+04	2.39E+03	6.17E+03	<b>4.90E+02</b>
	SD	3.15E+04	1.35E+03	1.03E+03	2.01E+04	8.05E+02	1.13E+04	1.88E+04	2.51E+03	4.22E+03	<b>3.41E+02</b>
$f_{19}$	Mean	1.36E+03	1.46E+03	4.87E+02	2.31E+03	2.19E+02	2.28E+03	1.01E+03	1.80E+03	1.34E+03	<b>1.48E+02</b>
	SD	1.82E+03	2.25E+03	7.85E+02	3.74E+03	4.63E+01	2.56E+03	1.30E+03	2.34E+03	1.39E+03	<b>3.91E+01</b>
$f_{20}$	Mean	2.10E+03	2.13E+03	1.73E+03	1.67E+03	1.94E+03	2.33E+03	1.72E+03	1.74E+03	1.74E+03	<b>1.38E+03</b>
	SD	2.25E+02	2.29E+02	2.38E+02	4.01E+02	4.37E+02	2.70E+02	3.67E+02	4.28E+02	1.71E+02	<b>3.16E+02</b>
$f_{21}$	Mean	4.67E+02	3.69E+02	3.39E+02	4.17E+02	3.59E+02	9.49E+02	4.16E+02	3.47E+02	3.23E+02	<b>3.13E+02</b>
	SD	2.32E+01	1.74E+01	1.65E+01	3.40E+01	2.34E+01	2.49E+01	3.56E+01	1.77E+01	1.16E+01	<b>1.35E+01</b>
$f_{22}$	Mean	1.40E+04	1.13E+04	1.07E+04	1.13E+04	1.21E+04	2.50E+04	1.93E+04	1.11E+04	1.07E+04	<b>9.50E+03</b>
	SD	6.78E+02	6.36E+02	5.14E+02	1.19E+03	1.18E+03	5.59E+02	6.24E+03	1.19E+03	6.45E+02	<b>1.07E+03</b>
$f_{23}$	Mean	6.56E+02	6.52E+02	6.15E+02	6.78E+02	6.97E+02	1.03E+03	7.04E+02	6.42E+02	<b>6.00E+02</b>	6.33E+02
	SD	1.34E+01	1.20E+01	1.43E+01	3.14E+01	2.83E+01	1.72E+01	2.70E+01	2.37E+01	<b>1.26E+01</b>	3.25E+01
$f_{24}$	Mean	1.08E+03	1.02E+03	9.91E+02	1.07E+03	1.03E+03	1.53E+03	1.06E+03	1.01E+03	<b>9.54E+02</b>	9.82E+02
	SD	2.12E+01	1.99E+01	2.17E+01	4.19E+01	3.28E+01	3.31E+01	2.96E+01	2.44E+01	<b>1.62E+01</b>	2.18E+01
$f_{25}$	Mean	7.61E+02	7.69E+02	7.40E+02	7.33E+02	7.53E+02	7.31E+02	<b>7.25E+02</b>	8.13E+02	8.14E+02	7.53E+02
	SD	5.26E+01	4.70E+01	5.00E+01	4.51E+01	4.99E+01	5.59E+01	<b>4.04E+01</b>	5.90E+01	6.15E+01	5.14E+01
$f_{26}$	Mean	5.28E+03	4.59E+03	4.37E+03	4.88E+03	4.57E+03	9.69E+03	4.86E+03	4.50E+03	<b>4.14E+03</b>	<b>4.14E+03</b>
	SD	3.09E+02	2.72E+02	2.78E+02	4.28E+02	3.36E+02	8.29E+02	3.64E+02	3.06E+02	2.83E+02	<b>2.83E+02</b>
$f_{27}$	Mean	6.70E+02	7.29E+02	6.97E+02	6.27E+02	6.94E+02	6.25E+02	6.54E+02	7.82E+02	7.33E+02	<b>6.18E+02</b>
	SD	3.12E+01	3.57E+01	2.64E+01	2.51E+01	2.93E+01	1.83E+01	1.80E+01	3.80E+01	3.00E+01	<b>2.75E+01</b>
$f_{28}$	Mean	5.60E+02	5.37E+02	5.26E+02	5.55E+02	5.25E+02	5.58E+02	5.45E+02	5.49E+02	5.50E+02	<b>4.86E+02</b>
	SD	2.69E+01	4.52E+01	4.82E+01	3.32E+01	3.40E+01	3.00E+01	1.56E+01	3.75E+01	3.09E+01	<b>7.92E+01</b>
$f_{29}$	Mean	2.14E+03	2.20E+03	2.11E+03	2.05E+03	2.44E+03	2.73E+03	2.19E+03	1.92E+03	2.00E+03	<b>1.59E+03</b>
	SD	2.43E+02	3.24E+02	3.07E+02	3.41E+02	4.03E+02	1.87E+02	3.37E+02	3.73E+02	2.63E+02	<b>3.13E+02</b>
$f_{30}$	Mean	3.75E+03	3.53E+03	<b>2.60E+03</b>	4.09E+03	<b>2.60E+03</b>	4.76E+03	3.85E+03	3.83E+03	5.07E+03	2.63E+03
	SD	1.89E+03	1.61E+03	2.28E+02	1.53E+03	<b>2.24E+02</b>	2.48E+03	1.39E+03	1.70E+03	1.84E+03	3.93E+02
	Nob	2	1	4	0	2	1	3	1	4	<b>19</b>

adaptive differential evolution (JADERcr) [21], adaptive differential evolution with sorting crossover rate for continuous optimization problems (JADEsort) [28], an improved adaptive differential evolution algorithm for continuous optimization (pbestr-JADE) [64], distance-based parameter adaptation for Success-History based differential evolution (DbSHADE) [65].

To compare and analyze the solution quality effectively from a statistical angle of various algorithms, the results are compared by adopting two nonparametric statistical hypothesis tests, namely, multi-problem Wilcoxon signed-rank test and Friedman test. The significance level is set to 0.05. For multi-problem Wilcoxon signed rank test,  $R^+$  and  $R^-$  denote the sum of ranks for test problems in which the first algorithm performs better than or worse than its competitor, respectively. A larger rank indicates a larger performance

discrepancy. Regarding Friedman test, the algorithms with small rankings have better performance according to the final rankings of different algorithms for all functions.

### C. COMPARISON AGAINST ADVANCED DIFFERENTIAL EVOLUTION VARIANTS

In this part, the comparison is performed between ADEwSE and nine advanced DE variants (i.e., jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, and EDE) on 29 CEC2017 test functions. All compared algorithms are under the recommended parameter settings as described in their original papers. The statistical results of the comparisons on test functions with 30, 50 and 100 dimensions are summarized in Tables 2, 3, and 4, respectively. It includes the obtained mean (denoted by Mean) and the standard deviations (denoted by SD) of error from optimum solution of ADEwSE and other

nine advanced DE variant algorithms over 51 runs for all 29 benchmark functions. The best results are marked in bold for all problems. In addition, the number of the best results (Nob) are summarized at the bottom of the tables. The multi-problem Wilcoxon signed-rank and Friedman tests between ADEwSE and other competitors are shown in Tables 5, 6 and 7, respectively.

According to the statistical results shown in Tables 2, 3 and 4, it can be observed that for 30D functions, ADEwSE achieves the best results in 21 cases, while the number of the best results obtained by jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, and EDE are 5, 4, 5, 8, 4, 5, 5, 5, and 7, respectively. Concerning 50D functions, ADEwSE achieves the best results in 22 cases, while jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, and EDE achieve that in 1, 3, 2, 3, 1, 1, 3, 1, and 1 cases, respectively. Concerning 100D functions, ADEwSE achieves the best results in 19 cases, while jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, and EDE are 2, 1, 4, 0, 2, 1, 3, 1, and 4, respectively.

**TABLE 5. Results of the multi-problem Wilcoxon's test for ADEwSE versus the advanced DE variants at 30D, 50D, and 100D**

Algorithm at 30D	R <sup>+</sup>	R <sup>-</sup>	p-value	$\alpha=0.05$	$\alpha=0.1$
jDE	390.5	15.5	1.9E-5	Yes	Yes
JADE	397.5	8.5	8.0E-6	Yes	Yes
SHADE	417.0	18.0	1.5E-5	Yes	Yes
CoBiDE	366.0	69.0	1.3E-3	Yes	Yes
MPEDE	389.0	17.0	2.1E-5	Yes	Yes
AGDE	416.0	19.0	1.6E-5	Yes	Yes
EFADE	372.0	34.0	1.1E-4	Yes	Yes
CIPDE	382.0	23.5	4.2E-5	Yes	Yes
EDE	383.0	51.5	3.1E-4	Yes	Yes
Algorithm at 50D	R <sup>+</sup>	R <sup>-</sup>	p-value	$\alpha=0.05$	$\alpha=0.1$
jDE	389.0	46.0	2.0E-4	Yes	Yes
JADE	381.0	25.0	4.8E-6	Yes	Yes
SHADE	428.5	6.5	5.0E-6	Yes	Yes
CoBiDE	357.0	78.0	2.5E-3	Yes	Yes
MPEDE	418.5	16.5	1.3E-6	Yes	Yes
AGDE	400.0	35.0	7.3E-6	Yes	Yes
EFADE	406.0	29.0	4.2E-6	Yes	Yes
CIPDE	402.0	4.0	5.0E-6	Yes	Yes
EDE	405.0	1.0	4.0E-6	Yes	Yes
Algorithm at 100D	R <sup>+</sup>	R <sup>-</sup>	p-value	$\alpha=0.05$	$\alpha=0.1$
jDE	389.0	46.0	2.0E-4	Yes	Yes
JADE	381.0	25.0	4.8E-6	Yes	Yes
SHADE	428.5	6.5	5.0E-6	Yes	Yes
CoBiDE	357.0	78.0	2.5E-3	Yes	Yes
MPEDE	418.5	16.5	1.3E-6	Yes	Yes
AGDE	400.0	35.0	7.3E-6	Yes	Yes
EFADE	406.0	29.0	4.2E-6	Yes	Yes
CIPDE	402.0	4.0	5.0E-6	Yes	Yes
EDE	405.0	1.0	4.0E-6	Yes	Yes

In addition, for the 30D unimodal functions ( $f_{01}, f_{03}$ ) and basic simple multimodal functions ( $f_{04}-f_{10}$ ), Table 2 shows that jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, EDE and ADEwSE achieve the best results in 3, 2, 3, 3, 2, 3, 3, 3, 4 and 5 cases, respectively. for the 30D hybrid and composition functions ( $f_{11}-f_{30}$ ), jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, EDE and ADEwSE achieve the best results in 2, 2, 2, 5, 2, 2, 2, 2, 3, and 16 cases, respectively. As for 50D, from Table 2, jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, EDE and ADEwSE can obtain the best results in 1, 3, 2, 1, 1, 1, 1, 1, 1, and 6 cases, respectively, on the 50D unimodal ( $f_{01}, f_{03}$

and basic simple multimodal functions ( $f_{04}-f_{10}$ ). For the 50D hybrid and composition functions ( $f_{11}-f_{30}$ ), jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, EDE and ADEwSE achieve the best results on 0, 1, 0, 2, 0, 0, 2, 0, 0, and 16 functions, respectively. At D = 100, ADEwSE and nine advanced DE variants can obtain the best results on 7, 1, 3, 2, 1, 1, 1, 1, 1, and 1 functions, respectively, on the 100D unimodal ( $f_{01}, f_{03}$ ) and basic simple multimodal functions ( $f_{04}-f_{10}$ ). However, for the 100D hybrid and composition functions ( $f_{11}-f_{30}$ ), jDE, JADE, SHADE, CoBiDE, MPEDE, AGDE, EFADE, CIPDE, EDE and ADEwSE achieve the best results on 0, 0, 2, 0, 1, 0, 2, 0, 3, and 14 functions, respectively. Obviously, the ADEwSE shows more clear superiority on hybrid and composition functions. In addition, the number of the best results obtained by all compared algorithms as summarized graphically in Figure 2 further proves the superior performance of ADEwSE on tested problems with 30D, 50D and 100D.

**TABLE 6. Average ranking values (ARV) of ADEwSE and other advanced DE variants according to Friedman test**

Rank	D=30		D=50		D=100	
	Algorithm	ARV	Algorithm	ARV	Algorithm	ARV
1	ADEwSE	2.62	ADEwSE	2.18	ADEwSE	2.15
2	EDE	4.09	CoBiDE	4.76	SHADE	3.71
3	CoBiDE	4.60	SHADE	4.84	EDE	4.78
4	SHADE	4.71	EDE	5.09	MPEDE	5.22
5	MPEDE	5.03	MPEDE	5.56	CIPDE	5.33
6	EFADE	5.76	CIPDE	5.72	JADE	5.93
7	CIPDE	6.14	EFADE	6.33	EFADE	6.17
8	jDE	7.05	jDE	6.43	CoBiDE	6.39
9	JADE	7.33	JADE	6.50	JADE	7.95
10	AGDE	7.67	AGDE	7.58	AGDE	7.95

**TABLE 7. Average ranks for all algorithms across all functions**

Ranks	Algorithm	30D	50D	100D	Mean ranking
1	ADEwSE	2.62	2.18	2.15	2.32
2	SHADE	4.71	4.84	3.71	4.42
3	EDE	4.09	5.09	4.78	4.65
4	CoBiDE	4.6	4.76	6.39	5.25
5	MPEDE	5.03	5.56	5.22	5.27
6	CIPDE	6.14	5.72	5.33	5.73
7	EFADE	5.76	6.33	6.17	6.08
8	jDE	7.05	6.43	7.36	6.95
9	JADE	7.33	6.5	7.95	7.26
10	AGDE	7.67	7.58	7.95	7.73

Furthermore, the results of multi-problem Wilcoxon's test presented in Table 5 clearly show that ADEwSE obtains higher R<sup>+</sup> values than R<sup>-</sup> in all cases, and all the p-values are less than 0.05. It demonstrates that ADEwSE outperforms other compared advanced DE variant algorithms significantly. Besides, Table 6 shows that ADEwSE obtains the first ranking among all compared algorithms for all 29 problems with 30D, 50D and 100D, based on the average ranking calculated by Friedman test. To analyze the performance of all algorithms on all functions at different dimensions, the mean aggregated rank of all the ten algorithms across all 29 problems and all dimensions included 30D, 50D and 100D is presented in Table 7. The statistical results show that ADEwSE is the best followed by SHADE as second best among all compared algorithms.

Moreover, to demonstrate the superiority of proposed algorithm intuitively, six representative test functions of CEC

TABLE 8. Experimental results of ADEwSE, JADE, SHADE, JADERank, JADEcr, JADEEP, JADEpbesttr, JADEsort, and DbSHADE on 30D

		JADE	SHADE	JADERank	JADEcr	JADEEP	JADEpbesttr	JADEsort	DbSHADE	ADEwSE
$f_1$	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
	SD	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_3$	Mean	8.49E+03	<b>0.00E+00</b>	6.90E+03	<b>0.00E+00</b>	3.31E+03	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
	SD	1.55E+04	<b>0.00E+00</b>	1.47E+04	<b>0.00E+00</b>	9.63E+03	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_4$	Mean	4.71E+01	3.99E+01	5.27E+01	4.99E+01	9.22E+00	4.37E+01	<b>3.36E+01</b>	3.90E+01	4.01E+01
	SD	2.49E+01	2.82E+01	1.93E+01	2.34E+01	1.90E+01	2.62E+01	<b>2.88E+01</b>	2.90E+01	2.84E+01
$f_5$	Mean	2.71E+01	1.61E+01	2.71E+01	2.10E+01	2.68E+01	2.03E+01	1.05E+01	1.58E+01	<b>9.78E+00</b>
	SD	3.47E+00	2.71E+00	4.01E+00	4.60E+00	4.20E+00	4.30E+00	2.10E+00	2.95E+00	<b>1.90E+00</b>
$f_6$	Mean	<b>0.00E+00</b>	2.47E-07	<b>0.00E+00</b>	8.14E-06	5.30E-06	2.84E-06	1.29E-04	1.40E-07	4.53E-08
	SD	<b>0.00E+00</b>	8.22E-07	<b>0.00E+00</b>	1.18E-05	4.71E-06	8.75E-06	7.60E-04	6.67E-07	2.81E-07
$f_7$	Mean	5.36E+01	4.60E+01	5.37E+01	4.87E+01	5.87E+01	4.68E+01	3.89E+01	4.66E+01	<b>3.83E+01</b>
	SD	3.61E+00	2.28E+00	4.04E+00	3.11E+00	4.07E+00	2.74E+00	1.48E+00	2.41E+00	<b>1.34E+00</b>
$f_8$	Mean	2.53E+01	1.71E+01	2.42E+01	2.12E+01	2.48E+01	1.90E+01	1.07E+01	1.76E+01	<b>9.44E+00</b>
	SD	3.89E+00	2.60E+00	4.22E+00	4.14E+00	4.20E+00	4.98E+00	2.17E+00	2.50E+00	<b>1.95E+00</b>
$f_9$	Mean	5.27E-03	<b>0.00E+00</b>	2.13E-02	9.76E-02	2.44E-01	5.32E-02	1.74E-01	<b>0.00E+00</b>	8.91E-03
	SD	2.78E-02	<b>0.00E+00</b>	9.01E-02	2.11E-01	4.12E-01	1.37E-01	2.30E-01	<b>0.00E+00</b>	6.36E-02
$f_{10}$	Mean	1.93E+03	1.71E+03	1.87E+03	1.94E+03	1.88E+03	1.94E+03	1.86E+03	1.69E+03	<b>1.63E+03</b>
	SD	1.81E+02	1.90E+02	2.36E+02	2.52E+02	2.76E+02	2.35E+02	3.05E+02	1.93E+02	<b>3.58E+02</b>
$f_{11}$	Mean	2.73E+01	2.02E+01	3.00E+01	4.45E+01	3.26E+01	3.41E+01	1.62E+01	1.78E+01	<b>1.16E+01</b>
	SD	2.21E+01	2.04E+01	2.53E+01	2.89E+01	1.63E+01	2.64E+01	1.66E+01	2.30E+01	<b>1.73E+01</b>
$f_{12}$	Mean	1.19E+03	1.28E+03	1.24E+03	1.17E+03	3.29E+03	1.14E+03	9.64E+02	1.07E+03	<b>3.22E+02</b>
	SD	4.20E+02	4.22E+02	3.53E+02	3.37E+02	2.92E+03	3.60E+02	1.09E+03	3.56E+02	<b>1.77E+02</b>
$f_{13}$	Mean	4.44E+02	3.91E+01	5.43E+01	5.21E+01	4.77E+01	3.67E+01	2.13E+01	2.52E+01	<b>1.52E+01</b>
	SD	2.80E+03	3.87E+01	7.62E+01	7.99E+01	3.09E+01	1.57E+01	8.29E+00	7.42E+00	<b>6.51E+00</b>
$f_{14}$	Mean	8.26E+03	2.71E+01	4.86E+03	3.29E+01	9.15E+03	2.89E+01	1.81E+01	2.44E+01	<b>1.39E+01</b>
	SD	1.37E+04	7.20E+00	9.92E+03	9.46E+00	1.27E+04	4.62E+00	9.68E+00	5.55E+00	<b>1.00E+01</b>
$f_{15}$	Mean	1.36E+03	1.32E+01	3.10E+02	3.94E+01	1.73E+02	2.48E+01	7.32E+00	8.68E+00	<b>2.33E+00</b>
	SD	3.58E+03	7.81E+00	1.47E+03	2.82E+01	9.68E+02	1.92E+01	4.46E+00	4.68E+00	<b>1.26E+00</b>
$f_{16}$	Mean	4.50E+02	3.01E+02	3.72E+02	3.36E+02	4.52E+02	3.47E+02	1.56E+02	3.27E+02	<b>1.50E+02</b>
	SD	1.53E+02	1.29E+02	1.58E+02	1.45E+02	1.50E+02	1.24E+02	1.17E+02	1.35E+02	<b>1.20E+02</b>
$f_{17}$	Mean	7.36E+01	4.47E+01	7.89E+01	6.79E+01	7.71E+01	7.60E+01	3.82E+01	4.28E+01	<b>3.46E+01</b>
	SD	2.19E+01	9.30E+00	3.31E+01	1.45E+01	2.63E+01	2.32E+01	1.21E+01	7.40E+00	<b>1.04E+01</b>
$f_{18}$	Mean	2.97E+04	7.81E+01	7.26E+03	1.06E+02	1.82E+04	6.77E+01	2.31E+01	3.28E+01	<b>2.09E+01</b>
	SD	6.94E+04	4.87E+01	3.59E+04	7.41E+01	4.66E+04	4.77E+01	3.49E+00	1.48E+01	<b>7.23E-01</b>
$f_{19}$	Mean	1.53E+03	9.68E+00	1.91E+02	1.87E+01	7.72E+02	1.13E+01	4.54E+00	6.90E+00	<b>2.99E+00</b>
	SD	5.00E+03	4.02E+00	1.25E+03	1.98E+01	2.87E+03	6.73E+00	1.60E+00	1.91E+00	<b>1.17E+00</b>
$f_{20}$	Mean	1.42E+02	6.76E+01	1.15E+02	1.18E+02	1.39E+02	1.06E+02	2.45E+01	5.90E+01	<b>2.31E+01</b>
	SD	5.43E+01	4.57E+01	5.65E+01	5.49E+01	5.67E+01	5.06E+01	1.48E+01	4.20E+01	<b>9.91E+00</b>
$f_{21}$	Mean	2.27E+02	2.18E+02	2.25E+02	2.20E+02	2.26E+02	2.19E+02	2.11E+02	2.18E+02	<b>2.10E+02</b>
	SD	4.89E+00	2.97E+00	3.55E+00	4.39E+00	4.23E+00	4.51E+00	1.84E+00	2.80E+00	<b>2.46E+00</b>
$f_{22}$	Mean	<b>1.00E+02</b>	<b>1.00E+02</b>	<b>1.00E+02</b>	<b>1.00E+02</b>	1.01E+02	<b>1.00E+02</b>	<b>1.00E+02</b>	<b>1.00E+02</b>	<b>1.00E+02</b>
	SD	2.57E+00	2.26E-13	2.28E+00	2.00E+00	3.02E+00	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_{23}$	Mean	3.73E+02	3.64E+02	3.72E+02	3.67E+02	3.70E+02	3.67E+02	3.56E+02	3.63E+02	<b>3.53E+02</b>
	SD	6.01E+00	5.32E+00	5.60E+00	7.17E+00	7.00E+00	6.95E+00	5.03E+00	4.75E+00	<b>4.88E+00</b>
$f_{24}$	Mean	4.40E+02	4.33E+02	4.40E+02	4.37E+02	4.42E+02	4.35E+02	4.29E+02	4.34E+02	<b>4.28E+02</b>
	SD	5.40E+00	4.04E+00	4.60E+00	3.74E+00	6.71E+00	4.56E+00	2.37E+00	4.08E+00	<b>2.94E+00</b>
$f_{25}$	Mean	<b>3.87E+02</b>	<b>3.87E+02</b>	<b>3.87E+02</b>	<b>3.87E+02</b>	<b>3.87E+02</b>	<b>3.87E+02</b>	<b>3.87E+02</b>	<b>3.87E+02</b>	<b>3.87E+02</b>
	SD	5.37E-01	1.51E-01	1.37E-01	1.42E-01	2.17E+00	1.47E-01	7.41E-02	1.07E-01	<b>2.07E-02</b>
$f_{26}$	Mean	1.18E+03	1.08E+03	1.17E+03	1.09E+03	1.21E+03	1.11E+03	1.00E+03	1.07E+03	<b>9.57E+02</b>
	SD	7.23E+01	6.39E+01	7.35E+01	7.25E+01	7.17E+01	7.31E+01	7.01E+01	7.23E+01	<b>6.48E+01</b>
$f_{27}$	Mean	5.06E+02	5.04E+02	5.05E+02	5.05E+02	5.06E+02	5.03E+02	5.02E+02	5.05E+02	<b>5.01E+02</b>
	SD	7.47E+00	6.90E+00	8.92E+00	6.16E+00	6.55E+00	6.99E+00	6.03E+00	7.04E+00	<b>7.72E+00</b>
$f_{28}$	Mean	3.49E+02	3.25E+02	3.39E+02	3.45E+02	3.24E+02	3.24E+02	3.22E+02	3.29E+02	<b>3.21E+02</b>
	SD	5.79E+01	4.58E+01	5.65E+01	6.06E+01	4.62E+01	4.62E+01	4.55E+01	4.95E+01	<b>4.28E+01</b>
$f_{29}$	Mean	4.85E+02	4.54E+02	4.82E+02	4.74E+02	4.74E+02	4.68E+02	<b>4.27E+02</b>	4.57E+02	<b>4.28E+02</b>
	SD	3.17E+01	1.90E+01	2.92E+01	3.38E+01	2.35E+01	2.14E+01	<b>1.99E+01</b>	2.64E+01	<b>1.43E+01</b>
$f_{30}$	Mean	2.19E+03	2.10E+03	2.13E+03	2.12E+03	2.16E+03	2.18E+03	2.03E+03	2.06E+03	<b>1.97E+03</b>
	SD	4.90E+02	1.36E+02	1.29E+02	1.50E+02	1.83E+02	1.62E+02	7.22E+01	9.18E+01	<b>2.60E+01</b>
	Nob	4	5	4	4	2	4	6	5	25

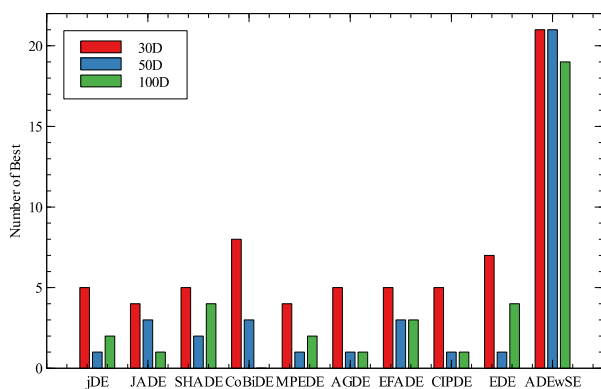


FIGURE 2. The number of the best results obtained by each advanced DE algorithm on CEC2017 with 30D, 50D and 100D

with 100D are chosen to plot the convergence graph of ten algorithms with respect to the mean fitness values. The convergence graphs plotted in Figure 1 show that ADEwSE gets the best results in most cases. However, the convergence graphs imply that the convergence speed of ADEwSE is slower than JADE and SHADE at the early stage of the searching process. The reason is that the modified parameter settings like opposition learning of crossover rate can strengthen the exploration ability, and avoid trapping into a local optimum.

Overall, from the above experimental results, comparison and analysis, the proposed ADEwSE is a highly competitive optimization algorithm with better searching quality, efficiency, and robustness for solving unconstrained global optimization problems.

**TABLE 9. Experimental results of ADEwSE, JADE, SHADE, JADERank, JADEcr, JADEEP, JADEpbestr, JADEsort, and DbSHADE on 50D**

		JADE	SHADE	JADERank	JADEcr	JADEEP	JADEpbestr	JADEsort	DbSHADE	ADEwSE
$f_1$	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	4.42E+03	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
	SD	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	4.77E+03	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_3$	Mean	3.38E+04	<b>0.00E+00</b>	1.61E+04	<b>0.00E+00</b>	1.97E+04	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
	SD	4.55E+04	<b>0.00E+00</b>	3.58E+04	<b>0.00E+00</b>	3.16E+04	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_4$	Mean	4.03E+01	5.43E+01	4.96E+01	4.62E+01	5.07E+01	5.63E+01	<b>2.96E+01</b>	4.73E+01	5.03E+01
	SD	4.60E+01	4.54E+01	4.81E+01	4.53E+01	4.29E+01	4.63E+01	<b>3.79E+01</b>	4.59E+01	4.42E+01
$f_5$	Mean	5.51E+01	3.56E+01	5.15E+01	4.33E+01	4.68E+01	4.07E+01	2.91E+01	3.60E+01	<b>2.47E+01</b>
	SD	7.00E+00	6.27E+00	7.86E+00	7.70E+00	6.63E+00	6.73E+00	4.24E+00	5.53E+00	<b>3.95E+00</b>
$f_6$	Mean	<b>0.00E+00</b>	8.28E-05	<b>0.00E+00</b>	1.96E-04	9.60E-05	7.85E-05	2.03E-02	1.04E-05	2.62E-06
	SD	<b>0.00E+00</b>	1.81E-04	<b>0.00E+00</b>	1.74E-04	2.01E-04	9.56E-05	2.05E-02	1.27E-05	3.48E-06
$f_7$	Mean	9.91E+01	8.32E+01	1.02E+02	9.23E+01	9.67E+01	9.10E+01	7.40E+01	8.16E+01	<b>6.89E+01</b>
	SD	7.17E+00	4.40E+00	8.13E+00	6.23E+00	6.38E+00	5.97E+00	4.48E+00	5.22E+00	<b>3.26E+00</b>
$f_8$	Mean	5.31E+01	3.64E+01	5.03E+01	4.74E+01	4.74E+01	4.16E+01	2.90E+01	3.52E+01	<b>2.51E+01</b>
	SD	7.08E+00	5.74E+00	8.48E+00	6.72E+00	7.06E+00	6.28E+00	5.27E+00	4.93E+00	<b>4.27E+00</b>
$f_9$	Mean	9.98E-01	6.71E-01	7.32E-01	3.77E-01	1.05E+00	2.09E+00	1.33E+01	2.31E-01	<b>9.25E-02</b>
	SD	7.76E-01	7.96E-01	6.81E-01	2.44E+00	1.07E+00	1.73E+00	1.02E+01	2.81E-01	<b>1.94E-01</b>
$f_{10}$	Mean	3.75E+03	3.32E+03	3.76E+03	3.83E+03	3.51E+03	3.83E+03	3.37E+03	3.31E+03	<b>2.78E+03</b>
	SD	2.95E+02	3.04E+02	2.40E+02	2.62E+02	3.24E+02	2.77E+02	3.22E+02	3.88E+02	<b>6.82E+02</b>
$f_{11}$	Mean	1.36E+02	1.02E+02	1.30E+02	1.45E+02	1.02E+02	1.36E+02	9.38E+01	6.88E+01	<b>4.49E+01</b>
	SD	4.18E+01	2.37E+01	3.17E+01	3.49E+01	3.69E+01	3.13E+01	2.28E+01	1.73E+01	<b>1.32E+01</b>
$f_{12}$	Mean	4.61E+03	4.91E+03	5.92E+03	5.92E+03	7.60E+03	6.04E+03	6.55E+03	5.46E+03	<b>2.89E+03</b>
	SD	3.15E+03	2.79E+03	4.34E+03	4.21E+03	1.92E+04	4.10E+03	6.58E+03	4.28E+03	<b>2.35E+03</b>
$f_{13}$	Mean	2.59E+02	2.65E+02	3.56E+02	4.46E+02	3.50E+03	2.26E+02	1.38E+02	1.42E+02	<b>6.13E+01</b>
	SD	1.54E+02	1.41E+02	2.77E+02	6.86E+02	3.91E+03	1.90E+02	9.18E+01	7.47E+01	<b>2.89E+01</b>
$f_{14}$	Mean	2.52E+04	2.11E+02	6.27E+03	2.34E+02	9.92E+03	1.95E+02	4.79E+01	1.11E+02	<b>2.91E+01</b>
	SD	6.69E+04	7.07E+01	2.09E+04	5.00E+01	1.49E+04	5.20E+01	1.24E+01	4.14E+01	<b>7.03E+00</b>
$f_{15}$	Mean	3.70E+02	2.36E+02	3.88E+02	3.73E+02	2.48E+03	2.47E+02	7.67E+01	1.28E+02	<b>3.31E+01</b>
	SD	4.54E+02	9.64E+01	5.92E+02	1.61E+02	2.73E+03	1.19E+02	3.30E+01	5.46E+01	<b>5.99E+00</b>
$f_{16}$	Mean	8.89E+02	7.35E+02	8.64E+02	8.35E+02	8.54E+02	8.01E+02	5.21E+02	7.41E+02	<b>4.44E+02</b>
	SD	1.98E+02	1.57E+02	1.82E+02	1.97E+02	1.44E+02	1.91E+02	1.82E+02	2.05E+02	<b>1.72E+02</b>
$f_{17}$	Mean	6.10E+02	5.12E+02	6.17E+02	6.00E+02	6.15E+02	5.44E+02	4.05E+02	4.90E+02	<b>3.27E+02</b>
	SD	1.58E+02	1.25E+02	1.48E+02	1.44E+02	1.25E+02	1.27E+02	1.28E+02	1.43E+02	<b>1.12E+02</b>
$f_{18}$	Mean	4.48E+03	1.57E+02	1.80E+02	1.98E+02	4.92E+04	1.39E+02	6.60E+01	1.38E+02	<b>3.30E+01</b>
	SD	3.06E+04	7.44E+01	1.03E+02	1.39E+02	2.00E+05	9.19E+01	4.15E+01	8.84E+01	<b>6.62E+00</b>
$f_{19}$	Mean	4.51E+02	1.29E+02	1.63E+02	1.51E+02	1.04E+04	1.33E+02	4.72E+01	8.80E+01	<b>1.58E+01</b>
	SD	2.23E+03	4.65E+01	5.81E+01	5.04E+01	7.45E+03	4.52E+01	2.33E+01	3.64E+01	<b>3.67E+00</b>
$f_{20}$	Mean	5.50E+02	3.17E+02	5.27E+02	5.02E+02	5.01E+02	4.97E+02	1.60E+02	3.26E+02	<b>1.37E+02</b>
	SD	1.18E+02	1.13E+02	1.25E+02	1.24E+02	1.27E+02	1.36E+02	1.26E+02	1.10E+02	<b>1.10E+02</b>
$f_{21}$	Mean	2.54E+02	2.37E+02	2.52E+02	2.48E+02	2.49E+02	2.42E+02	2.27E+02	2.36E+02	<b>2.24E+02</b>
	SD	8.06E+00	4.98E+00	9.15E+00	8.38E+00	6.88E+00	7.97E+00	3.97E+00	5.57E+00	<b>2.72E+00</b>
$f_{22}$	Mean	<b>2.65E+03</b>	2.74E+03	3.54E+03	3.73E+03	3.58E+03	3.81E+03	3.31E+03	3.48E+03	2.79E+03
	SD	<b>2.05E+03</b>	1.82E+03	1.63E+03	1.48E+03	1.62E+03	1.51E+03	1.56E+03	1.39E+03	1.38E+03
$f_{23}$	Mean	4.79E+02	4.57E+02	4.80E+02	4.69E+02	4.72E+02	4.63E+02	4.48E+02	4.59E+02	<b>4.44E+02</b>
	SD	1.16E+01	7.70E+00	1.20E+01	1.11E+01	1.11E+01	1.02E+01	9.53E+00	7.55E+00	<b>8.00E+00</b>
$f_{24}$	Mean	5.41E+02	5.27E+02	5.41E+02	5.33E+02	5.36E+02	5.31E+02	5.20E+02	5.26E+02	<b>5.18E+02</b>
	SD	8.99E+00	7.32E+00	1.03E+01	8.16E+00	7.76E+00	7.17E+00	6.63E+00	7.54E+00	<b>5.48E+00</b>
$f_{25}$	Mean	5.23E+02	5.28E+02	5.20E+02	5.16E+02	5.23E+02	<b>5.12E+02</b>	5.41E+02	5.30E+02	5.28E+02
	SD	3.34E+01	3.15E+01	3.88E+01	3.45E+01	4.03E+01	<b>3.85E+01</b>	2.83E+01	3.43E+01	3.29E+01
$f_{26}$	Mean	1.63E+03	1.44E+03	1.60E+03	1.51E+03	1.57E+03	1.47E+03	1.36E+03	1.40E+03	<b>1.27E+03</b>
	SD	9.74E+01	1.01E+02	1.24E+02	1.09E+02	1.04E+02	8.11E+01	8.52E+01	9.76E+01	<b>1.05E+02</b>
$f_{27}$	Mean	5.57E+02	5.39E+02	5.62E+02	5.64E+02	5.61E+02	5.51E+02	5.46E+02	5.32E+02	<b>5.18E+02</b>
	SD	3.03E+01	1.78E+01	3.59E+01	3.57E+01	2.84E+01	2.69E+01	2.39E+01	1.44E+01	<b>1.40E+01</b>
$f_{28}$	Mean	4.92E+02	4.97E+02	<b>4.87E+02</b>	4.94E+02	4.92E+02	4.89E+02	4.95E+02	4.91E+02	4.90E+02
	SD	2.12E+01	1.81E+01	<b>2.39E+01</b>	2.29E+01	2.08E+01	2.26E+01	1.73E+01	2.14E+01	2.19E+01
$f_{29}$	Mean	4.75E+02	4.73E+02	4.64E+02	4.58E+02	4.66E+02	4.65E+02	4.12E+02	4.43E+02	<b>3.62E+02</b>
	SD	8.28E+01	7.77E+01	7.37E+01	7.16E+01	9.16E+01	7.23E+01	5.39E+01	8.27E+01	<b>3.98E+01</b>
$f_{30}$	Mean	6.76E+05	6.25E+05	6.79E+05	6.93E+05	6.48E+05	7.00E+05	6.19E+05	6.52E+05	<b>6.05E+05</b>
	SD	8.69E+04	4.95E+04	1.06E+05	9.72E+04	7.69E+04	9.86E+04	4.32E+04	5.60E+04	<b>2.82E+04</b>
	Nob	3	2	3	2	0	3	3	2	24

**D. COMPARISON WITH OTHER “CURRENT-TO-PBEST/1” BASED ADAPTIVE DIFFERENTIAL EVOLUTION ALGORITHMS**

In this subsection, considering that the mutation strategy in ADEwSE is based on the classical “current-to-pbest/1” mutation strategy, it’s necessary to compare it with some up-to-date “current-to-best/1” mutation strategy based DE algorithms such as JADE, SHADE, JADEEP, JADERank, JADEcr, JADEpbestr, JADEsort, and DbSHADE. For a fair comparison, all parameters of compared algorithms in this part are kept the same as their original literature. The obtained mean values and standard deviations of errors on 30D,50D and 100D are provided in Tables 8, 9, and 10, respectively, in

which the minimum results are marked in bold. Besides, the number of cases on which each algorithm performs the best is listed at the bottom of the Tables. Table 11 and Tables 12, 13 respectively represent the statistical results obtained by multi-problem Wilcoxon signed-rank and Friedman tests. The histogram of the number of the best results of each algorithm is shown in Figure 3.

From Tables 8, 9 and 10, it can be seen that at D=30, ADEwSE obtains the best results in 25 cases, while JADE, SHADE, JADERank, JADEcr, JADEEP, JADEpbestr, JADEsort, and DbSHADE obtain the best results on 4, 5, 4, 4, 2, 4, 6, and 5 functions, respectively. Concerning 50D functions, JADE, SHADE, JADERank, JADEcr, JADEEP,



**TABLE 10. Experimental results of ADEwSE, JADE, SHADE, JADERank, JADEcr, JADEEP, JADEpbestrr, JADEsort, and DbSHADE on 100D**

		JADE	SHADE	JADERank	JADEcr	JADEEP	JADEpbestrr	JADEsort	DbSHADE	ADEwSE
$f_1$	Mean	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	4.24E+03	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
	SD	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	4.20E+03	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_3$	Mean	8.25E+04	<b>1.56E-08</b>	9.38E+04	1.27E+04	1.28E+05	4.14E+04	5.92E-03	3.45E-06	1.62E-05
	SD	1.51E+05	<b>2.65E-08</b>	1.48E+05	6.37E+04	1.38E+05	1.06E+05	9.10E-03	1.90E-05	5.21E-05
$f_4$	Mean	7.36E+01	7.18E+01	1.20E+02	7.71E+01	9.64E+01	7.26E+01	2.06E+01	3.96E+01	<b>1.68E+01</b>
	SD	6.60E+01	6.50E+01	7.04E+01	6.76E+01	6.62E+01	6.53E+01	3.73E+01	5.56E+01	<b>4.69E+01</b>
$f_5$	Mean	1.48E+02	1.19E+02	1.32E+02	1.33E+02	1.47E+02	1.23E+02	1.16E+02	1.09E+02	<b>1.04E+02</b>
	SD	1.81E+01	1.39E+01	1.83E+01	1.95E+01	1.88E+01	1.74E+01	2.01E+01	1.33E+01	<b>1.43E+01</b>
$f_6$	Mean	<b>9.91E-04</b>	9.61E-02	2.57E-03	2.67E-02	1.97E-01	1.42E-02	2.00E+00	1.44E-02	2.46E-02
	SD	<b>2.73E-03</b>	6.80E-02	6.78E-03	3.83E-02	1.21E-01	1.85E-02	8.29E-01	1.80E-02	2.16E-02
$f_7$	Mean	2.69E+02	2.22E+02	2.59E+02	2.70E+02	2.63E+02	2.49E+02	2.56E+02	<b>2.08E+02</b>	2.15E+02
	SD	2.41E+01	1.71E+01	2.38E+01	2.10E+01	2.17E+01	2.00E+01	2.43E+01	<b>1.79E+01</b>	1.83E+01
$f_8$	Mean	1.45E+02	1.13E+02	1.35E+02	1.36E+02	1.36E+02	1.20E+02	1.20E+02	1.08E+02	<b>1.03E+02</b>
	SD	1.84E+01	1.13E+01	1.53E+01	1.67E+01	1.84E+01	1.55E+01	1.27E+01	1.20E+01	<b>1.59E+01</b>
$f_9$	Mean	8.90E+01	4.74E+01	5.97E+01	1.63E+02	6.83E+01	9.89E+01	1.19E+03	<b>2.22E+01</b>	1.43E+02
	SD	1.03E+02	3.29E+01	4.00E+01	1.09E+02	5.27E+01	5.38E+01	3.57E+02	<b>1.51E+01</b>	1.47E+02
$f_{10}$	Mean	1.03E+04	9.31E+03	1.02E+04	1.02E+04	1.00E+04	1.03E+04	9.15E+03	9.37E+03	<b>8.37E+03</b>
	SD	4.84E+02	5.08E+02	5.46E+02	5.73E+02	5.49E+02	4.56E+02	5.84E+02	4.63E+02	<b>1.09E+03</b>
$f_{11}$	Mean	4.77E+02	1.09E+03	3.79E+03	3.79E+03	8.26E+02	2.04E+03	4.39E+02	9.00E+02	<b>3.22E+02</b>
	SD	3.76E+03	2.54E+02	3.32E+03	2.70E+03	2.65E+02	2.75E+03	1.47E+02	1.95E+02	<b>9.36E+01</b>
$f_{12}$	Mean	1.86E+04	<b>1.67E+04</b>	2.24E+04	2.28E+04	5.00E+05	2.09E+04	2.68E+04	1.78E+04	1.83E+04
	SD	9.71E+03	<b>7.72E+03</b>	2.40E+04	2.33E+04	5.95E+05	8.95E+03	1.05E+04	8.17E+03	8.54E+03
$f_{13}$	Mean	2.79E+03	1.36E+03	3.88E+03	2.98E+03	3.36E+03	2.39E+03	6.91E+02	1.43E+03	<b>3.75E+02</b>
	SD	2.84E+03	2.68E+03	3.80E+03	3.74E+03	3.46E+03	3.08E+03	3.29E+02	2.44E+03	<b>5.43E+02</b>
$f_{14}$	Mean	5.89E+02	5.72E+02	6.55E+02	6.74E+02	6.71E+02	6.58E+02	3.33E+02	4.76E+02	<b>2.18E+02</b>
	SD	2.10E+02	2.87E+02	2.23E+02	2.27E+02	2.48E+02	1.72E+02	1.08E+02	2.04E+02	<b>4.73E+01</b>
$f_{15}$	Mean	3.74E+02	3.35E+02	3.59E+02	3.63E+02	1.48E+03	4.44E+02	2.70E+02	3.43E+02	<b>2.75E+02</b>
	SD	8.57E+01	9.31E+01	8.97E+01	9.03E+01	1.24E+03	4.07E+02	5.91E+01	1.63E+02	<b>8.03E+01</b>
$f_{16}$	Mean	2.64E+03	2.33E+03	2.51E+03	2.57E+03	2.48E+03	2.54E+03	<b>2.01E+03</b>	2.43E+03	2.03E+03
	SD	3.06E+02	2.48E+02	2.45E+02	3.10E+02	3.47E+02	3.85E+02	<b>4.10E+02</b>	2.69E+02	3.83E+02
$f_{17}$	Mean	1.89E+03	1.88E+03	1.85E+03	1.95E+03	1.92E+03	1.92E+03	1.65E+03	1.81E+03	<b>1.34E+03</b>
	SD	2.41E+02	2.61E+02	2.66E+02	2.46E+02	2.55E+02	2.43E+02	2.84E+02	2.47E+02	<b>3.29E+02</b>
$f_{18}$	Mean	1.80E+03	1.56E+03	1.71E+03	1.94E+03	1.13E+03	1.64E+03	2.60E+03	1.95E+03	<b>4.90E+02</b>
	SD	1.35E+03	1.03E+03	1.12E+03	1.52E+03	6.49E+02	1.12E+03	2.39E+03	1.26E+03	<b>3.41E+02</b>
$f_{19}$	Mean	1.46E+03	4.87E+02	2.25E+03	1.01E+03	2.16E+03	5.74E+02	2.04E+02	3.69E+02	<b>1.48E+02</b>
	SD	2.25E+03	7.85E+02	3.31E+03	1.85E+03	2.24E+03	1.00E+03	4.56E+01	8.98E+02	<b>3.91E+01</b>
$f_{20}$	Mean	2.13E+03	1.73E+03	2.14E+03	2.10E+03	2.15E+03	2.09E+03	1.40E+03	1.70E+03	<b>1.38E+03</b>
	SD	2.29E+02	2.38E+02	2.53E+02	2.12E+02	2.43E+02	2.53E+02	3.91E+02	2.22E+02	<b>3.16E+02</b>
$f_{21}$	Mean	3.69E+02	3.39E+02	3.62E+02	3.63E+02	3.66E+02	3.42E+02	3.34E+02	3.29E+02	<b>3.13E+02</b>
	SD	1.74E+01	1.65E+01	2.03E+01	1.91E+01	2.24E+01	1.41E+01	1.67E+01	1.42E+01	<b>1.35E+01</b>
$f_{22}$	Mean	1.13E+04	1.07E+04	1.12E+04	1.14E+04	1.12E+04	1.14E+04	1.06E+04	1.06E+04	<b>9.50E+03</b>
	SD	6.36E+02	5.14E+02	5.54E+02	5.00E+02	5.54E+02	5.73E+02	7.54E+02	5.01E+02	<b>1.07E+03</b>
$f_{23}$	Mean	6.52E+02	6.15E+02	6.46E+02	6.45E+02	6.37E+02	6.40E+02	6.60E+02	<b>6.06E+02</b>	6.33E+02
	SD	1.20E+01	1.43E+01	1.41E+01	1.38E+01	1.57E+01	1.60E+01	3.68E+01	<b>1.62E+01</b>	3.25E+01
$f_{24}$	Mean	1.02E+03	9.91E+02	1.02E+03	1.01E+03	1.01E+03	1.00E+03	1.02E+03	<b>9.78E+02</b>	9.82E+02
	SD	1.99E+01	2.17E+01	2.29E+01	2.06E+01	2.07E+01	1.75E+01	2.07E+01	<b>1.88E+01</b>	2.18E+01
$f_{25}$	Mean	7.69E+02	7.40E+02	<b>7.37E+02</b>	7.40E+02	7.66E+02	7.61E+02	7.76E+02	7.49E+02	7.53E+02
	SD	4.70E+01	5.00E+01	<b>5.77E+01</b>	5.70E+01	5.45E+01	5.51E+01	5.32E+01	5.23E+01	5.14E+01
$f_{26}$	Mean	4.59E+03	4.37E+03	4.50E+03	4.53E+03	4.51E+03	4.26E+03	4.62E+03	4.17E+03	<b>4.14E+03</b>
	SD	2.72E+02	2.78E+02	2.55E+02	2.24E+02	2.15E+02	2.19E+02	3.05E+02	2.29E+02	<b>2.83E+02</b>
$f_{27}$	Mean	7.29E+02	6.97E+02	7.06E+02	7.31E+02	7.16E+02	7.20E+02	7.03E+02	6.76E+02	<b>6.18E+02</b>
	SD	3.57E+01	2.64E+01	3.18E+01	3.38E+01	3.79E+01	3.90E+01	4.01E+01	2.84E+01	<b>2.75E+01</b>
$f_{28}$	Mean	5.37E+02	5.26E+02	5.17E+02	5.37E+02	5.34E+02	5.32E+02	5.15E+02	5.17E+02	<b>4.86E+02</b>
	SD	4.52E+01	4.82E+01	4.99E+01	4.64E+01	3.34E+01	4.30E+01	5.18E+01	5.53E+01	<b>7.92E+01</b>
$f_{29}$	Mean	2.20E+03	2.11E+03	2.25E+03	2.22E+03	2.25E+03	2.11E+03	1.78E+03	2.01E+03	<b>1.59E+03</b>
	SD	3.24E+02	3.07E+02	2.74E+02	3.54E+02	3.64E+02	2.91E+02	3.20E+02	3.07E+02	<b>3.13E+02</b>
$f_{30}$	Mean	3.53E+03	2.60E+03	3.39E+03	4.12E+03	5.29E+03	4.41E+03	3.45E+03	<b>2.58E+03</b>	2.63E+03
	SD	1.61E+03	2.28E+02	1.44E+03	2.60E+03	3.00E+03	2.26E+03	1.20E+03	<b>2.24E+02</b>	3.93E+02
	Nob	2	3	2	1	0	1	2	6	<b>19</b>

JADEpbestrr, JADEsort, and DbSHADE achieve the best results in 3, 2, 3, 2, 0, 3, 3, 2 cases, respectively, while ADEwSE achieves that in 24 cases. About 100D functions, ADEwSE gets the best results in 19 cases, while JADE, SHADE, JADERank, JADEcr, JADEEP, JADEpbestrr, JADEsort, and DbSHADE achieve 2, 3, 2, 1, 0, 1, 2, 6 cases, respectively. In addition, the number of the best results achieved by each algorithm with all dimensions as depicted in Figure 3 further demonstrates the superior performance of the proposed algorithm. Furthermore, in Table 11, the statistical results of applying multi-problem Wilcoxon's test between ADEwSE and other "current-to-pbest/l" based algorithms are summarized. It clearly shows that ADEwSE gets higher  $R^+$  values than  $R^-$  values in all cases. According to the Wilcoxon's test at  $\alpha = 0.05$  and  $\alpha = 0.1$ , the significance

difference can be found in all cases, which means that ADEwSE performs significantly better than other compared algorithms. Additionally, Table 12 lists the average rankings of ADEwSE and other "current-to-pbest" based adaptative differential evolution algorithms. ADEwSE is ranked the first among the nine methods for 29 test problems with 30D, 50D and 100D, based on the average ranking achieved by the Friedman test. Furthermore, the performance of different algorithms is evaluated by using all dimensions functions. Therefore, the mean aggregated rank of all nine algorithms for all dimensions and problems is shown in Table 13. According to the statistical results in Table 13, one can conclude that ADEwSE is the best followed by DbSHADE as second best among all algorithms.

In a word, it is clear that the performance of proposed ADEwSE is significantly better or at least very competitive to those eight famous “current-to-pbest/1” based adaptive algorithms on CEC2017 with 30D, 50D, and 100D.

**E. COMPARISON WITH JADE AND JADESORT VARIANTS ALGORITHMS**

Considering that the proposed ADEwSE shares some similarities with JADE and JADEsort algorithms, a detailed discussion of ADEwSE with JADE and JADEsort algorithms is presented. In this comparison, all test instances at D=50 are studied. Since there are four mutation strategies in original JADE, there are four JADE, four JADEsort, and four ADEwSE

**TABLE 11. Results of the multi-problem Wilcoxon’s test for ADEwSE versus the “current-to-pbest/1” based algorithms**

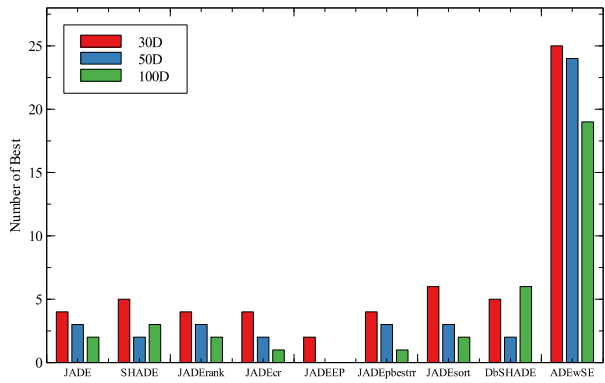
Algorithm at 30D	R <sup>+</sup>	R <sup>-</sup>	p-value	$\alpha=0.05$	$\alpha=0.1$
JADE	397.5	8.5	9.0E-6	Yes	Yes
SHADE	417.5	18.0	1.5E-6	Yes	Yes
JADEEP	417.0	16.5	1.3E-5	Yes	Yes
JADERank	401.5	4.5	6.0E-6	Yes	Yes
JADEcr	430.0	5.0	4.0E-6	Yes	Yes
JADEpbestr	430.0	19.0	4.0E-6	Yes	Yes
JADEsort	394.0	41.0	5.2E-5	Yes	Yes
DbSHADE	417.0	18.0	3.1E-4	Yes	Yes

Algorithm at 50D	R <sup>+</sup>	R <sup>-</sup>	p-value	$\alpha=0.05$	$\alpha=0.1$
JADE	389.0	46.0	1.3E-5	Yes	Yes
SHADE	381.0	25.0	4.8E-5	Yes	Yes
ADEEP	428.5	6.5	8.0E-6	Yes	Yes
JADERank	357.0	78.0	1.4E-5	Yes	Yes
JADEcr	418.5	16.5	1.6E-5	Yes	Yes
JADEpbestr	400.0	35.0	1.4E-5	Yes	Yes
JADEsort	406.0	29.0	6.4E-4	Yes	Yes
DbSHADE	405.0	1.0	1.0E-4	Yes	Yes

Algorithm at 100D	R <sup>+</sup>	R <sup>-</sup>	p-value	$\alpha=0.05$	$\alpha=0.1$
JADE	395.0	9.5	9.0E-6	Yes	Yes
SHADE	339.0	67.0	1.9E-3	Yes	Yes
JADEEP	425.0	10.0	6.0E-6	Yes	Yes
JADERank	392.0	14.0	1.5E-5	Yes	Yes
JADEcr	403.0	3.0	5.0E-6	Yes	Yes
JADEpbestr	396.0	10.0	1.1E-5	Yes	Yes
JADEsort	399.5	6.5	7.0E-6	Yes	Yes
DbSHADE	356.0	79.0	2.5E-3	Yes	Yes



**FIGURE 3. The number of the best results obtained by each “current-to-pbest/1” based variant algorithm CEC2017 with 30D, 50D and 100D in the comparison**

SE variants. They are:

- ADEwSE-S1, JADE-S1, and JADEsort-S1: based on “current-to-pbest/1 (without archive)”
- ADEwSE-S2, JADE-S2, and JADEsort-S2: based on “rand-to-pbest/1 (without archive)”

- ADEwSE-S3, JADE-S3, and JADEsort-S3: based on “current-to-pbest/1 (with archive)”
- ADEwSE-S4, JADE-S4, and JADEsort-S4: based on “rand-to-pbest/1 (with archive)”

The mutation strategies in JADE are presented as follows: current-to-pbest/1 without archive (denoted as S1)

$$v_i^G = x_i^G + F_i(x_{pbest}^G - x_i^G) + F_i(x_{r1}^G - x_{r2}^G) \quad (29)$$

rand-to-pbest/1 without archive (denoted as S2)

$$v_i^G = x_{r1}^G + F_i(x_{pbest}^G - x_{r1}^G) + F_i(x_{r2}^G - x_{r3}^G) \quad (30)$$

current-to-pbest/1 with archive (denoted as S3)

$$v_i^G = x_i^G + F_i(x_{pbest}^G - x_i^G) + F_i(x_{r1}^G - \bar{x}_{r2}^G) \quad (31)$$

rand-to-pbest/1 with archive (denoted as S4)

$$v_i^G = x_{r1}^G + F_i(x_{pbest}^G - x_{r1}^G) + F_i(x_{r2}^G - \bar{x}_{r3}^G) \quad (32)$$

**TABLE 12. Average ranking values (ARV) of ADEwSE and other “current-to-pbest/1” based algorithms**

Rank	D=30		D=50		D=100	
	Algorithm	ARV	Algorithm	ARV	Algorithm	ARV
1	ADEwSE	1.83	ADEwSE	2.17	ADEwSE	1.97
2	JADEsort	2.83	JADEsort	3.36	DbSHADE	3.03
3	DbSHADE	3.55	DbSHADE	3.47	SHADE	3.45
4	SHADE	4.03	SHADE	4.43	JADEsort	4.41
5	JADEpbestr	5.29	JADEpbestr	5.21	JADEpbestr	5.64
6	JADEcr	6.12	JADEcr	6.28	JADERank	5.86
7	JADERank	6.72	JADERank	6.53	JADEcr	6.86
8	JADEEP	7.26	JADE	6.59	JADEEP	6.88
9	JADE	7.36	JADEEP	6.97	JADE	6.90

**TABLE 13. Average ranks for all algorithms across all problems**

Ranks	Algorithm	30D	50D	100D	Mean ranking
1	ADEwSE	1.83	2.17	1.97	1.99
2	DbSHADE	3.55	3.47	3.03	3.35
3	JADEsort	2.83	3.36	4.41	3.53
4	SHADE	4.03	4.43	3.45	3.97
5	JADEpbestr	5.29	5.21	5.64	5.38
6	JADERank	6.72	6.53	5.86	6.37
7	JADEcr	6.12	6.28	6.86	6.42
8	JADE	7.36	6.59	6.90	6.95
9	JADEEP	7.26	6.97	6.88	7.03

In the third and fourth mutation strategies, an archive set is used to store the inferior parent individuals recently explored in the evolutionary process. Besides, the base vectors in the second and fourth mutation strategies are selected randomly.

The statistical results of ADEwSE, JADE, and JADEsort with four mutation strategies are listed in Table 14. Additionally, the compared results obtained by multi-problem Wilcoxon’s test and Friedman test are represented in Tables 15 and 16, respectively.

It is shown that ADEwSE-S3 (standard ADEwSE) obtains the best results in 13 cases in Table 14. Besides, based on Table 14, ADEwSE-S1, ADEwSE-S2, and ADEwSE-S4 achieve best results in 4, 6 and 9 cases, respectively, while JADE-S1, JADE-S2, JADE-S3, JADE-S4, JADEsort-S1, JADEsort-S2, JADEsort-S3, JADEsort-S4 achieve the best results 2, 1, 3, 2, 0, 0, 3, 2 cases, respectively. Furthermore, as shown in Table 15, ADEwSE-S3 performs better, and gets higher R+ values than R- values in all dimensions. According to the statistics of p=0.05 and p=0.1, significant difference can be observed in 12 and 12 cases, respectively, for the functions at 50D. Note that no significant difference can be found between ADEwSE-S3 and ADEwSE-S2, ADEwSE-S4,

TABLE 14. Experimental results of JADE, JADEsort, and ADEwSE variants on 50D

		JADE				JADEsort				ADEwSE			
		S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
f <sub>1</sub>	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.12E+01	3.38E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.20E+01	7.17E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f <sub>5</sub>	Mean	1.73E+04	5.80E+04	3.38E+04	5.96E+04	3.67E-02	3.19E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	3.58E+04	6.04E+04	4.55E+04	6.05E+04	8.46E-02	4.14E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f <sub>4</sub>	Mean	3.81E+01	7.35E+01	4.03E+01	5.51E+01	3.40E+01	4.76E+01	2.96E+01	6.25E+01	3.22E+01	4.71E+01	5.03E+01	5.83E+01
	SD	4.59E+01	4.17E+01	4.60E+01	4.58E+01	4.44E+01	4.56E+01	3.79E+01	4.78E+01	4.21E+01	4.21E+01	4.42E+01	4.44E+01
f <sub>5</sub>	Mean	5.41E+01	6.29E+01	5.51E+01	6.30E+01	3.87E+01	4.47E+01	2.91E+01	3.98E+01	3.07E+01	3.54E+01	2.47E+01	3.34E+01
	SD	8.58E+00	8.51E+00	7.00E+00	9.93E+00	5.80E+00	9.34E+00	4.24E+00	9.35E+00	4.73E+00	7.96E+00	3.95E+00	7.45E+00
f <sub>6</sub>	Mean	0.00E+00	3.21E-08	0.00E+00	0.00E+00	4.59E-02	6.02E-02	2.03E-02	2.49E-02	4.01E-06	3.63E-06	2.62E-06	4.01E-06
	SD	0.00E+00	2.29E-07	0.00E+00	0.00E+00	4.83E-02	5.40E-02	2.05E-02	2.34E-02	8.37E-06	8.92E-06	3.48E-06	1.15E-05
f <sub>7</sub>	Mean	1.02E+02	1.16E+02	9.91E+01	1.15E+02	8.01E+01	8.37E+01	7.40E+01	8.42E+01	7.30E+01	7.67E+01	6.89E+01	7.92E+01
	SD	7.23E+00	7.57E+00	7.17E+00	8.92E+00	5.32E+00	8.77E+00	4.48E+00	8.77E+00	3.34E+00	7.21E+00	3.26E+00	7.79E+00
f <sub>8</sub>	Mean	5.38E+01	6.58E+01	5.31E+01	6.44E+01	3.72E+01	4.44E+01	2.90E+01	3.95E+01	3.21E+01	3.34E+01	2.51E+01	3.29E+01
	SD	7.63E+00	9.46E+00	7.08E+00	8.20E+00	4.93E+00	9.93E+00	5.27E+00	8.71E+00	5.20E+00	5.61E+00	4.27E+00	9.01E+00
f <sub>9</sub>	Mean	1.65E+00	7.84E-01	9.98E-01	2.31E-01	2.56E+01	2.87E+01	1.33E+01	7.89E+00	2.37E-01	2.22E-01	9.25E-02	1.19E-01
	SD	1.50E+00	1.50E+00	7.76E-01	4.53E-01	2.07E+01	1.87E+01	1.02E+01	4.97E+00	4.42E-01	4.05E-01	1.94E-01	2.23E-01
f <sub>10</sub>	Mean	3.76E+03	4.73E+03	3.75E+03	4.75E+03	3.94E+03	4.60E+03	3.37E+03	4.80E+03	3.30E+03	3.98E+03	2.78E+03	4.16E+03
	SD	2.56E+02	3.15E+02	2.95E+02	3.27E+02	8.59E+02	8.32E+02	3.22E+02	7.78E+02	8.43E+02	7.18E+02	6.82E+02	7.82E+02
f <sub>11</sub>	Mean	1.30E+02	1.51E+02	1.36E+02	1.89E+02	8.63E+01	9.63E+01	9.38E+01	9.74E+01	4.20E+01	4.46E+01	4.49E+01	4.08E+01
	SD	3.43E+01	4.98E+01	4.18E+01	5.43E+01	2.46E+01	3.34E+01	2.28E+01	2.54E+01	1.07E+01	1.03E+01	1.32E+01	7.45E+01
f <sub>12</sub>	Mean	6.50E+03	7.99E+03	4.61E+03	5.17E+03	2.75E+04	2.33E+04	6.55E+03	8.27E+03	4.80E+03	6.41E+03	2.89E+03	7.85E+03
	SD	4.11E+03	6.29E+03	3.15E+03	2.69E+03	2.63E+04	1.70E+04	6.58E+03	6.44E+03	3.88E+03	5.55E+03	2.85E+03	9.78E+03
f <sub>13</sub>	Mean	3.20E+02	5.95E+02	2.59E+02	6.66E+02	4.56E+02	1.19E+03	1.38E+02	9.49E+01	7.38E+01	4.66E+01	6.13E+01	4.85E+01
	SD	1.67E+02	8.77E+02	1.54E+02	1.72E+03	4.42E+02	1.17E+03	9.18E+01	4.66E+01	3.17E+01	2.40E+01	2.89E+01	1.97E+01
f <sub>14</sub>	Mean	4.95E+03	7.14E+04	2.52E+04	1.52E+05	5.50E+01	6.79E+01	4.79E+01	6.86E+01	3.03E+01	3.20E+01	2.91E+01	3.18E+01
	SD	2.63E+04	1.11E+05	6.69E+04	1.44E+05	1.10E+01	1.81E+01	1.24E+01	1.91E+01	4.54E+00	5.03E+00	7.03E+00	3.93E+00
f <sub>15</sub>	Mean	2.98E+02	6.11E+02	3.70E+02	1.18E+03	6.68E+01	8.87E+01	7.67E+01	8.19E+01	3.73E+01	3.36E+01	3.31E+01	3.24E+01
	SD	3.29E+02	1.41E+03	4.54E+02	2.22E+03	3.11E+01	5.22E+01	3.30E+01	3.85E+01	9.36E+00	7.64E+00	5.99E+00	5.98E+00
f <sub>16</sub>	Mean	8.55E+02	8.72E+02	8.89E+02	8.90E+02	5.25E+02	6.85E+02	5.21E+02	5.80E+02	4.66E+02	3.57E+02	4.44E+02	3.82E+02
	SD	1.71E+02	2.19E+02	1.98E+02	2.13E+02	1.71E+02	2.95E+02	1.82E+02	2.67E+02	1.66E+02	1.81E+02	1.72E+02	2.03E+02
f <sub>17</sub>	Mean	6.18E+02	7.00E+02	6.10E+02	7.13E+02	4.19E+02	4.39E+02	4.05E+02	4.12E+02	3.19E+02	2.96E+02	3.27E+02	3.15E+02
	SD	1.26E+02	1.24E+02	1.58E+02	1.47E+02	1.27E+02	2.45E+02	1.28E+02	1.87E+02	1.12E+02	1.76E+02	1.12E+02	1.39E+02
f <sub>18</sub>	Mean	3.51E+04	2.82E+05	4.48E+03	6.83E+03	3.01E+02	6.76E+02	6.60E+01	7.68E+01	3.46E+01	3.29E+01	3.30E+01	3.28E+01
	SD	1.78E+05	5.08E+05	3.06E+04	5.98E+05	1.98E+02	4.92E+02	4.15E+01	4.34E+01	7.70E+00	9.24E+00	6.62E+00	7.51E+00
f <sub>19</sub>	Mean	1.39E+02	2.07E+03	4.51E+02	5.79E+03	3.01E+01	2.37E+01	4.72E+01	6.63E+01	1.54E+01	1.35E+01	1.58E+01	1.39E+01
	SD	4.79E+01	4.55E+03	2.23E+03	7.39E+03	1.51E+01	6.65E+00	2.33E+01	3.34E+01	4.11E+00	3.31E+00	3.67E+00	3.73E+00
f <sub>20</sub>	Mean	5.19E+02	5.79E+02	5.50E+02	5.25E+02	2.36E+02	3.33E+02	1.60E+02	1.95E+02	1.77E+02	1.71E+02	1.37E+02	1.43E+02
	SD	1.36E+02	1.21E+02	1.18E+02	1.27E+02	1.43E+02	2.24E+02	1.26E+02	1.60E+02	1.28E+02	1.53E+02	1.10E+02	1.41E+02
f <sub>21</sub>	Mean	2.53E+02	2.67E+02	2.54E+02	2.66E+02	2.33E+02	2.42E+02	2.27E+02	2.37E+02	2.28E+02	2.33E+02	2.24E+02	2.31E+02
	SD	8.10E+00	9.91E+00	8.06E+00	1.03E+01	5.14E+00	1.13E+01	3.97E+00	8.00E+00	4.19E+00	7.16E+00	2.72E+00	6.32E+00
f <sub>22</sub>	Mean	3.60E+03	4.04E+03	2.65E+03	4.65E+03	3.30E+03	3.96E+03	3.31E+03	3.84E+03	2.74E+03	3.11E+03	2.79E+03	4.07E+03
	SD	1.66E+03	2.26E+03	2.05E+03	1.81E+03	1.83E+03	2.54E+03	1.56E+03	2.54E+03	1.98E+03	2.18E+03	1.38E+03	1.54E+03
f <sub>23</sub>	Mean	4.78E+02	4.84E+02	4.79E+02	4.89E+02	4.50E+02	4.73E+02	4.48E+02	4.65E+02	4.43E+02	4.61E+02	4.44E+02	4.57E+02
	SD	1.04E+01	1.24E+01	1.16E+01	1.23E+01	8.78E+00	1.41E+01	9.53E+00	1.39E+01	8.28E+00	1.35E+01	8.00E+00	1.05E+01
f <sub>24</sub>	Mean	5.41E+02	5.35E+02	5.41E+02	5.39E+02	5.25E+02	5.35E+02	5.20E+02	5.31E+02	5.19E+02	5.27E+02	5.18E+02	5.30E+02
	SD	1.07E+01	1.26E+01	8.99E+00	1.33E+01	6.12E+00	9.36E+00	6.63E+00	1.02E+01	5.54E+00	9.25E+00	5.48E+00	8.19E+00
f <sub>25</sub>	Mean	5.43E+02	5.39E+02	5.23E+02	5.30E+02	5.43E+02	5.44E+02	5.41E+02	5.33E+02	5.34E+02	5.40E+02	5.28E+02	5.21E+02
	SD	3.14E+01	3.44E+01	3.34E+01	3.66E+01	3.55E+01	3.36E+01	2.83E+01	3.08E+01	3.39E+01	2.87E+01	3.29E+01	3.98E+01
f <sub>26</sub>	Mean	1.64E+03	1.62E+03	1.63E+03	1.64E+03	1.50E+03	1.56E+03	1.36E+03	1.47E+03	1.32E+03	1.40E+03	1.27E+03	1.37E+03
	SD	8.39E+01	1.23E+02	9.74E+01	1.18E+02	1.19E+02	1.18E+02	8.52E+01	1.37E+02	1.01E+02	1.11E+02	1.05E+02	8.82E+01
f <sub>27</sub>	Mean	5.71E+02	5.45E+02	5.57E+02	5.32E+02	5.73E+02	5.73E+02	5.46E+02	5.46E+02	5.20E+02	5.14E+02	5.18E+02	5.13E+02
	SD	3.33E+01	1.92E+01	3.03E+01	1.26E+01	4.22E+01	2.85E+01	2.39E+01	2.72E+01	1.69E+01	1.12E+01	1.40E+01	8.70E+00
f <sub>28</sub>	Mean	4.91E+02	4.93E+02	4.92E+02	4.90E+02	4.98E+02	4.95E+02	4.95E+02	4.96E+02	4.95E+02	4.89E+02	4.90E+02	4.85E+02
	SD	2.09E+01	1.89E+01	2.12E+01	2.31E+01	3.11E+01	2.21E+01	1.73E+01	1.94E+01	1.85E+01	2.35E+01	2.19E+01	2.46E+01
f <sub>29</sub>	Mean	4.88E+02	4.63E+02	4.75E+02	4.82E+02	4.08E+02	4.73E+02	4.12E+02	4.46E+02	3.65E+02	3.53E+02	3.62E+02	3.45E+02
	SD	8.05E+01	6.24E+01	8.28E+01	8.18E+01	3.76E+01	1.09E+02	5.39E+01	9.95E+01	3.46E+01	4.68E+01	3.98E+01	4.05E+01
f <sub>30</sub>	Mean	6.49E+05	6.36E+05	6.76E+05	6.60E+05	6.12E+05	5.97E+05	6.19E+05	6.21E+05	5.92E+05	5.94E+05	6.05E+05	5.98E+05
	SD	7.64E+04	5.54E+04	8.69E+04	8.07E+04	4.42E+04	2.58E+04	4.32E+04	4.36E+04	2.11E+04	2.24E+04	2.82E+04	2.73E+04
Nob		2	1	3	2	0	0	3	2	4	6	13	9

TABLE 15. Results of the multi-problem Wilcoxon's test between ADEwSE-S3 and JADE, JADEsort and ADEwSE variants

Algorithm at 50D	R <sup>+</sup>	R <sup>-</sup>	p-value	$\alpha=0.05$	$\alpha=0.1$
JADE-S1	401.0	5.0	6.0E-6	Yes	Yes
JADE-S2	405.0	1.0	4.0E-6	Yes	Yes
JADE-S3	381.0	25.0	4.8E-5	Yes	Yes
JADE-S4	430.5	4.5	4.0E-6	Yes	Yes
JADEsort-S1	422.0	13.0	9.0E-6	Yes	Yes
JADEsort-S2	404.0	31.0	5.3E-5	Yes	Yes
JADEsort-S3	419.5	15.5	1.2E-5	Yes	Yes
JADEsort-S4	433.5	1.5	3.0E-6	Yes	Yes
ADEwSE-S1	314.5	120.5	3.3E-2	Yes	Yes
ADEwSE-S2	272.5	162.5	2.		

**F. BENEFIT OF THE PROPOSED MODIFICATIONS TO THE PERFORMANCE OF ADEWSE**

In this subsection, to comprehensively investigate the contributions of the proposed modifications, five variants of

ADEwSE are constructed. The compared algorithms are list as follows.

- (1) Version 1 to study the effectiveness of the crossover matrix sorting scheme, the crossover matrix sorting strategy is not adopted in ADEwSE. This version denotes the ADEwSE-1.

**TABLE 18. Experimental results of ADEwSE, ADEwSE-1, ADEwSE-2, ADEwSE-3, ADEwSE-4, and ADEwSE-5 on 50D**

		ADeWSE-1	ADeWSE-2	ADeWSE-3	ADeWSE-4	ADeWSE-5	ADeWSE
$f_1$	Mean (SD)	2.01E-03(3.04E-03)	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>
$f_2$	Mean (SD)	6.08E+02(1.31E+03)	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>
$f_3$	Mean (SD)	5.40E+01(4.62E+01)	4.68E+01(4.80E+01)	<b>4.08E+01(4.97E+01)</b>	5.04E+01(4.66E+01)	4.77E+01(4.73E+01)	5.03E+01(4.42E+01)
$f_4$	Mean (SD)	5.59E+01(9.08E+00)	2.48E+01(4.79E+00)	2.62E+01(4.68E+00)	2.54E+01(4.27E+00)	3.10E+01(6.50E+00)	<b>2.47E+01(3.95E+00)</b>
$f_5$	Mean (SD)	5.88E-04(1.57E-04)	2.17E-03(4.83E-03)	1.05E-02(9.36E-03)	1.41E-04(9.85E-04)	9.73E-05(3.11E-04)	<b>2.62E-06(3.48E-06)</b>
$f_6$	Mean (SD)	9.95E+01(8.22E+00)	6.93E+01(3.12E+00)	7.16E+01(4.60E+00)	7.02E+01(3.47E+00)	7.89E+01(5.98E+00)	<b>6.89E+01(3.26E+00)</b>
$f_7$	Mean (SD)	5.66E+01(8.45E+00)	<b>2.41E+01(4.73E+00)</b>	2.56E+01(4.32E+00)	2.63E+01(5.35E+00)	3.08E+01(5.71E+00)	2.51E+01(4.27E+00)
$f_8$	Mean (SD)	<b>2.09E-07(9.47E-07)</b>	1.36E+00(1.63E+00)	7.51E+00(5.49E+00)	1.76E-01(3.98E-01)	9.03E-01(1.72E+00)	9.25E-02(1.94E-01)
$f_{10}$	Mean (SD)	3.08E+03(5.65E+02)	<b>2.75E+03(4.43E+02)</b>	2.86E+03(6.52E+02)	3.39E+03(2.96E+02)	2.77E+03(8.50E+02)	2.78E+03(6.82E+02)
$f_{11}$	Mean (SD)	7.03E+01(2.10E+01)	6.99E+01(1.71E+01)	1.05E+02(2.36E+01)	4.64E+01(1.27E+01)	6.37E+01(1.86E+01)	<b>4.49E+01(1.32E+01)</b>
$f_{12}$	Mean (SD)	3.30E+03(3.75E+03)	3.67E+03(2.59E+03)	6.40E+03(5.16E+03)	<b>2.66E+03(3.98E+03)</b>	5.70E+03(5.96E+03)	2.89E+03(2.35E+03)
$f_{13}$	Mean (SD)	1.28E+02(8.65E+01)	1.14E+02(6.06E+01)	1.20E+02(4.81E+01)	8.59E+01(3.71E+01)	8.88E+01(8.46E+01)	<b>6.13E+01(2.89E+01)</b>
$f_{14}$	Mean (SD)	3.69E+01(8.92E+00)	3.67E+01(5.64E+00)	5.57E+01(1.43E+01)	3.01E+01(4.54E+00)	3.67E+01(7.09E+00)	<b>2.91E+01(7.03E+00)</b>
$f_{15}$	Mean (SD)	6.15E+01(2.11E+01)	5.91E+01(2.02E+01)	1.01E+02(5.69E+01)	3.59E+01(8.59E+00)	5.36E+01(2.44E+01)	<b>3.31E+01(5.99E+00)</b>
$f_{16}$	Mean (SD)	8.22E+02(1.78E+02)	4.53E+02(1.50E+02)	4.84E+02(1.65E+02)	5.78E+02(1.57E+02)	<b>4.42E+02(1.58E+02)</b>	4.44E+02(1.72E+02)
$f_{17}$	Mean (SD)	5.03E+02(1.24E+02)	3.57E+02(1.13E+02)	3.60E+02(1.49E+02)	4.21E+02(1.25E+02)	3.53E+02(1.74E+02)	<b>3.27E+02(1.12E+02)</b>
$f_{18}$	Mean (SD)	5.82E+01(3.33E+01)	5.89E+01(3.05E+01)	7.22E+01(4.37E+01)	3.37E+01(7.96E+00)	4.49E+01(1.74E+01)	<b>3.30E+01(6.62E+00)</b>
$f_{19}$	Mean (SD)	3.32E+01(8.26E+00)	3.24E+01(9.50E+00)	6.20E+01(2.56E+01)	1.68E+01(3.79E+00)	2.99E+01(1.28E+01)	<b>1.58E+01(3.67E+00)</b>
$f_{20}$	Mean (SD)	2.77E+02(1.53E+02)	1.89E+02(1.25E+02)	1.50E+02(1.38E+02)	1.42E+02(1.02E+02)	2.32E+02(2.00E+02)	<b>1.37E+02(1.10E+02)</b>
$f_{21}$	Mean (SD)	2.57E+02(9.10E+00)	<b>2.24E+02(3.91E+00)</b>	2.26E+02(4.58E+00)	2.25E+02(3.47E+00)	2.32E+02(5.49E+00)	<b>2.24E+02(2.72E+00)</b>
$f_{22}$	Mean (SD)	<b>2.58E+03(1.63E+03)</b>	3.00E+03(1.15E+03)	3.08E+03(1.25E+03)	3.55E+03(1.35E+03)	3.13E+03(1.00E+03)	2.79E+03(1.38E+03)
$f_{23}$	Mean (SD)	4.82E+02(1.24E+01)	<b>4.42E+02(8.81E+00)</b>	4.43E+02(7.67E+00)	4.43E+02(7.04E+00)	4.57E+02(1.24E+01)	4.44E+02(8.00E+00)
$f_{24}$	Mean (SD)	5.44E+02(1.08E+01)	<b>5.18E+02(5.89E+00)</b>	<b>5.18E+02(6.23E+00)</b>	<b>5.18E+02(5.16E+00)</b>	5.31E+02(8.38E+00)	<b>5.18E+02(5.48E+00)</b>
$f_{25}$	Mean (SD)	5.25E+02(3.03E+01)	5.36E+02(3.14E+01)	5.35E+02(3.07E+01)	5.19E+02(3.49E+01)	<b>5.18E+02(3.58E+01)</b>	5.28E+02(3.29E+01)
$f_{26}$	Mean (SD)	1.68E+03(1.11E+02)	1.33E+03(8.75E+01)	1.36E+03(1.19E+02)	1.29E+03(7.04E+01)	1.49E+03(1.07E+02)	<b>1.27E+03(1.05E+02)</b>
$f_{27}$	Mean (SD)	5.20E+02(1.06E+01)	5.26E+02(1.75E+01)	5.46E+02(2.12E+01)	5.20E+02(1.00E+01)	5.26E+02(1.71E+01)	<b>5.18E+02(1.40E+01)</b>
$f_{28}$	Mean (SD)	4.98E+02(1.58E+01)	4.98E+02(1.58E+01)	4.97E+02(1.61E+01)	4.96E+02(1.90E+01)	4.91E+02(3.37E+01)	<b>4.90E+02(2.19E+01)</b>
$f_{29}$	Mean (SD)	4.46E+02(6.58E+01)	<b>3.53E+02(3.35E+01)</b>	3.66E+02(3.84E+01)	4.02E+02(5.77E+01)	4.03E+02(7.57E+01)	3.62E+02(3.98E+01)
$f_{30}$	Mean (SD)	<b>6.00E+05(3.36E+04)</b>	6.19E+05(4.60E+04)	6.33E+05(5.65E+04)	6.10E+05(4.00E+04)	6.31E+05(6.13E+04)	6.05E+05(2.82E+04)
	Nob	3	7	4	4	4	18

**TABLE 19. Experimental results of ADEwSE and different T values on 50D**

		T=50	T=100	T=170	T=300	T=500	ADeWSE
$f_1$	Mean (SD)	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>
$f_2$	Mean (SD)	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>	<b>0.00E+00(0.00E+00)</b>
$f_3$	Mean (SD)	4.76E+01(4.36E+01)	5.09E+01(4.68E+01)	4.45E+01(4.90E+01)	5.40E+01(4.75E+01)	<b>4.19E+01(4.27E+01)</b>	5.03E+01(4.42E+01)
$f_4$	Mean (SD)	2.80E+01(5.19E+00)	2.47E+01(4.64E+00)	<b>2.35E+01(3.89E+00)</b>	2.54E+01(4.14E+00)	2.51E+01(3.75E+00)	2.47E+01(3.95E+00)
$f_5$	Mean (SD)	5.33E-06(7.48E-06)	4.04E-06(5.01E-06)	3.58E-06(7.84E-06)	4.89E-06(7.28E-06)	5.38E-06(7.39E-06)	<b>2.62E-06(3.48E-06)</b>
$f_6$	Mean (SD)	7.17E+01(5.23E+00)	6.98E+01(3.94E+00)	7.05E+01(3.63E+00)	7.00E+01(2.77E+00)	7.09E+01(3.16E+00)	<b>6.89E+01(3.26E+00)</b>
$f_7$	Mean (SD)	2.51E+01(4.54E+00)	2.55E+01(4.85E+00)	2.44E+01(4.60E+00)	<b>2.43E+01(3.51E+00)</b>	2.57E+01(4.41E+00)	2.51E+01(4.27E+00)
$f_8$	Mean (SD)	1.35E+01(3.06E-01)	9.41E-02(2.03E-01)	2.03E-01(4.55E-01)	1.15E-01(2.25E-01)	<b>6.92E-02(1.46E-01)</b>	9.25E-02(1.94E-01)
$f_{10}$	Mean (SD)	4.29E+03(1.03E+03)	2.89E+03(9.73E+02)	<b>2.70E+03(8.89E+02)</b>	3.06E+03(3.26E+02)	3.27E+03(4.70E+02)	2.78E+03(6.82E+02)
$f_{11}$	Mean (SD)	4.80E+01(9.60E+00)	4.52E+01(9.83E+00)	<b>4.25E+01(1.27E+01)</b>	4.30E+01(1.12E+01)	4.67E+01(1.18E+01)	4.49E+01(1.32E+01)
$f_{12}$	Mean (SD)	3.17E+03(2.55E+03)	4.29E+03(6.41E+03)	3.85E+03(3.42E+03)	3.17E+03(2.57E+03)	3.34E+03(2.91E+03)	<b>2.89E+03(2.35E+03)</b>
$f_{13}$	Mean (SD)	5.44E+01(2.36E+01)	5.24E+01(2.63E+01)	<b>4.79E+01(2.83E+01)</b>	6.92E+01(3.59E+01)	8.02E+01(4.80E+01)	6.13E+01(2.89E+01)
$f_{14}$	Mean (SD)	3.03E+01(5.20E+00)	3.05E+01(5.54E+00)	<b>2.77E+01(8.60E+00)</b>	3.19E+01(3.16E+00)	2.93E+01(5.36E+00)	2.91E+01(7.03E+00)
$f_{15}$	Mean (SD)	3.52E+01(7.90E+00)	3.43E+01(6.13E+00)	3.50E+01(8.51E+00)	3.56E+01(8.66E+00)	3.37E+01(6.99E+00)	<b>3.31E+01(5.99E+00)</b>
$f_{16}$	Mean (SD)	4.66E+02(2.45E+02)	4.65E+02(1.92E+02)	<b>4.43E+02(1.57E+02)</b>	4.81E+02(1.85E+02)	5.09E+02(1.64E+02)	4.44E+02(1.72E+02)
$f_{17}$	Mean (SD)	3.65E+02(1.44E+02)	<b>3.20E+02(1.41E+02)</b>	3.47E+02(1.51E+02)	3.33E+02(1.29E+02)	3.85E+02(9.53E+01)	3.27E+02(1.12E+02)
$f_{18}$	Mean (SD)	3.51E+01(1.41E+01)	3.42E+01(8.62E+00)	<b>3.24E+01(6.63E+00)</b>	3.39E+01(7.53E+00)	3.36E+01(6.99E+00)	3.30E+01(6.62E+00)
$f_{19}$	Mean (SD)	<b>1.50E+01(3.31E+00)</b>	1.56E+01(4.03E+00)	1.58E+01(3.21E+00)	1.55E+01(3.28E+00)	1.55E+01(3.67E+00)	1.58E+01(3.67E+00)
$f_{20}$	Mean (SD)	1.92E+02(1.33E+02)	1.65E+02(1.13E+02)	1.54E+02(1.16E+02)	1.81E+02(1.15E+02)	1.43E+02(1.00E+02)	<b>1.37E+02(1.10E+02)</b>
$f_{21}$	Mean (SD)	2.26E+02(4.97E+00)	2.25E+02(3.65E+00)	<b>2.23E+02(3.36E+00)</b>	<b>2.23E+02(3.66E+00)</b>	2.25E+02(4.17E+00)	2.24E+02(2.72E+00)
$f_{22}$	Mean (SD)	3.86E+03(1.89E+03)	2.94E+03(1.69E+03)	3.09E+03(1.17E+03)	3.39E+03(1.04E+03)	3.45E+03(1.40E+03)	<b>2.79E+03(1.38E+03)</b>
$f_{23}$	Mean (SD)	4.48E+02(1.12E+01)	4.42E+02(8.87E+00)	<b>4.41E+02(5.73E+00)</b>	4.42E+02(7.90E+00)	4.43E+02(7.94E+00)	4.44E+02(8.00E+00)
$f_{24}$	Mean (SD)	5.23E+02(7.21E+00)	5.19E+02(6.16E+00)	<b>5.18E+02(5.70E+00)</b>	<b>5.18E+02(5.84E+00)</b>	<b>5.18E+02(4.90E+00)</b>	<b>5.18E+02(5.48E+00)</b>
$f_{25}$	Mean (SD)	5.34E+02(2.94E+01)	5.38E+02(3.11E+01)	5.35E+02(2.94E+01)	<b>5.28E+02(3.55E+01)</b>	5.31E+02(2.98E+01)	<b>5.28E+02(3.29E+01)</b>
$f_{26}$	Mean (SD)	1.33E+03(9.21E+01)	1.30E+03(9.28E+01)	1.28E+03(8.77E+01)	1.29E+03(9.96E+01)	<b>1.27E+03(7.48E+01)</b>	<b>1.27E+03(1.05E+02)</b>
$f_{27}$	Mean (SD)	5.17E+02(1.14E+01)	5.20E+02(1.36E+01)	<b>5.16E+02(1.14E+01)</b>	5.19E+02(1.07E+01)	5.18E+02(1.01E+01)	5.18E+02(1.40E+01)
$f_{28}$	Mean (SD)	4.99E+02(1.79E+01)	4.91E+02(2.22E+01)	4.99E+02(1.80E+01)	4.97E+02(1.97E+01)	4.98E+02(1.85E+01)	<b>4.90E+02(2.19E+01)</b>
$f_{29}$	Mean (SD)	3.54E+02(4.39E+01)	<b>3.39E+02(2.45E+01)</b>	3.50E+02(3.70E+01)	3.74E+02(4.01E+01)	3.81E+02(3.40E+01)	3.62E+02(3.98E+01)
$f_{30}$	Mean (SD)	6.08E+05(2.89E+04)	6.10E+05(3.25E+04)	6.11E+05(3.81E+04)	6.03E+05(2.89E+04)	6.20E+05(4.32E+04)	<b>6.05E+05(2.82E+04)</b>
	Nob	3	4	13	6	6	13

**TABLE 20. Average ranks of ADEwSE and different T values variants**

Algorithm	T=50	T=100	T=170	T=300	T=500	ADeWSE
Mean ranking	5.38	4.34	3.30	4.27	4.64	2.93
Rank	7	5	3	4	6	1

- (2) Version 2 to study the effectiveness of the opposition learning of crossover rate (OLCR), all modifications are kept

unchanged except the OLCR scheme. This version denotes the ADEwSE-2.

- (3) Version 3 to study the effectiveness of successful experience, the “current-to-pbest” mutation strategy is adopted and combined with other modifications. This version denotes the ADEwSE-3.

- (4) Version 4 to study the effectiveness of disturbance-based crossover (DX), the DX scheme is not used to enhance the exploitation ability. This version denotes the ADEwSE-4.
- (5) Version 5 to study the effectiveness of the adaptive  $p$  setting, the constant  $p$  value  $p = 0.05$  is used. This version denotes the ADEwSE-5.

The overall comparison results of the ADEwSE against its five versions (ADEwSE-1, ADEwSE-2, ADEwSE-3, ADEwSE-4, and ADEwSE-5) are summarized in Table 18. From Table 18, it is clear that the ADEwSE achieves the best results in 18 functions while ADEwSE-1, ADEwSE-2, ADEwSE-3, ADEwSE-4, and ADEwSE-5 achieve that in 3, 7, 4, 4 and 4, respectively. Furthermore, according to the best

**TABLE 21. Experimental results of LADEwSE, ADEwSE, and six powerful DE variants on 30D**

	LSHADE-EpSin	ELSHADE-SPACMA	LSHADESPACMA	EBWO	EBLSHADE	JSO	ADEwSE	LADEwSE
	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD
$f_1$	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/NA	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00
$f_2$	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/NA	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00
$f_3$	4.23E+01/3.07E+00	5.86E+01/0.00E+00	0.00E+00/0.00E+00	5.65E+01/1.11E+01	5.87E+01/NA	5.87E+01/7.78E-01	4.01E+01/2.84E+01	5.88E+01/1.09E+00
$f_4$	1.23E+01/2.34E+00	1.86E+01/8.04E+00	3.69E+00/2.48E+00	2.78E+00/1.74E+00	6.51E+00/NA	8.56E+00/2.10E+00	9.78E+00/1.90E+00	6.51E+00/1.35E+00
$f_5$	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	2.14E-08/NA	0.00E+00/0.00E+00	4.53E-08/2.81E-07	0.00E+00/0.00E+00
$f_6$	4.33E+01/2.17E+00	3.89E+01/3.43E+00	3.38E+01/8.24E-01	3.35E+01/8.37E-01	3.68E+01/NA	3.89E+01/1.46E+00	3.83E+01/1.34E+00	3.65E+01/1.04E+00
$f_7$	1.29E+01/2.86E+00	1.61E+01/7.46E+00	3.59E+00/1.62E+00	2.02E+00/1.32E+00	6.81E+00/NA	9.09E+00/1.84E+00	9.44E+00/1.95E+00	6.59E+00/1.51E+00
$f_8$	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/NA	0.00E+00/0.00E+00	8.91E-03/6.36E-02	0.00E+00/0.00E+00
$f_9$	1.39E+03/2.10E+02	1.70E+03/4.06E+02	1.44E+03/3.18E+02	1.41E+03/2.15E+02	1.37E+03/NA	1.53E+03/2.77E+02	1.63E+03/3.58E+02	1.44E+03/2.67E+02
$f_{10}$	1.35E+01/1.94E+01	7.80E+00/1.42E+01	3.79E+00/2.49E+00	4.49E+00/8.77E+00	2.07E+01/NA	3.04E+00/2.65E+00	1.16E+01/1.73E+01	9.74E+00/1.89E+01
$f_{11}$	3.72E+02/2.01E+02	2.47E+02/1.28E+02	5.54E+02/2.53E+02	4.63E+02/2.63E+02	7.94E+02/NA	1.70E+02/1.02E+02	3.22E+02/1.77E+02	1.02E+02/1.02E+02
$f_{12}$	1.73E+01/1.02E+01	1.57E+01/5.03E+00	1.52E+01/4.71E+00	1.49E+01/6.25E+00	1.41E+01/NA	1.48E+01/4.83E+00	1.52E+01/6.51E+00	1.90E+01/3.28E+00
$f_{13}$	2.16E+01/2.26E+00	2.37E+01/5.25E+00	2.28E+01/1.78E+00	2.19E+01/3.84E+00	1.93E+01/NA	2.18E+01/1.25E+00	1.39E+01/1.00E+01	1.47E+01/1.00E+01
$f_{14}$	3.24E+00/1.98E+00	1.86E+00/1.29E+00	5.37E+00/2.80E+00	3.69E+00/2.15E+00	2.41E+00/NA	1.09E+00/6.91E-01	2.33E+00/1.26E+00	1.61E+00/1.01E+00
$f_{15}$	2.29E+01/3.07E+01	6.68E+01/8.35E+01	3.77E+01/5.54E+01	4.26E+01/5.69E+01	3.89E+01/NA	7.89E+01/8.48E+01	1.50E+02/1.20E+02	4.50E+01/6.63E+01
$f_{16}$	2.86E+01/5.56E+00	2.97E+01/6.76E+00	3.17E+01/8.68E+00	2.98E+01/7.50E+00	3.27E+01/NA	3.29E+01/8.08E+00	3.46E+01/1.04E+01	2.87E+01/1.00E+01
$f_{17}$	2.11E+01/7.52E-01	2.09E+01/3.03E+00	2.34E+01/2.00E+00	2.21E+01/1.00E+00	2.14E+01/NA	2.04E+01/2.87E+00	2.09E+01/7.23E-01	2.02E+01/2.79E+00
$f_{18}$	5.83E+00/1.92E+00	4.61E+00/1.35E+00	9.45E+00/2.01E+00	8.04E+00/2.28E+00	4.83E+00/NA	4.50E+00/1.73E+00	2.99E+00/1.17E+00	4.21E+00/1.40E+00
$f_{19}$	3.03E+01/7.35E+00	2.73E+01/4.56E+00	8.16E+01/5.86E+01	3.57E+01/7.50E+00	3.05E+01/NA	2.94E+01/5.85E+00	2.31E+01/9.91E+00	2.22E+01/8.59E+00
$f_{20}$	2.12E+02/2.56E+00	2.22E+02/6.64E+00	2.09E+02/4.19E+00	1.99E+02/2.02E+01	2.07E+02/NA	2.09E+02/1.96E+00	2.10E+02/2.46E+00	2.07E+02/1.59E+00
$f_{21}$	1.00E+02/0.00E+00	1.00E+02/0.00E+00	1.00E+02/0.00E+00	1.00E+02/0.00E+00	1.00E+02/NA	1.00E+02/0.00E+00	1.00E+02/0.00E+00	1.00E+02/0.00E+00
$f_{22}$	3.56E+02/3.73E+00	3.69E+02/1.05E+01	3.57E+02/4.07E+00	3.51E+02/3.51E+00	3.49E+02/NA	3.51E+02/3.30E+00	3.53E+02/4.88E+00	3.50E+02/3.22E+00
$f_{23}$	4.28E+02/2.95E+00	4.41E+02/2.38E+00	4.29E+02/2.38E+00	4.18E+02/3.55E+01	4.25E+02/NA	4.26E+02/2.47E+00	4.28E+02/2.94E+00	4.26E+02/1.99E+00
$f_{24}$	3.87E+02/8.90E-03	3.87E+02/9.60E-03	3.78E+02/1.22E-02	3.87E+02/7.56E-01	3.87E+02/NA	3.87E+02/7.68E-03	3.87E+02/2.07E-02	3.87E+02/1.81E-02
$f_{25}$	9.49E+01/2.60E+00	1.08E+03/8.68E+01	9.43E+02/3.56E+01	9.01E+02/NA	9.01E+02/NA	9.20E+02/4.30E+01	9.57E+02/6.48E+01	9.06E+02/4.41E+01
$f_{26}$	5.04E+02/6.70E+00	4.99E+02/6.15E+00	4.97E+02/1.41E+01	5.02E+02/4.03E+00	5.01E+02/NA	4.97E+02/7.00E+00	5.01E+02/7.72E+00	4.98E+02/5.88E+00
$f_{27}$	3.15E+02/3.86E+01	3.02E+02/1.60E+01	3.34E+02/5.12E+01	3.08E+02/2.88E+01	3.22E+02/NA	3.09E+02/3.03E+01	3.21E+02/4.28E+01	3.06E+02/2.63E+01
$f_{28}$	4.35E+02/7.36E+00	4.33E+02/1.56E+01	3.95E+02/4.06E+01	4.33E+02/1.13E+01	4.32E+02/NA	4.34E+02/1.36E+01	4.28E+02/1.43E+01	4.30E+02/8.04E+00
$f_{29}$	1.98E+03/4.17E+01	1.98E+03/3.34E+01	4.01E+02/9.06E+01	1.99E+03/4.21E+01	1.99E+03/NA	1.97E+03/1.90E+01	1.97E+03/2.60E+01	1.97E+03/9.15E+00
$f_{30}$	9	7	9	11	8	10	9	10

**TABLE 22. Experimental results of LADEwSE, ADEwSE, and six powerful DE variants on 50D**

	LSHADE-EpSin	ELSHADE-SPACMA	LSHADESPACMA	EBWO	EBLSHADE	JSO	ADEwSE	LADEwSE
	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD
$f_1$	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/NA	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00
$f_2$	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/NA	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00
$f_3$	5.14E+01/4.43E+01	4.36E+01/3.62E+01	5.68E-01/1.45E+00	4.29E+01/3.32E+01	7.91E+01/NA	5.62E+01/4.88E+01	5.03E+01/4.42E+01	4.96E+01/3.60E+01
$f_4$	2.52E+01/6.44E+00	1.39E+01/5.55E+00	6.22E+00/2.10E+00	7.58E+00/2.42E+00	1.27E+01/NA	1.64E+01/3.46E+00	2.47E+01/3.95E+00	1.22E+01/2.10E+00
$f_5$	9.16E-07/1.08E-06	0.00E+00/0.00E+00	0.00E+00/0.00E+00	8.54E-08/1.14E-07	2.11E-05/NA	1.09E-06/2.63E-06	2.62E-06/3.48E-06	2.11E-08/6.17E-08
$f_6$	7.66E+01/6.06E+00	6.15E+01/3.86E+00	5.71E+01/1.07E+00	5.79E+01/1.53E+00	6.28E+01/NA	6.65E+01/3.47E+00	6.89E+01/3.26E+00	6.12E+01/1.52E+00
$f_7$	2.63E+01/6.59E+00	1.79E+01/7.47E+00	5.85E+00/2.29E+00	7.91E+00/2.47E+00	1.24E+01/NA	1.70E+01/3.14E+00	2.51E+01/4.27E+00	1.24E+01/2.10E+00
$f_8$	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/NA	0.00E+00/0.00E+00	9.25E-02/1.94E-01	0.00E+00/0.00E+00
$f_9$	3.20E+03/3.40E+02	3.69E+03/6.07E+02	3.61E+03/6.70E+02	3.11E+03/4.01E+02	3.03E+03/NA	3.14E+03/3.67E+02	2.78E+03/6.82E+02	2.94E+03/3.91E+02
$f_{10}$	2.14E+01/2.09E+00	2.62E+01/3.76E+00	9.17E+00/3.37E+00	2.64E+01/3.36E+00	4.40E+01/NA	2.79E+01/3.33E+00	4.49E+01/1.32E+01	2.57E+01/2.72E+00
$f_{11}$	1.48E+03/3.65E+02	1.36E+03/3.42E+02	1.58E+03/4.43E+02	1.94E+03/8.34E+02	2.15E+03/NA	1.68E+03/5.23E+02	2.89E+03/2.35E+03	8.13E+02/2.54E+02
$f_{12}$	6.94E+01/3.45E+01	3.68E+01/1.72E+01	3.53E+01/1.52E+01	4.14E+01/2.48E+01	4.60E+01/NA	3.06E+01/2.12E+01	6.13E+01/2.89E+01	2.40E+01/1.88E+01
$f_{13}$	2.65E+01/2.49E+00	3.07E+01/3.95E+00	2.89E+01/3.92E+00	3.12E+01/3.52E+00	2.69E+01/NA	2.50E+01/1.87E+00	2.91E+01/7.03E+00	2.49E+01/1.57E+00
$f_{14}$	2.56E+01/4.06E+00	2.28E+01/2.20E+00	1.57E+01/6.71E+00	2.94E+01/5.20E+00	3.44E+01/NA	2.39E+01/2.49E+00	3.31E+01/5.99E+00	1.96E+01/1.15E+00
$f_{15}$	2.75E+02/9.97E+01	4.15E+02/1.77E+02	4.09E+02/1.68E+02	3.46E+02/1.46E+02	3.34E+02/NA	4.51E+02/1.38E+02	4.44E+02/1.72E+02	3.10E+02/1.23E+02
$f_{16}$	2.07E+02/7.31E+01	2.30E+02/9.68E+01	2.94E+02/1.19E+02	2.75E+02/8.63E+01	2.35E+02/NA	2.83E+02/8.61E+01	3.27E+02/1.12E+02	1.98E+02/7.72E+01
$f_{17}$	2.43E+01/2.12E+00	2.51E+01/2.56E+00	3.32E+01/7.23E+00	3.20E+01/5.99E+00	3.28E+01/NA	2.43E+01/2.02E+00	3.30E+01/6.62E+00	2.15E+01/8.00E-01
$f_{18}$	1.74E+01/2.47E+00	1.44E+01/2.31E+00	2.18E+01/4.83E+00	2.45E+01/3.94E+00	1.88E+01/NA	1.41E+01/2.26E+00	1.58E+01/3.67E+00	9.98E+00/2.18E+00
$f_{19}$	1.14E+02/3.55E+01	1.08E+02/7.31E+01	1.63E+02/1.08E+02	1.47E+02/7.44E+01	1.50E+02/NA	1.40E+02/7.74E+01	1.37E+02/1.10E+02	9.80E+01/7.44E+01
$f_{20}$	2.27E+02/7.06E+00	2.42E+02/9.52E+00	2.15E+02/9.22E+00	2.11E+02/4.06E+00	2.13E+02/NA	2.19E+02/3.77E+00	2.24E+02/2.72E+00	2.13E+02/2.16E+00
$f_{21}$	1.60E+03/1.67E+03	7.16E+02/1.44E+03	8.17E+02/1.49E+03	3.65E+02/9.24E+02	2.36E+03/NA	1.49E+03/1.75E+03	2.79E+03/1.38E+03	2.37E+03/1.59E+03
$f_{22}$	4.39E+02/6.90E+00	4.62E+02/1.39E+01	4.39E+02/7.25E+00	4.34E+02/8.16E+00	4.27E+02/NA	4.40E+02/6.24E+00	4.44E+02/8.00E+00	4.30E+02/3.67E+00
$f_{23}$	5.13E+02/5.59E+00	5.34E+02/9.14E+00	5.13E+02/5.80E+00	5.06E+02/3.85E+00	5.05E+02/NA	5.07E+02/4.13E+00	5.18E+02/5.48E+00	5.07E+02/2.71E+00
$f_{24}$	4.80E+02/1.08E+00	4.81E+02/2.80E+00	4.63E+02/8.49E+00	4.89E+02/2.47E+01	4.89E+02/NA	4.81E+02/2.80E+00	5.28E+02/3.29E+01	5.16E+02/3.79E+01
$f_{25}$	1.20E+03/1.19E+02	1.34E+03/1.38E+02	1.14E+03/6.99E+01	7.06E+02/4.06E+02	1.12E+03/NA	1.13E+03/5.62E+01	1.27E+03/1.05E+02	1.08E+03/5.32E+01
$f_{26}$	5.25E+02/9.21E+00	5.10E+02/9.52E+00	5.00E+02/8.98E-05	5.22E+02/7.75E+00	5.24E+02/NA	5.11E+02/1.11E+01	5.18E+02/1.40E+01	5.05E+02/7.14E+00
$f_{27}$	4.59E+02/1.19E+01	4.60E+02/6.84E+00	4.40E+02/8.51E+00	4.67E+02/1.79E+01	4.73E+02/NA	4.60E+02/6.84E+00	4.90E+02/2.19E+01	4.92E+02/2.29E+01
$f_{28}$	3.53E+02/9.78E+00	3.58E+02/1.78E+01	2.75E+02/5.95E+01	3.47E+02/1.97E+01	3.53E+02/NA	3.63E+02/1.32E+01	3.62E+02/3.98E+01	3.44E+02/1.32E+01
$f_{29}$	6.58E+0							

300, and 500) are compared with the standard ADEwSE (T=200). All other control parameter settings remain unchanged to allow a direct comparison. The experimental results are presented in Table 19. The average ranking values obtained by each T are displayed in Table 20.

Firstly, the number of the minimum results of each T listed at the bottom of Table 19 indicates that the standard ADEwSE

(T=200) achieves the best results in 13 test functions, while T=100, T=170, T=300, and T=500 obtain 3, 4, 13, 6, and 6 best results, respectively. Furthermore, it should be noted that T=170 has similar performance to the standard ADEwSE in terms of the best results number obtained by each T value. Besides, Table 19 shows that T with too small or too large values are unsuitable for ADEwSE.

**TABLE 23. Experimental results of LADEwSE, ADEwSE, and six powerful DE variants on 100D**

	LSHADE-EpSin	ELSHADE-SPACMA	LSHADESPACMA	EBWO	EBLSHADE	JSO	ADEwSE	LADEwSE
	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD
$f_1$	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/NA	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00
$f_2$	0.00E+00/0.00E+00	1.60E-07/4.00E-07	0.00E+00/0.00E+00	2.99E-07/7.04E-07	1.68E-06/NA	2.39E-06/2.73E-06	1.62E-05/5.21E-05	2.03E-06/1.60E-06
$f_3$	1.98E+02/8.30E+00	2.01E+02/8.67E+00	1.12E+02/9.24E+00	1.93E+02/3.09E+01	1.54E+02/NA	1.90E+02/2.89E+01	1.68E+01/4.69E+01	3.65E+01/5.69E+01
$f_4$	5.59E+01/9.91E+00	1.78E+01/3.85E+00	1.08E+01/2.58E+00	2.87E+01/5.28E+00	4.06E+01/NA	4.39E+01/5.61E+00	1.04E+02/1.43E+01	3.40E+01/4.12E+00
$f_5$	6.02E-05/2.18E-05	0.00E+00/0.00E+00	0.00E+00/0.00E+00	1.63E-05/7.13E-06	1.12E-02/NA	2.02E-04/6.20E-04	2.46E-02/2.16E-02	3.12E-08/4.56E-08
$f_6$	1.62E+02/7.91E+00	1.11E+02/1.48E+00	1.12E+02/1.56E+00	1.22E+02/4.87E+00	1.40E+02/NA	1.45E+02/6.70E+00	2.15E+02/1.83E+01	1.29E+02/3.97E+00
$f_7$	5.35E+01/5.39E+00	1.82E+01/3.02E+00	1.13E+01/2.94E+00	2.97E+01/7.48E+00	4.12E+01/NA	4.22E+01/5.52E+00	1.03E+02/1.59E+01	3.27E+01/3.87E+00
$f_8$	0.00E+00/0.00E+00	0.00E+00/0.00E+00	0.00E+00/0.00E+00	1.76E-03/1.25E-02	8.60E-01/NA	4.59E-02/1.15E-01	1.43E+02/1.47E+02	0.00E+00/0.00E+00
$f_9$	1.03E+04/5.21E+02	1.08E+04/9.53E+02	9.53E+03/9.00E+02	9.91E+03/1.91E+03	1.05E+04/NA	9.70E+03/6.82E+02	8.37E+03/1.09E+03	9.22E+03/5.25E+02
$f_{10}$	4.92E+01/3.02E+01	7.34E+01/4.30E+01	3.54E+01/1.31E+01	6.56E+01/2.00E+01	4.31E+02/NA	1.13E+02/4.32E+01	3.22E+02/9.36E+01	9.04E+01/4.48E+01
$f_{11}$	4.62E+02/6.48E+02	7.79E+02/3.85E+03	4.83E+03/8.05E+02	4.19E+03/7.89E+02	2.01E+04/NA	1.84E+04/8.35E+03	1.83E+04/8.54E+03	7.20E+03/3.99E+03
$f_{12}$	1.25E+02/3.65E+01	1.49E+02/3.83E+01	1.30E+02/4.05E+01	2.35E+02/8.84E+01	2.42E+02/NA	1.45E+02/3.80E+01	3.75E+02/5.43E+02	2.16E+02/3.40E+02
$f_{13}$	4.97E+01/8.17E+00	4.75E+01/5.69E+00	7.77E+01/1.38E+01	1.38E+02/2.96E+01	2.29E+02/NA	6.43E+01/1.09E+01	2.18E+02/4.73E+01	3.46E+01/3.65E+00
$f_{14}$	8.99E+01/2.83E+01	1.08E+02/4.34E+01	1.09E+02/3.35E+01	1.65E+02/3.87E+01	2.53E+02/NA	1.62E+02/3.81E+01	2.25E+02/8.03E+01	8.28E+01/2.77E+01
$f_{15}$	1.22E+03/2.36E+02	1.76E+03/4.88E+02	1.35E+03/4.00E+02	1.41E+03/3.76E+02	1.50E+03/NA	1.86E+03/3.49E+02	2.03E+03/3.83E+02	1.41E+03/3.51E+02
$f_{16}$	9.32E+02/1.74E+02	1.27E+03/3.45E+02	9.88E+02/3.36E+02	1.21E+03/2.57E+02	1.10E+04/NA	1.28E+03/2.38E+02	1.34E+03/3.29E+02	8.88E+02/2.17E+02
$f_{17}$	7.79E+01/1.99E+01	1.05E+02/2.56E+01	1.32E+02/2.43E+01	2.37E+02/5.94E+01	2.33E+02/NA	1.67E+02/3.65E+01	4.90E+02/3.41E+02	4.62E+01/8.94E+00
$f_{18}$	5.55E+01/6.05E+00	6.05E+01/7.55E+00	7.41E+01/1.21E+01	1.15E+02/1.88E+01	1.75E+02/NA	1.05E+02/2.01E+01	1.48E+02/3.91E+01	3.74E+01/3.98E+00
$f_{19}$	1.08E+03/2.16E+02	9.28E+02/2.54E+02	1.28E+03/3.59E+02	1.36E+03/3.09E+02	1.52E+03/NA	1.38E+03/2.43E+02	1.38E+03/3.16E+02	1.25E+03/2.16E+02
$f_{20}$	2.77E+02/6.94E+00	2.96E+02/1.65E+01	2.43E+02/1.06E+01	2.60E+02/1.06E+01	2.57E+02/NA	2.64E+02/6.43E+00	3.13E+02/1.35E+01	2.54E+02/3.81E+00
$f_{21}$	1.04E+04/5.30E+02	9.70E+03/1.20E+03	9.00E+03/9.34E+02	1.02E+04/2.70E+03	1.12E+04/NA	1.02E+04/2.18E+03	9.50E+03/1.07E+03	9.96E+03/7.30E+02
$f_{22}$	5.98E+02/7.69E+00	6.03E+02/2.19E+01	5.82E+02/7.68E+00	5.77E+02/1.31E+01	5.71E+02/NA	5.71E+02/1.07E+01	6.33E+02/3.25E+01	5.48E+02/1.12E+01
$f_{23}$	9.17E+02/1.34E+01	9.32E+02/1.90E+01	9.16E+02/1.63E+01	9.19E+02/1.32E+01	9.03E+02/NA	9.02E+02/7.89E+00	9.82E+02/2.18E+01	8.96E+02/5.58E+00
$f_{24}$	6.84E+02/4.34E+01	7.00E+02/3.99E+01	6.72E+02/4.30E+01	7.16E+02/3.71E+01	7.47E+02/NA	7.36E+02/3.53E+01	7.53E+02/5.14E+01	7.64E+02/2.67E+01
$f_{25}$	3.11E+03/1.22E+02	3.24E+03/2.19E+02	3.14E+03/7.12E+01	2.77E+03/1.08E+03	3.23E+03/NA	3.27E+03/8.02E+01	4.14E+03/2.83E+02	3.13E+03/7.34E+01
$f_{26}$	5.89E+02/1.31E+01	5.62E+02/1.75E+01	5.00E+02/1.34E+01	5.88E+02/1.53E+01	6.14E+02/NA	5.85E+02/2.17E+01	6.18E+02/2.75E+01	5.67E+02/1.42E+01
$f_{27}$	5.15E+02/2.20E+01	5.21E+02/2.38E+01	5.00E+02/1.07E+04	5.10E+02/6.01E+01	5.31E+02/NA	5.27E+02/2.73E+01	4.86E+02/7.92E+01	5.34E+02/3.29E+01
$f_{28}$	1.12E+03/1.49E+02	1.21E+03/1.98E+02	1.25E+03/2.69E+02	1.28E+03/2.42E+02	1.08E+03/NA	1.26E+03/1.91E+02	1.59E+03/3.13E+02	1.01E+03/1.31E+02
$f_{29}$	2.36E+03/1.44E+02	2.25E+03/1.11E+02	5.62E+02/6.28E+01	2.40E+03/1.51E+02	2.37E+03/NA	2.33E+03/1.19E+02	2.63E+03/3.93E+02	2.13E+03/7.50E+01
Nob	6	4	12	3	1	1	4	10

The reason is that a small T value forces DX to perform frequently, which in turn significantly deteriorates the exploitation ability and makes the algorithm stagnate further. On the contrary, a large T value will weaken the benefits of DX. In addition, the Friedman test results listed in Table 20 show that standard ADEwSE with T=200 achieves the first ranking. Finally, according to these experimental results and analysis, we can conclude that T=200 may be the most appropriate choice for the most of problems.

### H ENHANCING ADEwSE ALGORITHM BY USING POPULATION SIZE REDUCTION STRATEGY

To improve the performance of ADEwSE further, a linear population size reduction scheme is introduced. As mentioned before, the population size significantly impacts the convergence rate of the evolution algorithm. Large population size during the early stage of the optimization process is beneficial to strengthen the exploration ability, then reducing the population size is needed for refining the solution quality [66]. The framework to decrease the population size is presented as follows.

$$NP^{G+1} = \text{round}[NP_{min} + (NP_{min} - NP_{max}) \cdot (\text{currFE}/\text{maxFE})] \quad (33)$$

where  $NP^{G+1}$  denotes the new population size in the next generation;  $\text{currFE}$  is the current function evaluations (FE);  $\text{maxFE}$  is the maximum number of function evaluations. The improved ADEwSE is named as LADEwSE. From Eq. 33, it's

obvious that the population reduction strategy is identical with LSHADE [46]. In LSHADE, the  $NP_{max}$  is set to 18D, then the population size decreased from  $NP_{max}=18D$  to  $NP_{min}=4$ .

**TABLE 24. Results of the multi-problem Wilcoxon's test for LADEwSE versus ADEwSE and six powerful DE variants**

Algorithm at 30D	R <sup>+</sup>	R <sup>-</sup>	p-value	$\alpha=0.05$	$\alpha=0.1$
ADEwSE	321.5	84.5	6.0E-3	Yes	Yes
LSHADE-EpSin	327.5	107.5	2.0E-2	Yes	Yes
ELSHADE-SPACMA	371.5	63.5	7.9E-4	Yes	Yes
LSHADESPACMA	231.5	203.5	0.75	No	No
EBWO	206.0	229.0	1.00	No	No
EBLSHADE	270.5	135.5	0.12	No	No
JSO	338.0	97.0	8.5E-3	Yes	Yes
Algorithm at 50D	R <sup>+</sup>	R <sup>-</sup>	p-value	$\alpha=0.05$	$\alpha=0.1$
ADEwSE	433.5	1.5	3.0E-6	Yes	Yes
LSHADE-EpSin	307.5	98.5	1.5E-2	Yes	Yes
ELSHADESPACMA	319.5	86.5	7.0E-3	Yes	Yes
LSHADESPACMA	214.5	191.5	0.78	No	No
EBWO	260.5	145.5	0.18	No	No
EBLSHADE	331.0	75.0	3.2E-3	Yes	Yes
JSO	335.0	71.0	2.4E-3	Yes	Yes
Algorithm at 100D	R <sup>+</sup>	R <sup>-</sup>	p-value	$\alpha=0.05$	$\alpha=0.1$
ADEwSE	346.0	60.0	1.0E-3	Yes	Yes
LSHADE-EpSin	277.5	157.5	0.19	No	No
ELSHADE-SPACMA	290.5	144.5	0.11	No	No
LSHADESPACMA	198.0	237.0	1	No	No
EBWO	293.0	113.0	3.0E-2	Yes	Yes
EBLSHADE	390.5	15.5	1.8E-5	Yes	Yes
JSO	372.0	34.0	1.1E-4	Yes	Yes

However, the adaptive adjustment of top p% value, crossover matrix sorting, and opposition learning of CR are used to improve the exploration ability, thus, LADEwSE is assigned a smaller  $NP_{max}$  than LSHADE, namely,  $NP_{max}=10D$ . Note that the parameter adjustments of LADEwSE are the same as ADEwSE.

In this work, the LADEwSE is compared with ADEwSE and six powerful DE algorithms. They are LSHADE-EpSin [67], ELSHADESPACMA [68], LSHADESPACMA [69], EBWO [70], EBLSHADE [29], and JSO [23]. The results of 6 state-of-art algorithms are adopted from their original papers.

**TABLE 25. Average ranking values (ARV) of LADEwSE, ADEwSE and six powerful DE variants**

Rank	D=30		D=50		D=100	
	Algorithm	ARV	Algorithm	ARV	Algorithm	ARV
1	LADEwSE	3.41	LADEwSE	2.83	LSHADE SPACMA	2.64
2	EBWO	4.12	LSHADE SPACMA	3.47	LADEwSE	3.10
3	JSO	4.14	EBWO	4.12	LSHADE -EpSin	3.86
4	LSHADE SPACMA	4.41	JSO	4.47	ELSHADE -SPACMA	4.00
5	EBLSH ADE	4.52	ELSHADE- SPACMA	4.50	EBWO	4.72
6	ADEwSE	4.91	LSHADE- EpSin	4.83	JSO	5.26
7	LSHADE- EpSin	5.21	EBLSH ADE	4.97	EBLSH ADE	5.78
8	ELSHADE- SPACMA	5.28	ADEwSE	6.83	ADEwSE	6.64

**TABLE 26. Average ranks for all algorithms across all problems**

Ranks	Algorithm	30D	50D	100D	Mean ranking
1	LADEwSE	3.41	2.83	3.10	3.11
2	LSHADESPACMA	4.41	3.47	2.64	3.51
3	EBWO	4.12	4.12	4.72	4.32
4	ELSHADE- SPACMA	5.28	4.50	4.00	4.59
5	JSO	4.14	4.47	5.26	4.62
6	LSHADE-EpSin	5.21	4.83	3.86	4.63
7	EBLSHADE	4.52	4.97	5.78	5.09
8	ADEwSE	4.91	6.83	6.64	6.13

Firstly, the overall comparison results of the LADEwSE against ADEwSE and other six powerful algorithms on the benchmarks with 30D, 50D and 100D are provided in Tables 21, 22 and 23, respectively. It includes the obtained mean and the standard deviation of the function solutions over 51 runs for all 29 benchmark functions. NA in Tables 21, 22 and 23 indicates that the results are not available in the original papers. The best results are emphasized in bold for all problems. For the 30D problems, as shown in Table 21, LSHADE-EpSin, ELSHADESPACMA, LSHADESPACMA, EBWO, EBLSHADE, JSO, ADEwSE, LADEwSE can obtain the best results on 9, 7, 9, 11, 8, 10, 8 and 9 functions, respectively. Concerning the 50D problems, LADEwSE obtains 10 best results, and EBWO, EBLSHADE, JSO, ADEwSE obtain the best results on 4, 4, 13, 8, 4, 3, 3 functions, respectively. With regard to 100D problems, LSHADE-EpSin, ELSHADE-SPACMA, LSHADESPACMA, EBWO, EBLSHADE, JSO, ADEwSE, LADEwSE achieve the best results in 4, 4, 13, 8, 4, 3, 3, 10 cases, respectively. Based on the number of best results, one can conclude that LADEwSE and LSHADESPACMA perform well on all dimensions, especially for 50D and 100D. At D=30, EBWO gets the first rank according to the number of best results. Besides, Table 24 reports the results of multi-problem Wilcoxon's test between LADEwSE and other competitive algorithms for 30D, 50D and 100D problems. From Table 24, LADEwSE obtains higher  $R^+$  values than  $R^-$  values in most cases while slightly lower  $R^+$  values than  $R^-$  values in comparison with EBWO at 30D and LSHADESPACMA at 100D. However, the LADEwSE is

competitive with EBWO and LSHADESPACMA. According to the Wilcoxon's test at  $\alpha=0.05$  and 0.1, the significance difference can be found in 13 and 13 cases for all dimension problems, respectively. Furthermore, the results of Friedman test summarized in Table 25 clearly show that among all algorithms, LADEwSE is ranked the first for test problems with 30D and 50D, and obtains the second ranking in 100D problems.

In addition, the mean aggregated rank of all the eight algorithms is presented in Table 26 to give a comprehensive comparison for all dimensions. From Table 26, it can be clearly concluded that LADEwSE is the best followed by LSHADESPACMA as second best among all algorithms. Overall, all these results and discussion shows that LADEwSE is of better searching quality and robustness for solving unconstrained problems. In addition, in comparison with ADEwSE, the LADEwSE is significantly better than ADEwSE in 30D, 50D and 100D benchmark problems according to experimental and statistical results shown in Tables 21-26. It means that the linear reduction of population size is an efficient strategy to improve the searching ability.

Although LADEwSE has exhibited robust performance in most of the test problems, from the precision displayed in Tables 21-23, it fails to obtain higher solution precision than LSHADESPACMA in many cases. Besides, the initial population size is set based on the trial and error method, a comprehensive investigation needs to be considered in the future.

## VI. CONCLUSION

To strengthen the search ability of adaptive differential evolution, a modified mutation strategy by adding a successful experience vector with the "current-to-pbest/1" mutation strategy is proposed. The successful experience vector is calculated when a trial vector has better fitness than its associated parent. Furthermore, opposition learning of crossover rate, crossover matrix sorting scheme, and adaptive adjustment of top p% values are added into the proposed algorithm to balance the exploration and exploitation ability. In addition, a disturbance-based crossover, which is helpful for improving the exploitation ability of the algorithm as well as avoiding the stagnation, is presented. Extensive numerical experiments have been conducted. The results demonstrate that the proposed algorithm can improve the global search ability and the solution precision in comparison with advanced DE variants and "current-to-best/1" based adaptive DE algorithms. Besides, the validity of modifications, such as crossover matrix sorting, opposition learning of crossover rate, and adaptive top p% value, are experimentally studied. In addition, the effectiveness of DX in handling stagnation is confirmed by using different threshold T values.

Considering that the strategies in ADEwSE are beneficial to improve the performance of DE in most cases, an improved ADEwSE with a linear reduction of population size is investigated. The LADEwSE is more powerful, efficient, and

robust than ADEwSE. The performance of LADEwSE is confirmed by comparing it with 6 winner algorithms in CEC2017 competition that LADEwSE is statistically superior to and competitive with its competitor algorithms.

As a continuation of this research, we will focus on how to modify the ADEwSE algorithm for multi-objective and constrained optimization problems and practical engineering applications. Another possible direction is integrating the successful experience vector with other self-adaptive DE variants like jDE, SHADE, and many others.

## REFERENCES

- [1] E. G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.
- [2] C. Ryan, J. J. Collins, and M. O'Neill, "Grammatical Evolution: Evolving Programs for an Arbitrary Language," in *Genetic Programming, First European Workshop, EuroGP'98, Paris, France, Apr., 1998, Proceedings*, 1998.
- [3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw. (ICNN)*, vol. 4, Nov. 1995, pp. 1942–1948.
- [4] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proc. IEEE Swarm Intell. Symp. (SIS)*, Apr. 2007, pp. 120–127.
- [5] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [6] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm," *International Journal of Machine Learning and Cybernetics*, pp. 1-29, 2019.
- [7] Z. Meng, Y. Chen, and X. Li, "Enhancing Differential Evolution with Novel Parameter Control," *IEEE Access*, vol. PP, no. 99, pp. 1-1, 2020.
- [8] R. Storn, and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, Dec. 1997.
- [9] L. Lakshminarasimman, and S. Subramanian, "Applications of Differential Evolution in Power System Optimization," *Advances in Differential Evolution*. U. K. Chakraborty, ed., pp. 257-273, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
- [10] D. Zhao Yang, L. Miao, L. Zhe *et al.*, "A Differential Evolution Based Method for Power System Planning," pp. 2699-2706.
- [11] P. P. Biswas, P. N. Suganthan, R. Mallipeddi *et al.*, "Optimal power flow solutions using differential evolution algorithm integrated with effective constraint handling techniques," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 81-100, Sep. 2018.
- [12] R. P. Singh, D. Prasad, P. Kumar, and M. I. Khan, "Modified differential evolution algorithm for optimal power flow with FACTS devices," in *2017 International Conference on Smart grids, Power and Advanced Control Engineering (ICSPACE)*. Aug. 2017, pp. 48-52.
- [13] A. P. Piotrowski, "Differential Evolution algorithms applied to Neural Network training suffer from stagnation," *Applied Soft Computing*, vol. 21, pp. 382-406, Aug. 2014.
- [14] A. Vilorio, T. Crissien Borrero, J. Vargas Villa *et al.*, "Differential Evolution Clustering and Data Mining for Determining Learning Routes in Moodle," *Data Mining and Big Data*. Jul. 2019, pp. 170-178.
- [15] X. Li, and M. Yin, "Application of differential evolution algorithm on self-potential data," *PLoS one*, vol. 7, no. 12, pp. e51199, 2012.
- [16] S. Sarkar, G. R. Patra, and S. Das, "A Differential Evolution Based Approach for Multilevel Image Segmentation Using Minimum Cross Entropy Thresholding," in *Swarm, Evolutionary, and Memetic Computing*, 2011, pp. 51-58.
- [17] S. Suresh, and S. Lal, "Modified differential evolution algorithm for contrast and brightness enhancement of satellite images," *Applied Soft Computing*, vol. 61, pp. 622-641, Dec. 2017.
- [18] G. Wu, R. Mallipeddi, P. N. Suganthan *et al.*, "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences*, vol. 329, pp. 329-345, Sep. 2016.
- [19] J. Brest, S. Greiner, B. Boskovic *et al.*, "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646-657, 2006.
- [20] J. Zhang, and A. C. Sanderson, "JADE: Adaptive Differential Evolution With Optional External Archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945-958, 2009.
- [21] R. Tanabe, and A. Fukunaga, "Success-history based parameter adaptation for Differential Evolution," in *2013 IEEE Congress on Evolutionary Computation*. Jun. 2013, pp. 71-78.
- [22] Y. Wang, H.-X. Li, T. Huang *et al.*, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Applied Soft Computing*, vol. 18, pp. 232-247, May. 2014.
- [23] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jSO," in *2017 IEEE Congress on Evolutionary Computation*. Jun. 2017, pp. 1311-1318.
- [24] L.-y. Jia, J.-x. He, C. Zhang *et al.*, "Differential evolution with controlled search direction," *Journal of Central South University*, vol. 19, no. 12, pp. 3516-3523, Dec. 2012.
- [25] Y. Li, Z. Zhan, Y. Gong *et al.*, "Differential Evolution with an Evolution Path: A DEEP Evolutionary Algorithm," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1798-1810, 2015.
- [26] X. Zhang, and S. Y. Yuen, "A directional mutation operator for differential evolution algorithms," *Applied Soft Computing*, vol. 30, pp. 529-548, May. 2015.
- [27] L. Ching-Hung, C. Pei-Yuan, and H. H. Chang, "A Novel Hybrid Differential Evolution Algorithm: A Comparative Study on Function Optimization," *Lecture Notes in Engineering & Computer Science*, vol. 2189, no. 1, 2011.
- [28] Y. Zhou, W. Yi, L. Gao *et al.*, "Adaptive Differential Evolution With Sorting Crossover Rate for Continuous Optimization Problems," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2742-2753, 2017.
- [29] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, "Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization," *Swarm and Evolutionary Computation*, vol. 50, pp. 100455, Nov. 2019.
- [30] J. Lampinen, and I. Zelinka, "On Stagnation of the Differential Evolution Algorithm," in *Sixth International Mendel Conference on Soft Computing*, 2000, pp. 76-83.
- [31] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution – An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1-30, Apr. 2016.
- [32] W. Gong, Z. Cai, and Y. Wang, "Repairing the crossover rate in adaptive differential evolution," *Applied Soft Computing*, vol. 15, pp. 149-168, Feb. 2014.
- [33] A. W. Mohamed, and A. K. Mohamed, "Adaptive guided differential evolution algorithm with novel mutation for numerical optimization," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 2, pp. 253-277, Feb. 2019.
- [34] R. Storn, and K. Price, *Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces*, ICSI, 1995.
- [35] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A Parameter Study for Differential Evolution," in *WSEAS Int. Conf. on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*. 2002, pp. 293-298.
- [36] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *2005 IEEE Congress on Evolutionary Computation*, Sep. 2005, Vol. 1, pp. 506-513.
- [37] K. Zielinski, P. Weitkemper, R. Laur *et al.*, "Parameter Study for Differential Evolution Using a Power Allocation Problem Including Interference Cancellation," in *2006 IEEE International Conference on Evolutionary Computation*, Jun. 2006, pp. 1857-1864.
- [38] J. Liu, and J. Lampinen, "A Fuzzy Adaptive Differential Evolution Algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448-462, Jul., 2005.
- [39] A. K. Qin, and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *2005 IEEE Congress on Evolutionary Computation*, Sep. 2005, Vol. 2, pp. 1785-1791.



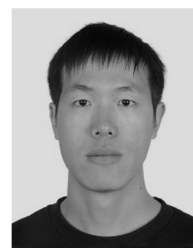
- [40] W. Yu, M. Shen, W. Chen *et al.*, "Differential Evolution With Two-Level Parameter Adaptation," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 1080-1099, 2014.
- [41] L. Cui, G. Li, Z. Zhu *et al.*, "Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism," *Information Sciences*, vol. 422, pp. 122-143, Jan. 2018.
- [42] G. Sun, J. Peng, and R. Zhao, "Differential evolution with individual-dependent and dynamic parameter adjustment," *Soft Computing*, vol. 22, no. 17, pp. 5747-5773, Sep. 2018.
- [43] J. Brest, and M. Sepesy Maučec, "Population size reduction for the differential evolution algorithm," *Applied Intelligence*, vol. 29, no. 3, pp. 228-247, Dec. 2008.
- [44] V. Tirronen, and F. Neri, "Differential Evolution with Fitness Diversity Self-adaptation," *Nature-Inspired Algorithms for Optimisation*, R. Chiong, ed., pp. 199-234, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
- [45] W. Zhu, Y. Tang, J.-a. Fang *et al.*, "Adaptive population tuning scheme for differential evolution," *Information Sciences*, vol. 223, pp. 164-191, Feb. 2013.
- [46] A. P. Piotrowski, "Review of Differential Evolution population size," *Swarm and Evolutionary Computation*, vol. 32, pp. 1-24, Feb. 2017.
- [47] R. Tanabe, and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *2014 IEEE Congress on Evolutionary Computation*, Jul. 2014, pp. 1658-1665.
- [48] S. Das, A. Abraham, U. K. Chakraborty *et al.*, "Differential Evolution Using a Neighborhood-Based Mutation Operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526-553, 2009.
- [49] A. W. Mohamed, "An improved differential evolution algorithm with triangular mutation for global numerical optimization," *Computers & Industrial Engineering*, vol. 85, pp. 359-375, Jul. 2015.
- [50] S. M. Islam, S. Das, S. Ghosh *et al.*, "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)*, vol. 42, no. 2, pp. 482-500, 2012.
- [51] L. M. Zheng, S. X. Zhang, K. S. Tang *et al.*, "Differential evolution powered by collective information," *Information Sciences*, vol. 399, pp. 13-29, Aug. 2017.
- [52] X. He, and Y. Zhou, "Enhancing the performance of differential evolution with covariance matrix self-adaptation," *Applied Soft Computing*, vol. 64, pp. 227-243, Mar. 2018.
- [53] H. Peng, Z. Guo, C. Deng *et al.*, "Enhancing differential evolution with random neighbors based strategy," *Journal of Computational Science*, vol. 26, pp. 501-511, May. 2018.
- [54] Y. Wang, Z. Cai, and Q. Zhang, "Enhancing the search ability of differential evolution through orthogonal crossover," *Information Sciences*, vol. 185, no. 1, pp. 153-177, 2012.
- [55] Y.-W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 41-53, 2001.
- [56] S. M. Guo and C. C. Yang, "Enhancing Differential Evolution Utilizing Eigenvector-Based Crossover Operator," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 31-49, 2015.
- [57] N.-D. Hoang, "NIDE: A Novel Improved Differential Evolution for Construction Project Crashing Optimization," *Journal of Construction Engineering*, vol. 2014, p. 136397, Oct. 2014.
- [58] C. Wang and J. H. Gao, "A differential evolution algorithm with cooperative coevolutionary selection operation for high-dimensional optimization," *Optimization Letters*, vol. 8, no. 2, pp. 477-492, Sep. 2014.
- [59] R. C. Addawe, "Analysis of the SA-like selection operator in differential evolution-simulated annealing (DESA) optimization algorithm," in *AIP Conference Proceedings*, 2019, vol. 2184, no. 1: AIP Publishing LLC, p. 060067.
- [60] G. Wu, R. Mallipeddi, and P. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization," Nanyang Technological University, Singapore, Tech. Rep. 2016.
- [61] J. J. Liang, B. Y. Qin, P. N. Suganthan *et al.*, "Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization," Zhengzhou University, Zhengzhou, Tech. Rep. 2012/2013.
- [62] A. W. Mohamed, and P. N. Suganthan, "Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation," *Soft Computing*, vol. 22, no. 10, pp. 3215-3235, May. 2018.
- [63] W. Gong, and Z. Cai, "Differential Evolution With Ranking-Based Mutation Operators," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2066-2081, 2013.
- [64] W. Yi, Y. Zhou, L. Gao *et al.*, "An improved adaptive differential evolution algorithm for continuous optimization," *Expert Systems with Applications*, vol. 44, pp. 1-12, Feb. 2016.
- [65] A. Viktorin, R. Senkerik, M. Pluhacek *et al.*, "Distance based parameter adaptation for Success-History based Differential Evolution," *Swarm and Evolutionary Computation*, vol. 50, pp. 100462, Nov. 2019.
- [66] A. K. Mohamed, A. W. Mohamed, E. Z. Elfeky, and M. Saleh, "Enhancing AGDE Algorithm Using Population Size Reduction for Global Numerical Optimization," in *International Conference on Advanced Machine Learning Technologies & Applications*, 2018.
- [67] N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, Jun. 2017, pp. 372-379.
- [68] A. A. Hadi, A. Wagdy, and K. Jambi, "Single-objective real-parameter optimization: Enhanced LSHADE-SPACMA algorithm," *King Abdulaziz Univ., Jeddah, Saudi Arabia, Tech. Rep.*, 2018.
- [69] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M. Jambi, "LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, Jun. 2017, pp. 145-152.
- [70] A. Kumar, R. K. Misra, and D. Singh, "Improving the local search capability of Effective Butterfly Optimizer using Covariance Matrix Adapted Retreat Phase," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, Jun. 2017, pp. 1835-1842.



**LIANZHENG CHENG** received his B.Sc. degree from China University of Geoscience (Wuhan), in 2010, the M.Sc. degree from the University of Chinese Academy of Sciences, in 2014. He is currently pursuing the Ph.D. Degree with the Institute of Geochemistry, Chinese Academy of Sciences, Guiyang, China. His research interests include evolutionary computing, numerical simulation of partial differential equations, and joint inverse problem in geophysics.



**YUN WANG** was born in Dalian, Liaoning, China in 1969. He received the B.S. in applied geophysics from the Changchun College of Geology in 1992 and the Ph.D. degree in solid geophysics from Institute of Geophysics, Chinese Academy of Sciences in 1998. From 2000 to 2009, he worked as an Associate Professor in Institute of Geology and Geophysics, CAS. Now he takes position as professor in China University of Geosciences, Beijing. He has long engaged in the theory of the seismic anisotropy and multi-component technology.



**CHAO WANG** received the M.S. and Ph.D. degree in information and communication engineering from Xi'an Jiaotong University, Xi'an, China in 2012. From 2013 to 2015, he was a Research Assistant, and since 2016, he has been an Assistant Professor with the State Key Laboratory of Ore Deposit Geochemistry, Institute of Geochemistry, Chinese Academy of Sciences. His research interests include geophysical signal time-frequency analysis,

multiwave and multicomponent seismic data denoising, wave-field separation, compressed sensing, and inverse problems.



**ALI WAGDY MOHAMED** received his B.Sc., M.Sc. and Ph.D. degrees from Cairo University, in 2000, 2004 and 2010, respectively. Ali Wagdy is an Associate Professor at Operations Research department, Faculty of Graduate Studies for Statistical Research, Cairo University, Egypt. Currently, He is an Associate Professor of Statistics at Wireless Intelligent Networks Center (WINC), Faculty of Engineering and Applied

Sciences, Nile University. He serves as reviewer of more than 50 international accredited top-tier journals and has been awarded the Publons Peer Review Awards 2018, for placing in the top 1% of reviewers worldwide in assorted field. He is editor in more than 10 journals of information sciences, applied mathematics, Engineering, system science and Operations Research. He has presented and participated in more than 5 international conferences. He participated as a member of the reviewer committee for 35 different conferences sponsored by Springer and IEEE. He has obtained Rank 3 in CEC'17 competition on single objective bound constrained real-parameter numerical optimization in Proc of IEEE Congress on Evolutionary Computation, IEEE-CEC 2017, San Sebastián, Spain. Besides, Obtained Rank 3 and Rank 2 in CEC'18 competition on single objective bound constrained real-parameter numerical optimization and Competition on Large scale global optimization, in Proc of IEEE Congress on Evolutionary Computation, IEEE-CEC 2017, Sao Paulo, Brazil. He published more than 55 papers in reputed and high impact journals like Information Sciences, Swarm and Evolutionary Computation, Computers & Industrial Engineering, Intelligent Manufacturing, Soft Computing and International Journal of Machine Learning and Cybernetics. He is interested in mathematical and statistical modeling, stochastic and deterministic optimization, swarm intelligence and evolutionary computation. Additionally, he is also interested in real world problems such as industrial, transportation, manufacturing, education and capital investment problems. 2 PhD and 1 master have been completed under his supervision.



**TIAOJIE XIAO** was born in 1991, he received his B.Sc. degree from Central South University in 2013, and received his M.Sc. and Ph.D. Degrees from University of Chinese Academy of Sciences. He is a research assistant professor of National University of Defense Technology. He serves as reviewer of several international journals. He published more than 10 papers in Journals such as Geophysical Prospecting, Journal of Applied Geophysics, Exploration Geophysics, Mathematical Problems in Engineering and so forth. Besides, he is interested in geophysical,

parallel computational and mathematical problems.