IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 August 2023, accepted 26 August 2023, date of publication 4 September 2023, date of current version 15 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3312010

RESEARCH ARTICLE

An Enhanced Adaptive Differential Evolution Algorithm With Multi-Mutation Schemes and Weighted Control Parameter Setting

MENGNAN TIAN, YANHUI MENG, XINGSHI HE, QINGQING ZHANG, AND YANGHAN GAO

School of Sciences, Xi'an Polytechnic University, Xi'an 710048, China

Corresponding authors: Mengnan Tian (mengnan_tian@xpu.edu.cn) and Xingshi He (xsh1002@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 12101477, and in part by the Scientific Research Startup Foundation of Xi'an Polytechnic University under Grant BS202052.

ABSTRACT Differential evolution (DE) algorithm is one of the most effective and efficient heuristic approaches for solving complex black box problems. But it still easily suffers from premature convergence and stagnation. To alleviate these defects, this paper presents a novel DE variant, named enhanced adaptive differential evolution algorithm with multi-mutation schemes and weighted control parameter setting (MWADE), to further strengthen its search capability. In MWADE, a multi-schemes mutation strategy is first proposed to properly exploit or explore the promising information of each individual. Herein, the whole population is dynamically grouped into three subpopulations according to their fitness values and search performance, and three different mutant operators with various search characteristics are respectively adopted for each subpopulation. Meanwhile, in order to ensure the exploration of algorithm at the later evolutionary stage, a weight-controlled parameter setting is proposed to suitably assign scale factors for different differential vectors. Moreover, a random opposition mechanism with greedy selection is introduced to avoid trapping in local optima or stagnation, and an adaptive population size reduction scheme is devised to further promote the search effectiveness of algorithm. Finally, to illustrate the performance of MWADE, thirteen typical algorithms are adopted and compared with MWADE on 30 functions from IEEE CEC 2017 test suite with different dimensions, and the effectiveness of its proposed components are also investigated. Numerical results indicate that the proposed algorithm has a better search performance.

INDEX TERMS Differential evolution, numerical optimization, mutation strategy, parameter control, population size reduction scheme.

I. INTRODUCTION

Generally, most engineering problems can always be converted and solved as optimization problems, such as UAV swarm configuration [1], image segmentation [2], [3], job-shop scheduling [4], and so on. Unlike traditional optimization algorithms, intelligent optimization algorithms have been proved to be one of the most effective methods to resolve such kind of complex engineering problems [5], [6], [7], [8], [9]. Due to its simple but robust structure, and few requirement of control parameters, DE algorithm [8] has been widely and successfully applied in the fields of clustering [10], [11], [12], neural networks [13], [14], [15], economic load dispatch [16], and so on. Although the above-mentioned extensive application results demonstrate the powerful search capability and application context of DE, it still suffers from falling into local optima [5].

Up to now, researchers have proposed a series of schemes to enhance the performance of DE [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41]. Among them, two popular approaches are designing proper mutation strategy and parameter control methods. For example, in terms of improving mutation strategy, by introducing an

The associate editor coordinating the review of this manuscript and approving it for publication was Bijoy Chand Chatterjee.

external archive to store inferior solutions from the historical population, Zhang and Sanderson [23] proposed a mutation strategy (JADE) to balance the global search and local exploitation of algorithm. Gong and Cai [24] proposed a ranking-based mutation operator (rank-DE) to enhance DE's exploitation ability, where some parents in the mutation operators are selected in proportion to their rankings in the current population. Cheng et al. [25] proposed a new ranking framework (FDDE) to balance exploration and exploration by combining the fitness ranking and diversity ranking of individual to define its final ranking. In addition, by properly integrating the benefits of multi-mutation operators, numerous DE variants have been also researched to adapt the varying search requirements of algorithm at different evolutionary stages. For instance, Mallipeddi et al. [26] used a pool of distinct mutation strategies along with a pool of values for each control parameter to produce offspring (EPSDE). Meng and Yang [27] proposed a novel historical-solution based mutation strategy and an inferior-solution based mutation strategy to simultaneously enhance the exploration and exploitation of algorithm. Ghosh et al. [28] proposed a stochastically switchable manner mutation strategy to resolve complex optimization problems, in which a population centrality based mutation strategy and the difference mean based mutation strategy are suitably incorporated and integrated. Wang et al. [29] proposed a novel mechanism by introducing an accompanying population composed of the suboptimal solutions to adaptively optimize the mutation strategy and control parameters for population. Sun et al. [30] proposed a two-level parameter cooperation-based mechanism to determine whether the regeneration operation is executed, thus regulating the global exploration and local exploitation of algorithm.

Meanwhile, to avoid manually tuning parameters and strengthen the adaptivity of algorithm, some adaptive parameter settings have been developed [31], [32], [33], [34], [35], [36], [37], [38], [39], [40]. In detail, Tanabe and Fukunaga [34] proposed an adaptive DE variant using the success-history feedbacks, called SHADE, to improve the robustness of DE. Through utilizing the information of high-quality solutions and previous experiences, Qin et al. [35] proposed a self-adaptive control parameters update setting to avoid the expensive computational cost of finding the most appropriate relevant parameter values. Meng et al. [36] proposed a novel adaptive parameter control techniques for DE to expect a better performance by employing the success probability and the position information of population. Zhang et al. [37] proposed a general framework for adaptive parameter control based on neural network to avoid the excessive dependence on its hyperparameters. To address the problem of bias in parameter adaptation, Stanovov et al. [38] proposed a generalized Lehmer mean and a linear bias reduction to create the scale factor and crossover rate for every individual during the evolution process. To avoid the defect that the bad F is often considered as the good value and vice versa, Meng and Yang [32] proposed a parameter adaptation mechanism based on grouping strategy, where the scale factors of all individuals obey the same Cauchy distribution and the crossover rates of the individuals in the same group obey the same Gaussian distribution. Additionally, Tanabe and Fukunaga [39] developed a linear reduction mechanism for population size (L-SHADE) to further upgrade the exploration and convergence of algorithm at the earlier and later search stages respectively. Besides, Zeng et al. [40] proposed a novel population adaptive method to improve the global search ability of algorithm by combing the characteristics of linear and sawtooth functions.

Even though the mentioned methods above have effectively enhanced the performance of DE, there are still some shortcomings. For example, they do not take into consideration the compatibility between the mutation operator and the individual, and the exploration of algorithm is always decreasing during the later evolution stage. In this paper, a novel DE variant, named MWADE, is proposed to mitigate these deficiencies. It introduces a multi-schemes mutation strategy, a weighted control parameter setting, a random opposition learning mechanism and an adaptive population size reduction mechanism. Specifically, the main contributions of the paper are as follows.

(1) A multi-scheme mutation strategy is proposed by adaptively dividing the whole population into three subpopulations according to their fitness values and search performance, and suitably assigning one matched mutation operator for each subpopulation. In this strategy, the size of each subpopulation is dynamically adjusted according to its search performance, and the subpopulation with the better or worse fitness values are separately assigned with a more exploitative or explorative mutation operators. So, this new strategy can effectively adjust the exploration and exploitation of algorithm, and fully make use of the promising information of distinct individuals.

(2) A weight control parameter setting is developed by using the evolutionary information of population. In this method, smaller scale factors will be generated for individuals at the beginning of evolution, and gradually increase when the iteration goes. Thereby, it can be available to strengthen the search range of algorithm at the later evolution stage.

(3) A stochastic opposition learning mechanism is further introduced to degrade the probability of algorithm falling into a local optimum by randomly generating opposing solutions to the successful solution in the selection phase. Meanwhile, a new nonlinear scheme is also presented to adaptively adjust the size of population based on the individual information.

(4) The performance of MWADE is verified by comparing it with both 7 typical DE algorithms and six other heuristic approaches on 30 benchmark functions from IEEE CEC2017 test suite [42] with various dimensions. The experimental results demonstrate the effectiveness and superiority of the proposed MWADE.

The paper is organized as follows. The classical DE algorithm is introduced in Section II. The proposed algorithm

MWADE is introduced in detail in Section III. In Section IV, the numerical experiments and statistics test are presented. Section V summarizes this paper.

II. THE CLASSICAL DE

This section shall describe the original differential evolution algorithm, which comprises four steps: initialization, mutation, crossover, and selection.

A. INITIALIZATION

Similar to other evolutionary algorithms, DE first creates a population $\boldsymbol{P} = \left\{x_{i,j} | i = 1, 2, \ldots, NP; j = 1, 2, \ldots, D\right\}$ by randomly generating multiple points in the decision space, where NP is the population size, and D is the individual dimension. The i-th individuals $\boldsymbol{x}_i$ at the first generation $(G = 0)$ can be generated by

$$\boldsymbol{x}_{i,j}^0 = \boldsymbol{lb}_j + rand \cdot (\boldsymbol{ub}_j - \boldsymbol{lb}_j) \tag{1}$$

where $\boldsymbol{x}_{i,j}^0$ denotes the j-th component of $\boldsymbol{x}_i^0$, and $rand$ is a random number between 0 and 1 within uniform distribution, $\boldsymbol{ub}_j$ and $\boldsymbol{lb}_j$ are respectively the upper and lower bounds of search space.

B. MUTATION

In this step, the mutation operator is applied on each individual $\boldsymbol{x}_i^G$ to produce its mutant individual $\boldsymbol{v}_i^G$. Specifically, the detail procedures of five commonly used mutation operations are described as follows:

$$DE/rand/1: \boldsymbol{v}_i^G = \boldsymbol{x}_{r1}^G + F \times (\boldsymbol{x}_{r2}^G - \boldsymbol{x}_{r3}^G) \tag{2}$$

$$DE/rand/2:$$
$$\boldsymbol{v}_i^G = \boldsymbol{x}_{r1}^G + F \times (\boldsymbol{x}_{r2}^G - \boldsymbol{x}_{r3}^G) + F \times (\boldsymbol{x}_{r4}^G - \boldsymbol{x}_{r5}^G) \tag{3}$$

$$DE/best/1: \boldsymbol{v}_i^G = \boldsymbol{x}_{best}^G + F \times (\boldsymbol{x}_{r1}^G - \boldsymbol{x}_{r2}^G) \tag{4}$$

$$DE/best/2:$$
$$\boldsymbol{v}_i^G = \boldsymbol{x}_{best}^G + F \times (\boldsymbol{x}_{r1}^G - \boldsymbol{x}_{r2}^G) + F \times (\boldsymbol{x}_{r3}^G - \boldsymbol{x}_{r4}^G) \tag{5}$$

$$DE/current-to-best/1:$$
$$\boldsymbol{v}_i^G = \boldsymbol{x}_i^G + F \times (\boldsymbol{x}_{best}^G - \boldsymbol{x}_i^G) + F \times (\boldsymbol{x}_{r1}^G - \boldsymbol{x}_{r2}^G) \tag{6}$$

where $\boldsymbol{x}_{r1}^G$, $\boldsymbol{x}_{r2}^G$, $\boldsymbol{x}_{r3}^G$, $\boldsymbol{x}_{r4}^G$, and $\boldsymbol{x}_{r5}^G$ represent five individuals randomly selected from population $\boldsymbol{P}^G$, $\boldsymbol{x}_{best}^G$ and $\boldsymbol{x}_i^G$ respectively represent the best and current individuals of the G-th generation population, and F is scale factor.

C. CROSSOVER

Subsequently, the crossover operation is used to produce the offspring $\boldsymbol{u}_i^G$ for each individual $\boldsymbol{x}_i^G$. In detail, the binomial crossover can be expressed by

$$\boldsymbol{u}_{i,j}^G = \begin{cases} \boldsymbol{x}_{i,j}^G, & if\ rand[0,1]_{i,j} < CR\ or\ j = jrand, \\ \boldsymbol{v}_{i,j}^G, & otherwise. \end{cases} \tag{7}$$

Herein $\boldsymbol{v}_{i,j}^G$ and $\boldsymbol{u}_{i,j}^G$ represent the j-th components of $\boldsymbol{v}_i^G$ and $\boldsymbol{u}_i^G$ respectively, $CR \in [0, 1]$ is the crossover rate, and $jrand$ is a random integer located in $[1, D]$.

D. SELECTION

Finally, for one target individual $\boldsymbol{x}_i^G$ and its corresponding offspring $\boldsymbol{u}_i^G$, the selection operation is executed to decide which one among them will enter the next population. The greedy selection strategy can be drawn as

$$\boldsymbol{x}_i^{G+1} = \begin{cases} \boldsymbol{u}_i^G, & if\ f\left(\boldsymbol{u}_i^G\right) < f\left(\boldsymbol{x}_i^G\right), \\ \boldsymbol{x}_i^G, & otherwise. \end{cases} \tag{8}$$

Here, $f\left(\boldsymbol{u}_i^G\right)$ and $f\left(\boldsymbol{x}_i^G\right)$ denote the fitness values of $\boldsymbol{u}_i^G$ and $\boldsymbol{x}_i^G$, respectively.

Noticeably, once the DE algorithm is called, the first step executed is initialization, followed by mutation, crossover and selection operations in turn until the prescribed termination criterion is met.

III. PROPOSED MWADE

In this section, a new adaptive differential evolution algorithm (MWADE) is proposed, including a multi-schemes mutation strategy, a weighted control parameter setting, a random opposition learning mechanism and an adaptive population size reduction mechanism.

A. MULTI-SCHEMES MUTATION STRATEGY

As well known, mutation strategy is crucial in determining the performance of DE. Although numerous single mutation strategies are existing up to now, they are still inadequate in adaptively meeting the varying search requirements at different evolutionary stages or on various problems. Following this, we propose a multi-schemes mutation strategy by effectively integrating the advantages of different strategies here.

In the proposed strategy, the population is dynamically divided into three subpopulations according to their fitness values in each iteration. Distinct mutation operators are assigned to each sub-population based on its specific characteristic. Specifically, based on the fitness values of individuals, the top $s_{better}\%$ of them are considered as elite ones, the bottom $s_{worse}\%$ of those are classified as disadvantaged individuals, and the rest $(1\text{-}s_{better}\text{-}s_{worse})\%$ ones are called as ordinary individuals. Notably, the elite individuals are likely closer to optimal solution, thus requiring better exploitation to ensure the convergence of algorithm, the ordinary individuals are more helpful to balance the exploitation and exploration, while the disadvantaged individuals are usually far from optimal solution, so requiring to enhance the diversity of population. Based on this consideration, this paper respectively utilizes the following three distinct mutation operators for elite, ordinary and disadvantaged individuals:

$$\boldsymbol{v}_i^G = \boldsymbol{x}_{best,G} + F_i \times (\boldsymbol{x}_{r1}^G - \boldsymbol{x}_{r2}^G) + F_\tau \times (\boldsymbol{x}_{r3}^G - \tilde{\boldsymbol{x}}_{r4}^G), \tag{9}$$

$$\boldsymbol{v}_i^G = \boldsymbol{x}_i^G + F_i \times (\boldsymbol{x}_{pbest}^G - \boldsymbol{x}_i^G) + F_i \times (\boldsymbol{x}_{r1}^G - \tilde{\boldsymbol{x}}_{r2}^G), \tag{10}$$

$$\boldsymbol{v}_i^G = \boldsymbol{x}_{r1}^G + F_i \times (\boldsymbol{x}_{r2}^G - \boldsymbol{x}_{r3}^G) + F_\tau \times (\boldsymbol{x}_{r4}^G - \tilde{\boldsymbol{x}}_{r5}^G). \tag{11}$$

Here, F_i and F_τ denote the scale factor and the weighted scale factor separately, $\tilde{\boldsymbol{x}}_{r4}^G$ is an individual randomly selected from the set $\boldsymbol{A} \cup \boldsymbol{B}$, $\boldsymbol{x}_{pbest}^G$ represents an individual randomly selected from the top *100p%* of population $\boldsymbol{P}$, $\tilde{\boldsymbol{x}}_{r2}^G$ represents an individual randomly selected from the set $\boldsymbol{P} \cup \boldsymbol{A}$, and $\tilde{\boldsymbol{x}}_{r5}^G$ is an individual random selected from the set $\boldsymbol{B}$. Moreover, $\boldsymbol{A}$ is the set of history failure solutions as the same as in [33], and $\boldsymbol{B}$ is another archive with size r^{brc*} *NP*, storing the failure solutions created by opposition learning, which will be defined in the part A of Section III.

Particularly, from the definitions of Eqs. (9), (10) and (11), one can see that Eq. (9) takes the best individual as the base vector and simultaneously employs one difference vector generated from population and one difference vector formed by a randomly chosen individual from population and one from the set $\boldsymbol{A} \cup \boldsymbol{B}$ to create the mutant individual. Moreover, based on the constructions of the sets $\boldsymbol{A}$ and $\boldsymbol{B}$, the last difference term shall enhance the diversity of search direction, but likely form the more promising ones, thus being helpful to improve the search effectiveness of algorithm. Thereby, Eq. (9) has promising exploitation ability but is more explorative than the original operators *DE/best/1* and *DE/best/2*. Thus, the assignment of Eq. (9) for the elite individuals can effectively improve the convergence of algorithm, but not rapidly cause the reduction of population diversity. Meanwhile, for Eq. (10), it takes the current individual as the base vector and makes it search towards a randomly chosen best individual from population, while it also employs one difference term formed by population and the set $\boldsymbol{A}$ to disturb its search range. Then, compared to Eq. (9), it can effectively make full use of the promising information of the better individuals to guide the search, so ensuring the effectiveness of algorithm. Thus, the assignment of Eq. (10) for the ordinary individuals be capable of availably promote the balance of exploitation and exploration of algorithm. Moreover, with respect to Eq. (11), it takes one randomly selected individual from population as the base vector, and employs two difference terms to enhance its search range. In detail, the first difference term is generated by the current population, and the last one is formed by both the current population and the set $\boldsymbol{B}$, which might also more likely create more promising search direction. Then compared to the original operator *DE/rand/2*, Eq. (11) have more exploitation ability on the search space, while compared to Eqs. (9) and (10), it has more powerful ability to enhance the diversity of population during the search process. Therefore, the assignment of Eq. (11) for the disadvantaged individuals is able to ensure the exploration of algorithm, thus avoiding trapping into local optima. In all, from the above descriptions, the proposed strategy can effectively exploit or explore the promising information of every individual, and thus improve the search effectiveness of algorithm.

Moreover, it can be also found that the percentages of the elite, ordinary and disadvantaged individuals play an important role in the performance of DE. To adjust properly and adaptively the exploration or exploitation ability of algorithm, the values of s_{better} and s_{worse} are dynamically updated by

$$\begin{cases} s_{better} = \frac{nu_b}{nu_b+nu_w+nu_a}, \\ s_{worse} = \frac{nu_w}{nu_b+nu_w+nu_a}. \end{cases} \tag{12}$$

where nu_b, nu_w and nu_a are the number of the successful elite, worse and ordinary individuals respectively, and we let $s_{better}, s_{worse} \in [0.05, 0.5]$ in this paper to avoid the empty of each subpopulation. Specifically, from Eq. (12), one can see that the last search performance of each operator on its corresponding individuals is utilized to measure its adaptivity on them and further adjust their size. Particularly, when smaller number of successful individuals are obtained by one operator, this means that the search requirement of its corresponding individuals may be not matched with the search characteristic of the current operator. Thereby, a fewer number of individuals should be assigned for it in the later iterations. Moreover, for one operator, the percentage of its successful individuals among all the successful ones in the last generation is used to measure its match level with its corresponding individuals and further adjust its later search resources, which can not only validly reflect the search state of each operator, but also its relative overall performance compared to other operators. Thereby, this setting can dynamically and properly adjust the sizes of the elite, ordinary and disadvantaged individuals during the search process, and thus further effectively balance the exploration and exploitation of algorithm.

B. WEIGHTED CONTROL PARAMETER SETTING

Similar to the mutation operator, parameter control setting also has a vital effect on the performance of algorithm. Over the last years, many methods have been developed to enhance the adjust ability of DE. However, the existing approaches always regard the scale factors of all different terms in mutation scheme with the same value, and not fully consider their special targets during the parameter setting, which might not availably adjust the search capability of algorithm. Particularly, with respect to the proposed first and third mutation operators in the last subsection, their difference terms are all different, and the first ones in them are just formed by the current population, while the second ones are constructed by both the current population and an external archive, which more likely provides the promising search direction and can also enhance the diversity of search direction, and thus be helpful to further improve the search performance of algorithm. Based on this consideration and to availably improve the adaptivity of algorithm at different search stages, by additionally making full use of both the characteristic of each difference term and the evolution information, a weighted control parameter setting is developed based on the history-adaptive method [39] in this subsection.

Specifically, for each individual x_i^G, its weighted scale factor F_τ can be generated by

$$\begin{cases} F_\tau = \tau \cdot F_i, \\ \tau = 0.2 + \sin(\frac{\pi}{6} \cdot (1 + \frac{2 \cdot nfe}{nfe_{\max}})). \end{cases} \tag{13}$$

Herein, nfe and $nfe_{\max}$ denote the number of the current fitness evaluations and the maximum number of the fitness evaluations respectively, F_i is generated by the adaptive method based on the history successful records in literature [39]. In particular, we have that if $F_\tau > 1$, it will be truncated to 1, while a new value is generated for it when $F_\tau < 0$.

In detail, the values of F_i and CR_i can be created as follow:

$$\begin{cases} CR_i = randn_i(M_{CR,r_i}, 0.1), \\ F_i == randc_i(M_{F,r_i}, 0.1), \\ M_{CR,k} = \begin{cases} mean_{WA}\,(\boldsymbol{S}_{CR}), & if\ \boldsymbol{S}_{CR} \neq \emptyset, \\ M_{CR,k}, & otherwise. \end{cases} \\ M_{F,k} = \begin{cases} mean_{WL}\,(\boldsymbol{S}_F), & if\ \boldsymbol{S}_F \neq \emptyset, \\ M_{F,k}, & otherwise. \end{cases} \\ mean_{WL}\,(\boldsymbol{S}) = \frac{\sum_{k=1}^{|S_F|} w_k \cdot \boldsymbol{S}^2}{\sum_{k=1}^{|S|} w_k \cdot \boldsymbol{S}}, \\ w_k = \frac{|f(u_k^G) - f(x_k^G)|}{\sum_{k=1}^{|S_{CR}|} |f(u_k^G) - f(x_k^G)|}. \end{cases} \tag{14}$$

Here, $randn_i(\mu, \sigma^2)$ and $randc_i(\mu, \sigma^2)$ is the normal distribution and Cauchy distribution with a mean of μ and variance of σ^2, respectively. If $CR_i > 1$ or $CR_i < 0$, it will be truncated to 1 or 0. Similarly, if $F_i > 1$, it will be truncated to 1. Else, if $F_i < 0$, it will be regenerated. Moreover, $M_{CR,i}$ and $M_{F,i}\,(i = 1, 2, \ldots, H)$ are the *i-th* mean value or location value stored in one history archive, and are all initialized to 0.5 at the beginning of algorithm, and $\boldsymbol{S}_{CR}$ and $\boldsymbol{S}_F$ are the sets of the successful crossover rates and scale factor, separately. For clear, the detail procedure about them can be further found in literature [39].

From Eqs. (13) and (14), one can find that with respect to the mutation schemes shown by Eqs. (9) and (11), their first difference vectors are scaled by F_i, and this is always generated by making full use of the history feedback information of individuals. Meanwhile, for their second difference vectors, the weighted scale factor F_τ is used to scale them, and dynamically and increasingly adjusted as the iteration goes. Particularly, F_τ is a smaller value at the beginning of evolution, while becomes more and more large when iteration goes. Moreover, the crossover rates of all target individuals are the same ones, and they are always generated based on their history successful records. Then the proposed setting can not only adapt to the varying search need of population during the evolution process, but also effectively adjust the search ability of algorithm further. In detail, the weighted scale factor F_τ is able to effectively enhance the capability of algorithm to jump out of local optima. Compared to the existing methods, where all the difference terms in mutation are usually dealt with the same role, the new setting further considers their special characteristic to more properly adjust their associated parameters. Therefore, this proposed setting can more effectively adjust the search ability of algorithm at the varying evolution environment, and further enhance the exploration of algorithm at the later search process.

C. RANDOM OPPOSITION LEARNING MECHANISM

Generally, the diversity of population is gradually degraded as the evolution undergoes, and this may affect the capability of searching a broader range in the domain space for DE. In this paper, in order to avoid this flaw as soon as possible, a random opposition learning mechanism is also introduced in the proposed method after the selection operation.

In detail, the concrete procedure of this technique can be simply described as follows. With respect to every individual $\boldsymbol{x}_i^G$ and its associated offspring $\boldsymbol{u}_i^G$, the greedy selection method is first implemented to create a candidate solution $\boldsymbol{y}_i$ entering into the next generation by

$$\boldsymbol{y}_i^G = \begin{cases} \boldsymbol{u}_i^G, & if\ f\left(\boldsymbol{u}_i^G\right) < f\left(\boldsymbol{x}_i^G\right), \\ \boldsymbol{x}_i^G, & otherwise. \end{cases} \tag{15}$$

Here, the symbols are the same as in Section II. Then, the random opposition learning method [43] is conducted on $\boldsymbol{y}_i^G$ within the whole search space to generate its corresponding opposite individual $\widetilde{\boldsymbol{y}}_i^G$ with

$$\widetilde{\boldsymbol{y}}_i^G = (\boldsymbol{ub} + \boldsymbol{lb}) - rand(0, 1) \cdot \boldsymbol{y}_i^G \tag{16}$$

Finally, the greedy strategy is also utilized for $\boldsymbol{y}_i^G$ and $\widetilde{\boldsymbol{y}}_i^G$ based on their fitness values, and the one with better performance among them is chosen into the next population. The specific procedure of this step can be shown as

$$\boldsymbol{x}_i^{G+1} = \begin{cases} \widetilde{\boldsymbol{y}}_i^G, & if\ f\left(\widetilde{\boldsymbol{y}}_i^G\right) < f\left(\boldsymbol{y}_i^G\right), \\ \boldsymbol{y}_i^G, & otherwise. \end{cases} \tag{17}$$

For clarity, the following descriptions are further provided to show the detail procedure and effect of the proposed mechanism. For instance, with respect to one solution $\boldsymbol{x}_i^G$ and its associated trial offspring $\boldsymbol{u}_i^G$, they will be first compared based on their fitness values, and the one ($\boldsymbol{x}_i^G$ or $\boldsymbol{u}_i^G$) with smaller fitness value is recorded as $\boldsymbol{y}_i^G$ and used to randomly create its opposite individual $\widetilde{\boldsymbol{y}}_i^G$. Then $\boldsymbol{y}_i^G$ and $\widetilde{\boldsymbol{y}}_i^G$ are subsequently compared with their fitness values, and the one ($\boldsymbol{y}_i^G$ or $\widetilde{\boldsymbol{y}}_i^G$) with smaller fitness value is used to enter into the next generation. Notably, the opposite individual $\widetilde{\boldsymbol{y}}_i^G$ is randomly generated within the whole search space, and thus might not be near the current individuals, while be able to enter into the next population when it owns a more promising performance. Thereby, this mechanism can not only ensure the convergence of algorithm, but also promote its exploration during the search process.

On the other hand, during this process, aiming at making the best use of the obtained information from the DE search, we further store the failure individual $\boldsymbol{y}_i^G$ into the external archive $\boldsymbol{B}$, which is fully used in the proposed mutation operator shown in the part A of Section III.

From the above descriptions, one can see that the random opposite process is further incorporated after the general selection operation in DE, and this can possibly enhance the exploration of algorithm. So, this proposed mechanism is helpful to avoid the issue of trapping into the local optima.

D. ADAPTIVE POPULATION SIZE REDUCTION

As pointed out in literature [33], the reduction process of population size is conductive to further enhance the performance of algorithm. However, the existing methods are just designed and relative with the number of iterations. This might not be reasonable since the real search state is not always consistent with the hypothesis that more exploration and exploitation are separately needed at the earlier and later evolution stages, and the search requirement of algorithm is actually dynamical and varied with the evolution undergoing. Based on this consideration, a new adaptive population size reduction mechanism is designed by making full use of the distribution information of population.

Specifically, to effectively and dynamically estimate the search environment of algorithm during the whole evolution process, the differences between individuals are utilized to evaluate the search requirement of population, and then adjust the reduction speed of population size. For clear, at the *G-th* generation, the corresponding population size can be computed by

$$NP_G = round((NP_{\min} - NP_{ini}) \cdot (\frac{nfes}{nfes_{\max}})^{(1+e^{-\Delta f})} + NP_{ini}) \tag{18}$$

$$\Delta f = \sum_{i=1}^{NP} \left\| \boldsymbol{x}_i^G - \boldsymbol{x}_{best}^G \right\|_2 \tag{19}$$

where $NP_{\min}$ and NP_{ini} refers to the minimum and initial population size respectively, and nfe and $nfes_{max}$ are the number of the current and maximum function evaluations, respectively. Obviously, Δf denotes the sum of the euclidean distances between all individuals $\boldsymbol{x}_i$ and the current best individual $\boldsymbol{x}_{best}$, and a larger one means that there are enough diversity between individuals, so needing to reduce slowly the population size to maintain the exploration of algorithm. In contracts, a smaller value for Δf implies that the population is more likely to fall into one promising region, thus needing to speed up the convergence of algorithm by quickly reducing the population size. Thereby, this developed method can further strengthen the search capability of algorithm. Furthermore, due to the proposed mutation strategy, where the whole population is required to divide into three subpopulations and adopt different search schemes for them respectively, to ensure the search validity for each subpopulation, we let $NP_{\min}$ be 10 in this paper. Besides, the suitable choice of NP_{ini} is further discussed in the part A of Section IV.

Unlike the existing population reduction methods, the proposed mechanism makes full use of the diversity of population to adaptively adjust the reduction degree of population size, thus enhancing the search capability of algorithm further.

In summary, by integrating the multi-schemes mutation strategy, weighted control parameter setting, random opposition learning mechanism, and adaptive population size reduction mechanism, the overall framework of the proposed algorithm MWADE is shown in **Algorithm 1**.

Algorithm 1 The Framework of MWADE

Input: the initial population size NP_{ini}, the minimum population NP_{min}, the maximum number of fitness evaluations $nfe_{\max}$; Set all elements in M_{CR}, M_F to 0.5; Set external archive $\boldsymbol{A}, \boldsymbol{B} = \emptyset$;
Let $NP = NP_{ini}$;
Initialize population $\boldsymbol{P}$ randomly, and evaluate it;
Set $nfes = NP, s_{better} = 0.2$ and $s_{worse} = 0.3$;
While $nfes \leq nfes_{\max}$ **do**
  Let $\boldsymbol{S}_{CR}, \boldsymbol{S}_F = \emptyset$;
  Sort the individuals in $\boldsymbol{P}$ based on their fitness values;
  For $i = 1 : NP$ **do**
    Calculate $F_i,\ F_\tau,\ CR_i$ by Eqs. (13)-(17);
    If $i < s_{better} \cdot NP$ **then**
      Create the corresponding mutant individual $\boldsymbol{v}_i^G$ by Eq. (9);
    Elseif $s_{better} \cdot NP < i < s_{worse} \cdot NP$ **then**
      Create the corresponding mutant individual $\boldsymbol{v}_i^G$ by Eq. (10);
    Else $i > s_{worse} \cdot NP$ **then**
      Create the corresponding mutant individual $\boldsymbol{v}_i^G$ by Eq. (11);
    End if
    Execute the crossover operation to generate $\boldsymbol{u}_i^G$ by Eq. (7) and calculate its fitness value;
    $nfes = nfes + 1$;
    Generate $\boldsymbol{y}_i^G$ by Eq. (15), update the archive $\boldsymbol{A}$, and generate its opposition individual $\tilde{\boldsymbol{y}}_i^G$ by Eq. (16);
    Calculate the fitness value of $\tilde{\boldsymbol{y}}_i^G$;
    $nfes = nfes + 1$;
    Generate $\boldsymbol{x}_i^{G+1}$ by Eq. (17), and update the archive $\boldsymbol{B}$;
  End for
  Update s_{better} and s_{worse} by Eq. (13);
  Update $\boldsymbol{S}_F$ and $\boldsymbol{S}_{CR}$;
  Calculate the new population size NP_G by Eqs. (18)-(19);
  If $NP_G < NP$ **then**
    Sort the individuals in $\boldsymbol{P}$ based on their fitness values, and remove the $NP - NP_G$ individuals from $\boldsymbol{P}$; Let $NP{=}NP_G$; Update the archives $\boldsymbol{A}$ and $\boldsymbol{B}$;
  End if
End while
Output: The best individual and its fitness value.

E. COMPLEXITY ANALYSIS

In this subsection, we will analyze the complexity of MWADE. Obviously, the difference between MWADE and the classical DE algorithm lies in the multi-schemes mutation strategy, the weighted control parameter setting, the random opposition learning mechanism and the adaptive population size reduction mechanism. Among them, the complexity of the classical DE algorithm is $O(G_{\max} \cdot NP \cdot D)$, where NP and $G_{\max}$ denote the population size and the maximum number of iterations, respectively. Then the complexity of the proposed strategies is analyzed as follows:

For the multi-scheme mutation strategy, it needs to extra rank the individuals based on their fitness values, and calculate the value of s_{better} and s_{worse}. In detail, their complexities are $O(NP \cdot \log_2 NP)$ and $O(NP)$, respectively. Then the extra complexity of the multi-scheme mutation strategy is $O(G_{\max} \cdot NP \cdot (\log_2 NP + 1))$.

For the weighted control parameter setting, it needs to extra calculate M_{CR,r_i} and M_{F,r_i}, and the value of F_τ. In detail, the complexity of this procedure is $O(2G_{\max} \cdot NP)$. Then the extra complexity of the weighted control parameter setting is $O(2G_{\max} \cdot NP)$.

For the random opposition learning mechanism, it needs to extra calculate $\widetilde{y}_i^G$. In detail, its complexity is $O(G_{\max} \cdot NP \cdot D)$. Then the extra complexity of the random opposition learning mechanism is $O(G_{\max} \cdot NP \cdot D)$. Moreover, the complexity of the adaptive population size reduction mechanism is $O(G_{\max})$.

In summary, the extra complexity of MWADE is $O(G_{\max} \cdot NP \cdot (\log_2 NP + 3 + D) + G_{\max})$, and thus its complexity is $O(G_{\max} \cdot NP \cdot (\log_2 NP + 3 + 2D) + G_{\max})$. Therefore, MWADE algorithm will not cause serious additional computational burden.

IV. NUMERICAL EXPERIMENTS

In this section, the performance of MWADE is evaluated by using 30 benchmark functions from the IEEE CEC2017 test suite [42], which includes unimodal functions (f1-f3), simple multimodal functions (f4-f9), hybrid functions (f10-f20) and composition functions (f21-f30). Meanwhile, the sensitivities of parameters in MWADE are analyzed, and we compare MWADE with both 7 well-known DE variants and six other typical non-DE algorithms to show its advantage. Moreover, the effectiveness of the proposed strategies is indicated, and the proposed algorithm is further applied for a practical problem.

In these experiments, the performance of algorithm is measured by the mean (Mean) and standard deviation (Std) of the objective function errors of 30 independent runs. For every algorithm, the maximum number of function evaluations $nfes_{\max}$ is always set to 10000*D, and the best results on each test function are marked in bold. In addition, to reveal the differences between the algorithms more intuitively, three nonparametric statistical tests, including Wilcoxon rank sum test [44], Wilcoxon signed rank test [45] and Friedman test [45], are also used to compare the performance of each algorithm. In particular, Wilcoxon rank sum test [44] and Wilcoxon signed rank test [45] at 0.05 significance level are used to verify the differences between two algorithms, while Friedman test [45] is adopted to evaluate the overall performance of each algorithm.

A. SENSITIVITY ANALYSES OF NP_{ini} AND r^{brc}

In this subsection, the sensitivities of parameters NP_{ini} and r^{brc} in MWADE are investigated, and 8 typical functions are chosen from IEEE CEC2017 test suite to indicate their effects, including unimodal functions f1 and f2, simple multimodal functions f4 and f9, hybrid functions f11 and f19, and composition functions f21 and f22. In particular, to reasonably and comprehensively analyze the effects of NP_{ini} and r^{brc} on the performance of MWADE, a series of experiments are designed and conducted by setting NP_{ini} and r^{brc} to different values. Herein, NP_{ini} and r^{brc} are set to 16D, 18D, 23D and 25D, and 1, 1.2, 1.5, 2 and 2.5, respectively. Table 1 lists the numerical results of MWADE with various NP_{ini} and r^{brc}.

From Table 1, one can find that MWADE obtains the best results on f4 and f21 when $NP_{ini} = 18D$ and $r^{brc} = 1$. Meanwhile, on function f11, MWADE has the best performance when $NP_{ini} = 25D$ and $r^{brc} = 2$, and MWADE has the best results on f19 when $NP_{ini} = 16D$ and $r^{brc} = 1.2$. Besides, MWADE gets the best results on f22 when $NP_{ini} = 23D$ and $r^{brc} = 1.5$. Moreover, with respect to functions f1, f2 and f9, MWADE can get the best results in most cases for NP_{ini} and r^{brc}, and when $r^{brc} = 1$, MWADE can always obtain the best performance on them. Furthermore, to clearly show the performance of MWADE with different NP_{ini} and r^{brc}, Fig. 1 depicts their overall ranks on these eight problems using Friedman test, where X and Y axis are the various settings of NP_{ini} and r^{brc} and the over ranks of MWADE with different settings based on Friedman test, respectively. From Fig. 1, it can be seen that MWADE has the best rank when $NP_{ini} = 18D$ and $r^{brc} = 1$. The reason for this might be that a too large NP_{ini} may cause a waste of computing resources at the beginning of search, while a two small one may degrade the exploration of algorithm. Meanwhile, a too large r^{brc} may lead to the fact that there are too many historical failure solutions incorporating the search process, resulting in degrading the search effectiveness of algorithm. Therefore, we let $NP_{ini} = 18D$ and $r^{brc} = 1$ in this paper since its excellent performance.

B. COMPARISON AND DISCUSSION

In this subsection, MWADE is compared with seven typical DE variants (EPSDE [26], jDE [46], JADE [23], SHADE [34], FDDE [25], rank-DE [24] and LSHADE [39]) and six other typical heuristic algorithms (TAPSO [47], HSOGA [48], HSSOGSA [49], EPSO [50], HSJOA [51] and ExPSO [52]) to evaluate its performance.

It should be mentioned that EPSDE [26] is a famous DE version by integrating multiple mutation operators, and jDE [46] and JADE [23] are two typical DE variants with adaptive parameter control mechanisms. Meanwhile, SHADE [34] and LSHADE [39] are two enhanced version of JADE by introducing a successful history-based parameter setting and a linear population reduction mechanism, respectively. Moreover, FDDE [25] and rank-DE [24] are two typical DE variants by making full use of the ranks of individuals based on their fitness values, and FDDE [25] is a recent developed DE algorithm. On the other hand, with respect to the six chosen non-DE algorithms, TAPSO [47] is a new variant of particle swarm algorithm (PSO), which introduces three archives to store the promising exemplars and designs an efficient learning model for each particle, while HSOGA [48] is a hybrid self-adaptive orthogonal genetic

TABLE 1. Numerical results of MWADE with various NP_{ini} and r^{brc} on eight typical functions.

Function		f1		f2		f4		f9	
NP_{ini}	r^{brc}	mean	std	mean	std	mean	std	mean	std
	1	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.49E-14	1.82E-14	**0.00E+00**	**0.00E+00**
	1.2	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.31E-14	2.08E-14	**0.00E+00**	**0.00E+00**
16D	1.5	**0.00E+00**	**0.00E+00**	3.33E-02	1.83E-01	5.12E-14	2.29E-14	**0.00E+00**	**0.00E+00**
	2	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.49E-14	1.82E-14	**0.00E+00**	**0.00E+00**
	2.5	**0.00E+00**	**0.00E+00**	1.00E-01	4.03E-01	6.06E-14	2.96E-14	**0.00E+00**	**0.00E+00**
	1	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**5.12E-14**	**1.73E-14**	**0.00E+00**	**0.00E+00**
	1.2	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.11E+00	1.16E+01	**0.00E+00**	**0.00E+00**
18D	1.5	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.87E-14	3.80E-14	**0.00E+00**	**0.00E+00**
	2	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.49E-14	3.16E-14	**0.00E+00**	**0.00E+00**
	2.5	**0.00E+00**	**0.00E+00**	3.33E-02	1.83E-01	7.20E-14	8.82E-14	**0.00E+00**	**0.00E+00**
	1	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.72E-14	9.76E-14	**0.00E+00**	**0.00E+00**
	1.2	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.11E+00	1.16E+01	**0.00E+00**	**0.00E+00**
23D	1.5	4.74E-16	2.59E-15	**0.00E+00**	**0.00E+00**	7.20E-14	2.96E-14	**0.00E+00**	**0.00E+00**
	2	9.47E-16	3.61E-15	**0.00E+00**	**0.00E+00**	1.12E-13	2.07E-13	**0.00E+00**	**0.00E+00**
	2.5	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.20E-11	6.54E-11	**0.00E+00**	**0.00E+00**
	1	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.27E-13	1.40E-13	**0.00E+00**	**0.00E+00**
	1.2	1.42E-15	4.34E-15	**0.00E+00**	**0.00E+00**	3.89E-09	2.13E-08	**0.00E+00**	**0.00E+00**
25D	1.5	1.42E-15	4.34E-15	**0.00E+00**	**0.00E+00**	3.89E-09	2.13E-08	**0.00E+00**	**0.00E+00**
	2	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.28E-14	6.24E-14	**0.00E+00**	**0.00E+00**
	2.5	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.11E-10	1.70E-09	**0.00E+00**	**0.00E+00**
Function		f11		f19		f21		f22	
Np_{ini}	r^{brc}	mean	std	mean	std	mean	std	mean	std
	1	1.22E+01	4.42E+00	1.07E+03	3.62E+02	3.03E+00	1.66E+01	2.48E+01	1.01E+01
	1.2	1.21E+01	4.27E+00	**9.86E+02**	**3.65E+02**	5.00E-13	4.68E-13	2.52E+01	1.05E+01
16D	1.5	1.16E+01	4.30E+00	9.90E+02	3.61E+02	1.71E-11	9.13E-11	2.34E+01	7.97E+00
	2	1.17E+01	4.53E+00	1.09E+03	4.60E+02	4.40E-13	2.80E-13	2.16E+01	9.04E+00
	2.5	1.37E+01	4.90E+00	9.82E+02	3.92E+02	9.82E+02	3.92E+02	9.82E+02	3.92E+02
	1	1.24E+01	6.59E+00	9.94E+02	3.54E+02	**4.40E-13**	**2.23E-13**	2.33E+01	7.82E+00
	1.2	1.13E+01	3.77E+00	1.11E+03	3.10E+02	6.04E+00	2.30E+01	2.54E+01	9.00E+00
18D	1.5	1.18E+01	4.21E+00	1.05E+03	4.97E+02	3.02E+00	1.65E+01	2.24E+01	7.59E+00
	2	1.19E+01	3.88E+00	1.06E+03	3.62E+02	4.40E-13	3.04E-13	2.51E+01	7.48E+00
	2.5	1.07E+01	5.03E+00	1.14E+03	4.15E+02	3.02E+00	1.65E+01	2.33E+01	7.54E+00
	1	1.09E+01	4.78E+00	1.11E+03	3.41E+02	2.34E-08	1.24E-07	2.61E+01	8.50E+00
	1.2	1.03E+01	3.25E+00	1.22E+03	4.04E+02	3.02E+00	1.65E+01	2.44E+01	7.97E+00
23D	1.5	1.07E+01	4.32E+00	1.12E+03	3.58E+02	2.52E-05	1.38E-04	**2.14E+01**	**9.66E+00**
	2	1.04E+01	3.84E+00	1.06E+03	3.01E+02	8.38E-06	4.59E-05	2.61E+01	9.32E+00
	2.5	9.82E+00	4.36E+00	1.06E+03	4.33E+02	5.39E-04	2.53E-03	2.25E+01	7.47E+00
	1	1.11E+01	5.53E+00	1.04E+03	4.33E+02	3.03E+00	1.66E+01	2.54E+01	8.89E+00
	1.2	1.13E+01	4.10E+00	1.09E+03	3.88E+02	1.91E-09	7.48E-09	2.37E+01	7.32E+00
25D	1.5	1.13E+01	4.10E+00	1.09E+03	3.88E+02	1.91E-09	7.48E-09	2.37E+01	7.32E+00
	2	**8.29E+00**	**3.61E+00**	1.17E+03	3.94E+02	1.49E-06	8.16E-06	2.41E+01	7.70E+00
	2.5	8.95E+00	3.52E+00	1.10E+03	3.57E+02	3.02E+00	1.65E+01	2.15E+01	7.67E+00

algorithm (GA) based on orthogonal experimental design method, which presents a self-adaptive orthogonal crossover operator and a local search scheme to enhance its exploration. HSSOGSA [49] is a new hybrid optimization algorithm by organically combining the exploitation of gravitational search algorithm (GSA) and the exploration of sperm swarm optimization (SSO). EPSO [50] is a recent and well-known version of PSO by properly integrating five different top PSO variants. Meanwhile, HSJOA [51] is an enhanced joint operations algorithm (JOA) by reexamining the positioning of the three core operations in balancing global exploration and local exploitation and adjusting their execution mechanism. Besides, ExPSO [52] is another typical PSO variant, where the swarm population is divided into three equal sub-populations, and a new search strategy is presented based on an exponential function to make particles leaps in the search space, while an adapted control of the velocity range is designed to balance the exploration and exploitation of

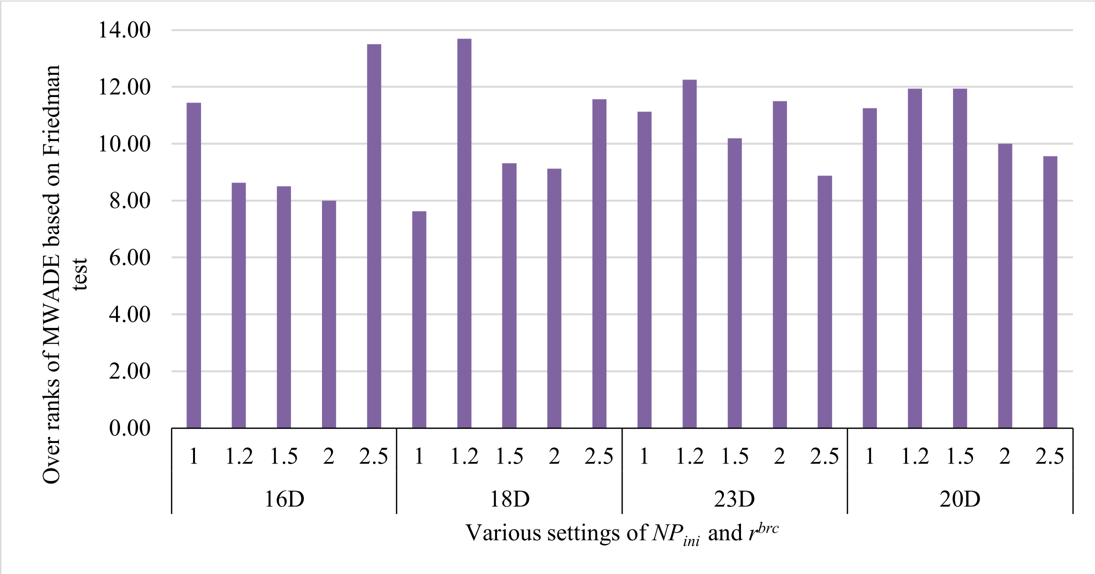

FIGURE 1. Overall ranks of MWADE with various NP_{ini} and r^{brc} on eight typical functions based on Friedman test.

algorithm. Therefore, it is reasonable to compare MWADE with these chosen approaches, and these comparisons can availably demonstrate its benefit.

In these experiments, so as to ensure a fair comparison between MWADE and its compared methods, the parameters in these chosen compared approaches are consistent with their original papers, and those in MWADE are set to the same as in Section III. In detail, the concrete parameter settings of these algorithms are further described in Table 2. Moreover, to get a sound and statistic conclusion, three test approaches, including Wilcoxon rank sum test [44], Wilcoxon signed rank test [45] and Friedman test [45], are both adopted here. In the following, the symbols "+", "-" and "≈" indicates that MWADE performs significantly better, worse and equivalent to its competitor on each function based on Wilcoxon rank sum test respectively, "R+" and "R-" denotes the rank sum that MWADE is better and worse than the compared algorithm respectively, and "p-value" is the significant result based on Wilcoxon signed rank test.

1) COMPARISON WITH SEVEN DE VARIANTS

First, seven DE variants are selected as the comparison algorithms to verify the performance of MWADE. These compared methods are EPSDE [26], jDE [46], JADE [23], SHADE [34], FDDE [25], rank-DE [24] and LSHADE [39], and 30 benchmark functions from IEEE CEC2017 test suite [35] are used as the test platform with D=30, D=50 and D=100. Tables 3-4 report the numerical and statistic results of MWADE and its seven counterparts when D=30, D=50 and D=100, respectively.

When D=30, from Table 3, one can see that MWADE has a better performance than its compared methods. In detail, MWADE gets the best results on 23 out of 30 functions, including f1-f4, f9, f11, f13-f15 and f17-f30, EPSDE has the best results on f6, f12 and f16, jDE on f6 and LSHADE on f1, f2, f5 and f7-f10. Meanwhile, JADE, SHADE, FDDE and rank-DE have no best performance on all functions. Moreover, based on the results of Wilcoxon rank sum test reported in Table 3, MWADE gets better performance than EPSDE, jDE, JADE, SHADE, FDDE, rank-DE and LSHADE on 27, 26, 26, 29, 24, 28 and 17 functions respectively, and worse results than them on 3, 1, 1, 0, 1, 0 and 4 functions, respectively. Furthermore, from the results of Friedman test in Table 3, MWADE and EPSDE, jDE, JADE, SHADE, FDDE, rank-DE and LSHADE obtain 1.63, 5.25, 5.20, 5.12, 5.02, 5.27, 6.20 and 6.33 in term of rank on all these instances. So, MWADE is more promising optimizer compared to these chosen approaches.

When D=50, from Table 4, one can also find that MWADE outperforms than other algorithms in most of instances. Specifically, MWADE has the best performance on all functions except for f4-f11, f14, f16-f17 and f19-f20, EPSDE on f6 and f16, jDE on f6 and f19, SHADE on f4, and LSHADE on f3, f5, f7-f12, f14, f17 and f20. According to Wilcoxon rank sum test, the statistic results reported in Table 4 indicate that MWADE is better than EPSDE, jDE, JADE, SHADE, FDDE, rank-DE and LSHADE on 23, 25, 25, 22, 25, 28 and 12 functions respectively, and worse than them on 5, 3, 4, 8, 4, 0 and 10 functions, respectively. Besides, based on Friedman test, MWADE and EPSDE, jDE, JADE, SHADE, FDDE, rank-DE and LSHADE obtain 2.20, 4.95, 4.95, 4.73, 4.33, 5.77, 5.97 and 3.10 in term of rank on all these instances. Thereby, MWADE has a better performance on these instances.

When D=100, from Table 5, it can be also found that MWADE performs better than other algorithms in most of

TABLE 2. Parameter settings.

Algorithms	Parameters setting
EPSDE [26]	NP=50, $F\in[0.4,0.9]$, $CR\in[0.1,0.9]$ with step size = 0.1
jDE [46]	NP=100, $F=0.5$, $Cr=0.9$, $\tau_F=\tau_{Cr}=0.1$, $F_l=0.1$, $F_u=0.9$
JADE [23]	NP=100, $\mu_F=0.5$, $F\sim C(\mu_F,0.1)$, $Cr\sim N(\mu_{Cr},0.1)$, $p=0.05$, $c=0.1$
SHADE [34]	NP=100, $\mu_F=0.5$, $F\sim C(\mu_F,0.1)$, $\mu_{Cr}=0.5$, $Cr\sim N(\mu_{Cr},0.1)$, $p=0.2$, H=100
FDDE [25]	NP=100, $F=0.5$, $Cr=0.9$, $\tau_F=\tau_{Cr}=0.1$, $F_l=0.1$, $F_u=0.9$, $w=0.2\sim0.8$
rank-DE [24]	NP=100, $F=0.5$, $Cr=0.9$, $\tau_F=\tau_{Cr}=0.1$, $F_l=0.1$, $F_u=0.9$
LSHADE [39]	NP_{ini}=18D, $F\sim C(\mu_F,0.1)$, $Cr\sim N(\mu_{Cr},0.1)$, $p=0.11$, , H=6, $NP_{\min}$=4
TAPSO [47]	w=0.7298, p_c=0.5, p_m=0.02, R_n=10
HSOGA [48]	NP=200, Q=2, S=5, P_c=0.6, P_m=0.1
HSSOGSA [49]	NP=30, G_0=1, $\alpha=20$, pH=rand(7,14)
EPSO [50]	N_{POP}=200, N_{sr}=30, d_{max}=10^{-8}, c=2
HSJOA [51]	C_{NU}=6, C_{EP}=18, NS=86
ExPSO [52]	N_1=N_2=N_3=10, c_1=-1, c_2=2, w=0.9, r=0.9, tv=0.2, k=0.2, a=b=c=2, d=e=-1
MWADE	NP_{ini}=18D, H=5, r^{brc}=1, MF=0.5, MCR=0.5, $NP_{\min}$=10

instances. In detail, MWADE has the best performance on 14 out of 30 functions, including f2, f11, f15, f18 and f21-f30, EPSDE on f5-f6, f12-f13 and f16, JADE on f4, and LSHADE on f1-f2, f7-f10, f14, f17 and f20. From statistic results of the Wilcoxon rank sum test in Table 5, MWADE is better than EPSDE, jDE, JADE, SHADE, FDDE, rank-DE and LSHADE on 24, 29, 27, 25, 30, 29 and 17 functions respectively, and worse than them on 6, 1, 3, 5, 0, 1 and 11 functions, respectively. Besides, according to Friedman test, MWADE gets the top rank among them. Therefore, MWADE is a more promising optimizer on these instances.

Moreover, in order to further show the convergence of algorithm, Fig. 2 depicts the evolution curves of eight approaches, including our proposed MWADE and its seven compared methods, on eight typical functions, including f1, f2, f4, f9, f17, f18, f27 and f28 when D=30. From Fig. 2, one can see that MWADE always owns a faster convergence speed during the evolutionary process, and gets the more accurate results when the stop condition is met. Thus, MWADE has a more promising optimization ability.

Furthermore, to further demonstrate the differences between MWADE and the chosen compared approaches, Table 6 also reports their comparison results based on Wilcoxon signed-rank tests with both D=30, D=50 and D=100. From Table 6, we can see that the significant difference between MWADE and its each compared method can always be found in all cases except for LSHADE with D=50 and D=100 at significant level $\alpha = 0.05$, and higher R+ is always obtained by MWADE. The reason for this is that the multi-schemes mutation strategy and the new adaptive population reduction scheme can effectively and adaptively adjust the exploration and exploitation of algorithm during the search process, while the random opposition learning mechanism and the weighted control parameter setting can availably reduce the risks of algorithm trapping into the local optima. Therefore, MWADE is capable of achieving a better balance between the diversity of population and the convergence of algorithm.

C. COMPARISON WITH FIVE OTHER HEURISTIC ALGORITHMS

Besides, five non-DE heuristic algorithms are further selected as comparison algorithms to show the performance of MWADE, including TAPSO [47], HSOGA [48], HSSOGSA [49], EPSO [50], HSJOA [51] and ExPSO [52]. Tables 7-8 provide the numerical and statistic results of MWADE and these six compared algorithms on IEEE CEC2017 test suite with D=30 and D=50, respectively.

When D=30, it can be seen from Table 7 that the proposed MWADE obtains the best results on all problems except for f10, f21 and f22. Specifically, TAPSO gets the best results on f21, and EPSO on f10 and f22. Moreover, based on the results of the Wilcoxon rank sum test and Friedman test in Table 7, MWADE is better than TAPSO, HSOGA, HSSOGSA, EPSO, HSJOA and ExPSO on 29, 30, 30, 27, 28 and 30 functions respectively, and MWADE and TAPSO, HSOGA, HSSOGSA, EPSO, HSJOA and ExPSO obtain 4.67, 5.83, 5.40. 2.20. 3.48, 5.22 and 1.20 in term of rank on all these instances. Thereby, MWADE has a better performance on these instances.

Meanwhile, when D=50, one can also find from Table 8 that MWADE gets absolutely superior performance on the most functions, while EPSO gets the best results on f7 and f21-f22. Besides, compared to TAPSO, HSOGA, HSSOGSA, EPSO, HSJOA and ExPSO, MWADE has better performance

TABLE 3. Numerical and statistic results of MWADE and 7 typical DE variants on IEEE CEC2017 test suite when D = 30.

Algorithms	EPSDE	jDE	JADE	SHADE	FDDE	rank-DE	LSHADE	MWADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f1	1.16E-13/	1.18E-14/	1.33E-14/	7.58E-15/	5.21E-15/	1.28E-14/	**0.00E+00/**	**0.00E+00/**
	4.66E-13(+)	5.39E-15(+)	5.19E-15(+)	7.21E-15(+)	6.97E-15(+)	5.72E-15(+)	**0.00E+00(≈)**	**0.00E+00**
f2	2.63E+11/	9.84E+03/	1.37E+01/	5.33E-01/	3.63E+00/	5.09E+03/	**0.00E+00/**	**0.00E+00/**
	5.98E+10(+)	3.66E+04(+)	2.39E+01(+)	1.14E+00(+)	1.38E+01(≈)	2.35E+04(+)	**0.00E+00(≈)**	**0.00E+00**
f3	3.45E+04/	4.64E-05/	2.86E+03/	9.09E-14/	1.04E-07/	1.82E-09/	9.47E-15/	**5.68E-15/**
	5.74E+04(+)	1.18E-04(+)	9.52E+03(+)	3.20E-14(+)	2.06E-07(+)	3.39E-09(+)	2.15E-14(≈)	**1.73E-14**
f4	1.04E+00/	6.06E+01/	3.87E+01/	7.01E-14/	5.83E+01/	4.46E+01/	1.64E-08/	**3.60E-14/**
	1.70E+00(+)	2.91E+00(+)	2.92E+01(+)	3.56E-14(+)	1.13E+01(+)	2.69E+01(+)	8.73E-08(+)	**2.79E-14**
f5	2.17E+01/	4.35E+01/	2.71E+01/	1.50E+01/	1.72E+02/	3.52E+01/	**7.58E+00/**	1.15E+01/
	1.05E-01(+)	6.21E+00(+)	5.01E+00(+)	3.46E+00(+)	9.50E+00(+)	1.07E+01(+)	**1.37E+00(-)**	3.89E+00
f6	**1.14E-13/**	**1.14E-13/**	1.67E-13/	2.22E-05/	1.77E-07/	4.37E-05/	7.02E-11/	1.48E-13/
	0.00E+00(-)	**0.00E+00(-)**	7.15E-14(≈)	2.94E-05(+)	4.86E-07(≈)	1.15E-04(+)	2.15E-10(≈)	5.30E-14
f7	7.45E+01/	7.89E+01/	5.42E+01/	4.44E+01/	2.04E+02/	8.69E+01/	**3.78E+01/**	4.26E+01/
	6.96E+00(+)	6.76E+00(+)	3.59E+00(+)	2.40E+00(+)	1.01E+01(+)	3.53E+01(+)	**1.39E+00(-)**	4.09E+00
f8	4.98E+01/	4.95E+01/	2.60E+01/	1.45E+01/	1.74E+02/	3.87E+01/	**7.44E+00/**	1.04E+01/
	7.02E+00(+)	7.02E+00(+)	3.95E+00(+)	2.55E+00(+)	9.29E+00(+)	1.44E+01(+)	**1.40E+00(-)**	2.92E+00
f9	1.70E+00/	0.00E+00/	1.81E-02/	7.23E-02/	1.51E-02/	1.96E+00/	**0.00E+00/**	**0.00E+00/**
	3.96E+00(+)	0.00E+00(≈)	8.40E-02(≈)	1.42E-01(+)	8.29E-02(+)	2.44E+00(+)	**0.00E+00(≈)**	**0.00E+00**
f10	3.58E+03/	2.75E+03/	1.99E+03/	1.30E+03/	6.65E+03/	4.09E+03/	**1.13E+03/**	2.53E+03/
	3.86E+02(+)	2.74E+02(≈)	1.88E+02(-)	1.87E+02(≈)	2.77E+02(+)	1.11E+03(+)	**1.94E+02(-)**	4.27E+02
f11	2.45E+01/	3.42E+01/	3.20E+01/	3.16E+01/	2.05E+01/	5.72E+01/	9.84E+00/	**3.89E+00/**
	1.37E+01(+)	2.89E+01(+)	2.28E+01(+)	1.45E+01(+)	2.06E+01(+)	3.67E+01(+)	3.95E+00(+)	**2.43E+00**
f12	**1.48E+01/**	7.48E+03/	1.11E+03/	1.38E+03/	1.31E+04/	1.14E+04/	1.33E+03/	4.05E+02/
	3.47E+00(-)	5.41E+03(+)	3.87E+02(+)	4.90E+02(+)	1.00E+04(+)	1.21E+04(+)	4.06E+02(+)	1.95E+02
f13	2.07E+01/	4.97E+01/	4.41E+01/	2.45E+01/	7.84E+01/	6.66E+01/	1.83E+01/	**1.66E+01/**
	3.59E+00(+)	9.49E+01(+)	1.91E+01(+)	1.51E+01(+)	1.81E+01(+)	1.34E+02(+)	4.24E+00(+)	**1.15E+00**
f14	4.01E+01/	2.40E+01/	7.22E+03/	2.89E+01/	4.90E+01/	4.33E+01/	2.22E+01/	**2.19E+01/**
	5.17E+00(+)	9.53E+00(+)	9.61E+03(+)	4.18E+00(+)	1.80E+01(+)	1.81E+01(+)	1.24E+00(≈)	**4.05E+00**
f15	7.22E+01/	1.17E+01/	2.11E+03/	3.56E+01/	1.11E+01/	4.08E+01/	5.83E+00/	**2.95E+00/**
	8.66E+01(+)	4.52E+00(+)	4.78E+03(+)	2.04E+01(+)	9.06E+00(+)	5.40E+01(+)	2.93E+00(+)	**1.36E+00**
f16	**1.50E+01/**	4.55E+02/	3.99E+02/	1.82E+02/	2.11E+02/	3.61E+02/	4.49E+01/	1.07E+02/
	1.91E-01(-)	1.42E+02(+)	1.40E+02(+)	1.20E+02(+)	2.84E+02(≈)	2.21E+02(+)	4.55E+01(≈)	9.77E+01
f17	2.13E+02/	9.64E+01/	7.36E+01/	9.43E+01/	6.70E+01/	8.39E+01/	5.80E+01/	**5.45E+01/**
	8.83E+01(+)	2.26E+01(+)	1.86E+01(+)	2.18E+01(+)	1.90E+01(+)	6.49E+01(≈)	7.79E+00(≈)	**1.40E+01**
f18	1.55E+03/	5.53E+01/	1.95E+04/	1.03E+02/	2.25E+01/	3.97E+02/	2.79E+01/	**2.21E+01/**
	1.90E+03(+)	3.26E+01(+)	5.13E+04(+)	3.23E+01(+)	6.42E+00(+)	8.59E+02(+)	5.17E+00(+)	**1.19E+00**
f19	1.89E+01/	1.19E+01/	6.24E+02/	1.49E+01/	7.52E+00/	2.37E+01/	7.07E+00/	**6.51E+00/**
	2.45E+00(+)	3.69E+00(+)	1.86E+03(+)	5.33E+00(+)	2.24E+00(≈)	2.39E+01(+)	1.75E+00(≈)	**1.89E+00**
f20	1.77E+02/	1.23E+02/	8.97E+01/	1.11E+02/	1.64E+01/	9.65E+01/	7.21E+01/	**6.05E+01/**
	7.87E+01(+)	6.72E+01(+)	4.61E+01(+)	3.57E+01(+)	2.36E+01(-)	8.04E+01(≈)	8.98E+00(+)	**1.66E+01**
f21	2.47E+02/	2.51E+02/	2.27E+02/	1.50E+02/	3.62E+02/	2.38E+02/	1.50E+02/	**1.00E+02/**
	6.90E+00(+)	6.53E+00(+)	4.68E+00(+)	1.85E-13(+)	9.13E+00(+)	8.89E+00(+)	2.89E-14(+)	**0.00E+00**
f22	3.04E+03/	1.00E+02/	1.00E+02/	1.50E+02/	7.23E+02/	7.09E+02/	1.50E+02/	**1.00E+02/**
	1.64E+03(+)	0.00E+00(≈)	0.00E+00(≈)	9.41E-14(+)	1.90E+03(≈)	1.59E+03(+)	2.89E-14(+)	**0.00E+00**
f23	4.00E+02/	3.89E+02/	3.74E+02/	5.07E+02/	4.82E+02/	3.92E+02/	5.00E+02/	**2.00E+02/**
	7.54E+00(+)	6.37E+00(+)	4.96E+00(+)	6.97E+00(+)	5.60E+01(+)	1.03E+01(+)	4.26E+00(+)	**0.00E+00**
f24	4.76E+02/	4.60E+02/	4.41E+02/	6.97E+02/	5.73E+02/	4.63E+02/	9.66E+02/	**2.00E+02/**
	1.07E+01(+)	8.13E+00(+)	5.89E+00(+)	3.10E+02(+)	3.49E+01(+)	1.05E+01(+)	3.72E+00(+)	**0.00E+00**
f25	3.78E+02/	3.87E+02/	3.87E+02/	4.14E+02/	3.87E+02/	3.87E+02/	4.05E+02/	**2.00E+02/**
	1.42E-01(+)	1.52E-01(+)	1.72E-01(+)	1.73E+01(+)	6.01E-02(+)	1.03E+00(+)	6.70E+00(+)	**0.00E+00**
f26	1.02E+03/	1.34E+03/	1.19E+03/	1.85E+03/	1.47E+03/	1.43E+03/	1.91E+03/	**2.00E+02/**
	4.81E+02(+)	1.27E+02(+)	6.78E+01(+)	4.26E+02(+)	6.09E+02(+)	1.25E+02(+)	2.13E+01(+)	**0.00E+00**
f27	5.00E+02/	5.01E+02/	5.05E+02/	7.53E+02/	4.97E+02/	5.10E+02/	7.39E+02/	**2.00E+02/**
	1.13E-04(+)	6.40E+00(+)	5.59E+00(+)	3.40E+01(+)	9.15E+00(+)	8.33E+00(+)	1.74E+01(+)	**0.00E+00**
f28	5.00E+02/	3.57E+02/	3.50E+02/	4.42E+02/	3.46E+02/	3.59E+02/	3.84E+02/	**2.00E+02/**
	7.35E-01(+)	5.98E+01(+)	5.96E+01(+)	4.41E+01(+)	6.29E+01(+)	7.09E+01(+)	3.42E+01(+)	**0.00E+00**
f29	4.67E+02/	5.19E+02/	4.79E+02/	3.53E+02/	4.28E+02/	5.29E+02/	3.21E+02/	**2.00E+02/**
	8.26E+01(+)	3.37E+01(+)	2.09E+01(+)	2.04E+01(+)	3.28E+01(+)	8.84E+01(+)	1.09E+01(+)	**0.00E+00**
f30	2.19E+02/	2.15E+03/	2.13E+03/	1.03E+03/	2.03E+03/	2.21E+03/	9.34E+02/	**2.00E+02/**
	5.93E+00(+)	1.24E+02(+)	1.44E+02(+)	1.40E+02(+)	7.49E+01(+)	2.77E+02(+)	6.01E+01(+)	**0.00E+00**
+	27	26	26	29	24	28	17	/
≈	0	3	3	1	5	2	9	/
-	3	1	1	0	1	0	4	/
Rank	5.25	5.20	5.12	5.02	5.27	6.20	3.33	**1.63**

TABLE 4. Numerical and statistic results of MWADE and 7 typical DE variants on IEEE CEC2017 test suite when D = 50.

Algorithms	EPSDE	jDE	JADE	SHADE	FDDE	rank-DE	LSHADE	MWADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f1	5.62E-09/ 9.70E-09(+)	9.38E-07/ 4.83E-06(+)	5.83E-14/ 5.07E-14(+)	4.88E-14/ 2.07E-14(+)	1.16E-04/ 3.03E-04(+)	1.35E-07/ 3.82E-07(+)	1.89E-14/ 7.77E-15(≈)	**1.85E-14/ 7.60E-15**
f2	4.06E+11/ 8.11E+10(+)	5.10E+12/ 2.47E+13(+)	1.15E+12/ 3.36E+12(+)	6.38E+14/ 3.49E+15(+)	2.34E+11/ 9.43E+11(+)	1.69E+15/ 9.20E+15(+)	7.33E-01/ 1.20E+00(≈)	**3.33E-01/ 6.06E-01**
f3	2.11E+05/ 2.16E+05(+)	1.53E+00/ 2.34E+00(+)	2.56E+04/ 4.10E+04(+)	3.15E-13/ 7.57E-14(+)	1.52E+04/ 4.62E+03(+)	7.52E+00/ 9.98E+00(+)	**1.48E-13/ 3.84E-14(≈)**	**1.72E-13/ 7.83E-14**
f4	2.87E+01/ 4.04E+00(≈)	7.65E+01/ 4.64E+01(+)	5.20E+01/ 5.02E+01(≈)	**2.53E+01/ 4.28E+01(≈)**	7.12E+01/ 4.98E+01(≈)	5.04E+01/ 4.23E+01(≈)	6.42E+01/ 4.54E+01(≈)	5.48E+01/ 4.71E+01
f5	2.17E+01/ 1.04E-01(-)	9.84E+01/ 9.59E+00(+)	5.19E+01/ 7.03E+00(+)	3.29E+01/ 5.59E+00(≈)	3.41E+02/ 1.61E+01(+)	7.30E+01/ 1.53E+01(+)	**1.15E+01/ 2.06E+00(-)**	4.53E+01/ 1.30E+01
f6	**1.17E-13/ 2.08E-14(-)**	**1.17E-13/ 2.08E-14(-)**	1.52E-13/ 6.21E-14(-)	8.26E-04/ 8.26E-04(+)	1.13E-04/ 5.65E-04(+)	1.02E-02/ 2.98E-02(+)	5.64E-08/ 1.30E-07(-)	2.60E-05/ 1.41E-04
f7	2.02E+02/ 1.47E+01(+)	1.49E+02/ 1.06E+01(+)	1.03E+02/ 7.92E+00(+)	8.25E+01/ 5.16E+00(≈)	3.96E+02/ 1.46E+01(+)	1.56E+02/ 6.89E+01(+)	**6.33E+01/ 1.93E+00(-)**	8.74E+01/ 1.11E+01
f8	1.58E+02/ 1.94E+01(+)	9.89E+01/ 1.03E+01(+)	5.20E+01/ 6.73E+00(+)	3.30E+01/ 6.28E+00(≈)	3.34E+02/ 2.26E+01(+)	8.41E+01/ 1.32E+01(+)	**1.11E+01/ 2.24E+00(-)**	4.10E+01/ 1.23E+01
f9	4.77E+02/ 5.24E+02(+)	5.74E-02/ 1.53E-01(+)	9.61E-01/ 9.95E-01(+)	9.57E-01/ 8.61E-01(+)	5.20E-01/ 5.64E-01(+)	1.29E+01/ 8.65E+00(+)	**3.03E-14/ 5.11E-14(-)**	2.98E-03/ 1.63E-02
f10	9.01E+03/ 8.59E+02(+)	4.95E+03/ 3.93E+02(+)	3.75E+03/ 2.61E+02(≈)	3.18E+03/ 2.87E+02(≈)	1.30E+04/ 3.32E+02(+)	9.09E+03/ 1.08E+03(+)	**3.04E+03/ 3.72E+02(-)**	4.26E+03/ 1.25E+03
f11	5.63E+01/ 1.59E+01(≈)	5.19E+01/ 1.52E+01(≈)	1.33E+02/ 2.59E+01(+)	2.19E+02/ 6.29E+01(+)	6.19E+01/ 4.49E+01(≈)	1.44E+02/ 6.20E+01(+)	**4.72E+01/ 1.05E+01(≈)**	4.96E+01/ 1.05E+01
f12	1.41E+01/ 3.47E+00(-)	5.89E+04/ 4.83E+04(+)	3.99E+03/ 1.74E+03(+)	2.51E+03/ 4.68E+02(≈)	6.30E+04/ 4.03E+04(+)	4.44E+04/ 3.04E+04(+)	2.52E+03/ 5.56E+02(≈)	**2.41E+03/ 5.09E+02**
f13	2.51E+01/ 4.17E+00(-)	2.54E+03/ 4.38E+03(+)	3.27E+02/ 2.55E+02(+)	2.42E+02/ 1.82E+02(+)	8.52E+02/ 1.27E+03(+)	4.60E+02/ 6.92E+02(+)	7.93E+01/ 3.34E+01(+)	**4.99E+01/ 2.27E+01**
f14	5.50E+02/ 2.72E+02(+)	6.57E+01/ 4.75E+01(+)	6.09E+03/ 3.20E+04(+)	1.26E+02/ 4.70E+01(+)	1.20E+02/ 1.55E+01(+)	3.97E+02/ 5.43E+02(+)	**3.16E+01/ 3.98E+00(-)**	3.81E+01/ 6.42E+00
f15	5.56E+03/ 9.89E+03(+)	1.31E+02/ 2.40E+02(≈)	5.21E+02/ 6.53E+02(+)	2.15E+02/ 5.40E+01(+)	7.69E+01/ 3.85E+01(≈)	5.02E+02/ 1.46E+03(+)	6.18E+01/ 1.80E+01(≈)	**6.14E+01/ 2.28E+01**
f16	**2.51E+01/ 3.64E-01(-)**	9.62E+02/ 1.84E+02(≈)	8.36E+02/ 2.08E+02(≈)	5.80E+02/ 1.67E+02(≈)	1.21E+03/ 8.99E+02(≈)	1.07E+03/ 2.72E+02(+)	3.00E+02/ 1.11E+02(-)	8.41E+02/ 3.01E+02
f17	8.94E+02/ 1.48E+02(+)	6.94E+02/ 1.73E+02(+)	5.90E+02/ 1.27E+02(+)	4.85E+02/ 1.23E+02(+)	7.77E+02/ 3.71E+02(+)	5.01E+02/ 1.98E+02(+)	**1.78E+02/ 3.94E+01(-)**	4.09E+02/ 1.71E+02
f18	7.74E+03/ 9.07E+03(+)	3.00E+03/ 2.82E+03(+)	2.00E+02/ 1.04E+02(+)	2.41E+02/ 9.07E+01(+)	5.99E+02/ 4.36E+02(+)	1.39E+04/ 1.29E+04(+)	9.80E+01/ 2.87E+01(+)	**8.53E+01/ 2.77E+01**
f19	5.14E+01/ 8.35E+00(+)	**2.47E+01/ 7.49E+00(-)**	5.94E+02/ 1.84E+03(+)	1.93E+02/ 8.25E+01(+)	2.64E+01/ 1.37E+01(-)	9.52E+01/ 4.76E+01(+)	2.70E+01/ 3.91E+00(≈)	2.81E+01/ 4.92E+00
f20	5.79E+02/ 1.76E+02(+)	5.80E+02/ 1.51E+02(+)	4.08E+02/ 1.40E+02(≈)	3.07E+02/ 9.99E+01(≈)	6.66E+02/ 3.08E+02(+)	4.13E+02/ 2.60E+02(≈)	**1.71E+02/ 2.72E+01(-)**	4.07E+02/ 1.89E+02
f21	3.54E+02/ 2.03E+01(+)	3.04E+02/ 1.08E+01(+)	2.54E+02/ 5.78E+00(+)	1.50E+02/ 2.76E-13(+)	5.26E+02/ 5.53E+01(+)	2.82E+02/ 1.44E+01(+)	1.50E+02/ 1.97E-13(+)	**1.50E+02/ 0.00E+00**
f22	9.35E+03/ 6.99E+02(+)	4.69E+03/ 2.11E+03(+)	4.00E+03/ 1.34E+03(+)	1.50E+02/ 2.29E-13(+)	1.26E+04/ 2.38E+03(+)	8.89E+03/ 2.17E+03(+)	1.50E+02/ 9.41E-14(+)	**1.50E+02/ 0.00E+00**
f23	5.56E+02/ 2.59E+01(+)	5.13E+02/ 1.17E+01(+)	4.78E+02/ 9.84E+00(+)	7.44E+02/ 1.46E+01(+)	7.32E+02/ 8.16E+01(+)	5.09E+02/ 1.86E+01(+)	7.35E+02/ 1.81E+01(+)	**2.00E+02/ 0.00E+00**
f24	6.56E+02/ 2.57E+01(+)	5.75E+02/ 1.41E+01(+)	5.43E+02/ 9.57E+00(+)	8.62E+02/ 3.84E+02(+)	8.28E+02/ 4.72E+01(+)	5.87E+02/ 1.89E+01(+)	1.33E+03/ 1.02E+01(+)	**2.00E+02/ 0.00E+00**
f25	4.42E+02/ 1.62E+01(+)	5.15E+02/ 3.92E+01(+)	5.19E+02/ 3.21E+01(+)	5.09E+02/3 .10E+01(+)	5.12E+02/ 3.65E+01(+)	5.26E+02/ 3.96E+01(+)	5.02E+02/ 2.73E+01(+)	**2.00E+02/ 0.00E+00**
f26	2.02E+03/ 1.02E+03(+)	2.00E+03/ 1.20E+02(+)	1.63E+03/ 1.12E+02(+)	1.32E+03/ 1.49E+03(+)	1.88E+03/ 9.92E+02(+)	2.12E+03/ 2.16E+02(+)	3.15E+03/ 5.73E+01(+)	**2.00E+02/ 0.00E+00**
f27	5.00E+02/ 1.25E-04(+)	5.37E+02/ 1.66E+01(+)	5.56E+02/ 2.09E+01(+)	9.59E+02/ 6.28E+01(+)	5.28E+02/ 1.52E+01(+)	6.21E+02/ 6.81E+01(+)	8.80E+02/ 3.06E+01(+)	**2.00E+02/ 0.00E+00**
f28	5.00E+02/ 1.35E-04(+)	4.90E+02/ 2.29E+01(+)	4.94E+02/ 1.99E+01(+)	5.10E+02/ 4.51E+01(+)	4.73E+02/ 2.23E+01(+)	4.93E+02/ 1.84E+01(+)	5.37E+02/ 4.27E+01(+)	**2.00E+02/ 0.00E+00**
f29	9.10E+02/ 1.60E+02(+)	5.21E+02/ 7.90E+01(+)	4.81E+02/ 7.58E+01(+)	7.83E+02/ 1.27E+02(+)	3.35E+02/ 2.14E+01(+)	7.10E+02/ 2.35E+02(+)	5.74E+02/ 3.96E+01(+)	**2.00E+02/ 0.00E+00**
f30	1.06E+03/ 1.40E+03(+)	6.09E+05/ 3.99E+04(+)	6.63E+05/ 8.05E+04(+)	6.03E+03/ 4.33E+02(+)	6.17E+05/ 3.44E+04(+)	6.90E+05/ 1.01E+05(+)	5.79E+03/ 4.02E+02(+)	**2.00E+02/ 0.00E+00**
+	23	25	25	22	25	28	12	/
≈	2	2	1	0	1	2	8	/
-	5	3	4	8	4	0	10	/
Rank	4.95	4.95	4.73	4.33	5.77	5.97	3.10	**2.20**

TABLE 5. Numerical and statistic results of MWADE and 7 typical DE variants on IEEE CEC2017 test suite when D = 100.

Algorithms	EPSDE	jDE	JADE	SHADE	FDDE	rank-DE	LSHADE	MWADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f1	5.44E-04/ 2.70E-03(+)	1.40E-10/ 3.63E-10(+)	3.57E-10/ 5.59E-10(+)	2.35E-10/ 4.91E-10(+)	1.62E+03/ 2.15E+03(+)	1.94E-07/ 6.14E-07(+)	**5.01E-13/ 7.74E-13(-)**	1.10E-11/ 1.46E-11
f2	8.29E+11/ 9.11E+10(+)	3.19E+31/ 1.62E+32(+)	2.01E+50/ 1.10E+51(+)	2.25E+43/ 1.04E+44(+)	5.53E+50/ 2.92E+51(+)	1.85E+66/ 1.01E+67(+)	4.37E+12/ 1.97E+13(+)	**2.46E+06/ 8.41E+06**
f3	9.01E+05/ 4.63E+05(+)	2.38E+03/ 6.34E+03(+)	1.45E+05/ 1.72E+05(+)	1.55E+04/ 8.50E+04(+)	2.30E+05/ 2.65E+04(+)	1.53E+04/ 5.63E+03(+)	**3.24E-06/ 3.58E-06(-)**	9.47E-05/ 8.56E-05
f4	1.32E+02/ 4.45E+01(-)	2.07E+02/ 1.44E+01(+)	**7.83E+01/ 6.84E+01(-)**	1.07E+02/ 6.00E+01(-)	1.93E+02/ 4.54E+01(+)	1.60E+02/ 6.50E+01(-)	1.85E+02/ 3.15E+01(+)	1.77E+02/ 3.19E+01
f5	**2.17E+01/ 5.64E-02(-)**	2.50E+02/ 2.53E+01(+)	1.47E+02/ 1.55E+01(+)	1.09E+02/ 1.27E+01(+)	7.92E+02/ 1.29E+02(+)	2.36E+02/ 4.07E+01(+)	2.64E+01/ 6.57E+00(-)	9.22E+01/ 2.34E+01
f6	**2.31E-13/ 2.08E-14(-)**	7.03E-09/ 2.37E-08(-)	2.98E-04/ 8.48E-04(-)	1.52E-02/ 1.41E-02(+)	2.50E-02/ 2.34E-02(+)	3.52E-01/ 3.42E-01(+)	1.51E-03/ 1.79E-03(+)	5.62E-04/ 7.65E-04
f7	7.41E+02/ 3.56E+01(+)	3.53E+02/ 1.96E+01(+)	2.70E+02/ 2.39E+01(+)	2.15E+02/ 1.28E+01(+)	9.45E+02/ 2.35E+01(+)	3.73E+02/ 4.58E+01(+)	**1.38E+02/ 4.03E+00(-)**	1.73E+02/ 1.78E+01
f8	6.22E+02/ 4.22E+01(+)	2.47E+02/ 2.22E+01(+)	1.46E+02/ 1.54E+01(+)	1.11E+02/ 1.38E+01(+)	8.12E+02/ 2.39E+01(+)	2.41E+02/ 4.69E+01(+)	**2.93E+01/ 6.59E+00(-)**	8.21E+01/ 2.33E+01
f9	2.88E+03/ 1.12E+03(+)	5.31E+00/ 7.21E+00(+)	7.88E+01/ 6.99E+01(+)	1.48E+01/ 8.50E+00(+)	3.02E+01/ 3.82E+01(+)	4.22E+02/ 2.45E+02(+)	**2.71E-02/ 8.64E-02(-)**	5.12E-02/ 1.16E-01
f10	2.56E+04/ 1.04E+03(+)	1.28E+04/ 6.51E+02(+)	1.01E+04/ 5.48E+02(-)	9.42E+03/ 5.48E+02(-)	2.95E+04/ 4.96E+02(+)	2.31E+04/ 5.49E+02(+)	**1.04E+04/ 5.55E+02(-)**	1.28E+04/ 1.16E+03
f11	5.57E+02/ 3.08E+02(+)	3.16E+02/ 7.33E+01(+)	3.23E+03/ 3.34E+03(+)	7.50E+02/ 1.61E+02(+)	3.69E+02/ 2.43E+02(+)	6.88E+02/ 1.95E+02(+)	3.00E+02/ 3.67E+01(+)	**2.21E+02/ 3.80E+01**
f12	**1.10E+01/ 1.91E+00(-)**	1.96E+05/ 7.71E+04(+)	2.68E+04/ 3.27E+04(+)	5.50E+03/ 8.91E+02(-)	2.47E+05/ 9.23E+04(+)	3.05E+05/ 2.34E+05(+)	5.14E+03/ 9.12E+02(-)	6.23E+03/ 1.77E+03
f13	**1.32E+01/ 9.83E-01(-)**	3.00E+03/ 2.67E+03(+)	4.72E+03/ 4.50E+03(+)	2.18E+03/ 9.11E+02(+)	2.44E+03/ 2.71E+03(+)	9.15E+02/ 7.43E+02(+)	5.04E+02/ 1.55E+02(+)	2.47E+02/ 7.41E+01
f14	1.43E+03/ 1.76E+02(+)	1.69E+04/ 1.41E+04(+)	6.34E+02/ 1.76E+02(+)	5.64E+02/ 1.45E+02(+)	1.19E+04/ 1.19E+04(+)	1.81E+04/ 1.70E+04(+)	**3.20E+01/ 4.44E+00(-)**	3.46E+01/ 5.14E+00
f15	1.28E+03/ 1.63E+03(+)	1.33E+03/ 1.76E+03(+)	3.71E+02/ 1.07E+02(+)	4.46E+02/ 1.33E+02(+)	3.52E+03/ 4.99E+03(+)	1.51E+03/ 1.97E+03(+)	3.78E+02/ 8.58E+01(+)	**1.47E+02/ 4.02E+01**
f16	**5.01E+01/ 6.28E-01(-)**	3.00E+03/ 3.64E+02(+)	2.65E+03/ 3.80E+02(+)	2.52E+03/ 2.48E+02(-)	7.01E+03/ 1.15E+03(+)	2.77E+03/ 5.25E+02(+)	1.64E+03/ 2.48E+02(-)	2.62E+03/ 4.62E+02
f17	2.78E+03/ 4.08E+02(+)	2.29E+03/ 2.64E+02(+)	1.88E+03/ 2.90E+02(+)	1.70E+03/ 3.11E+02(+)	3.13E+03/ 1.03E+03(+)	2.09E+03/ 3.57E+02(+)	**5.87E+02/ 1.16E+02(-)**	1.17E+03/ 5.67E+02
f18	4.42E+05/ 9.30E+05(+)	6.43E+04/ 4.23E+04(+)	1.43E+03/ 7.09E+02(+)	1.69E+03/ 1.28E+03(+)	8.21E+04/ 3.64E+04(+)	1.18E+05/ 5.75E+04(+)	3.03E+02/ 5.17E+01(+)	**2.77E+02/ 6.81E+01**
f19	1.84E+03/ 2.00E+03(+)	1.96E+03/ 2.63E+03(+)	7.85E+02/ 7.09E+02(+)	9.67E+02/ 1.14E+03(+)	4.51E+03/ 5.51E+03(+)	2.81E+02/ 7.62E+01(+)	4.24E+02/ 5.43E+01(+)	1.47E+02/ 9.36E+01
f20	2.51E+03/ 3.34E+02(+)	2.30E+03/ 2.36E+02(+)	1.85E+03/ 2.42E+02(+)	1.50E+03/ 1.47E+02(-)	3.42E+03/ 9.99E+02(+)	1.82E+03/ 4.76E+02(+)	**9.67E+02/ 1.75E+02(-)**	1.68E+03/ 4.55E+02
f21	8.35E+02/ 4.29E+01(+)	4.78E+02/ 2.35E+01(+)	3.68E+02/ 1.78E+01(+)	1.50E+02/ 2.95E-13(+)	1.05E+03/ 2.51E+01(+)	4.69E+02/ 3.37E+01(+)	1.50E+02/ 0.00E+00(+)	**1.50E+02/ 0.00E+00**
f22	2.67E+04/ 1.13E+03(+)	1.43E+04/ 5.48E+02(+)	1.13E+04/ 4.81E+02(+)	1.50E+02/ 2.11E-13(+)	3.03E+04/ 5.19E+02(+)	2.26E+04/ 4.09E+03(+)	**1.50E+02/ 0.00E+00(≈)**	**1.50E+02/ 0.00E+00**
f23	9.99E+02/ 2.74E+01(+)	6.62E+02/ 1.17E+01(+)	6.47E+02/ 1.36E+01(+)	1.24E+03/ 3.44E+01(+)	6.49E+02/ 2.38E+01(+)	7.56E+02/ 3.43E+01(+)	1.19E+03/ 3.37E+01(+)	**2.00E+02/ 0.00E+00**
f24	1.53E+03/ 5.13E+01(+)	1.09E+03/ 2.60E+01(+)	1.03E+03/ 2.85E+01(+)	2.69E+02/ 2.88E+02(+)	9.82E+02/ 1.93E+01(+)	1.13E+03/ 4.36E+01(+)	2.77E+03/ 1.74E+01(+)	**2.00E+02/ 0.00E+00**
f25	7.78E+02/ 7.62E+01(+)	7.61E+02/ 5.73E+01(+)	7.45E+02/ 4.92E+01(+)	7.77E+02/ 5.95E+01(+)	7.50E+02/ 4.44E+01(+)	7.73E+02/ 6.13E+01(+)	6.99E+02/ 2.06E+01(+)	**2.00E+02/ 0.00E+00**
f26	5.22E+03/ 3.70E+03(+)	5.43E+03/ 2.88E+02(+)	4.58E+03/ 2.45E+02(+)	4.67E+03/ 4.79E+03(+)	4.23E+03/ 2.72E+02(+)	5.95E+03/ 4.35E+02(+)	7.66E+03/ 1.25E+02(+)	**2.00E+02/ 0.00E+00**
f27	5.00E+02/ 1.11E-04(+)	6.94E+02/ 2.95E+01(+)	7.20E+02/ 3.88E+01(+)	1.85E+03/ 1.85E+02(+)	6.30E+02/ 2.96E+01(+)	7.92E+02/ 6.42E+01(+)	1.62E+03/ 9.64E+01(+)	**2.00E+02/ 0.00E+00**
f28	5.00E+02/ 1.54E-01(+)	5.67E+02/ 3.08E+01(+)	5.32E+02/ 5.45E+01(+)	4.96E+02/ 1.49E+01(+)	5.47E+02/ 2.60E+01(+)	5.48E+02/ 3.88E+01(+)	5.07E+02/ 1.45E+01(+)	**2.00E+02/ 0.00E+00**
f29	2.68E+03/ 3.09E+02(+)	2.13E+03/ 2.71E+02(+)	2.21E+03/ 3.41E+02(+)	1.79E+03/ 2.10E+02(+)	1.45E+03/ 8.35E+02(+)	2.69E+03/ 4.71E+02(+)	1.01E+03/ 1.28E+02(+)	**2.00E+02/ 0.00E+00**
f30	5.07E+02/ 9.31E+02(+)	4.13E+03/ 2.04E+03(+)	3.96E+03/ 2.47E+03(+)	4.42E+03/ 2.28E+03(+)	4.47E+03/ 2.41E+03(+)	3.25E+03/ 4.38E+02(+)	3.64E+03/ 1.77E+02(+)	**2.00E+02/ 0.00E+00**
+	24	29	27	25	30	29	17	/
≈	0	0	0	0	0	0	1	/
-	6	1	3	5	0	1	11	/
Rank	5.14	5.36	4.31	4.03	6.31	5.79	3.00	**2.05**

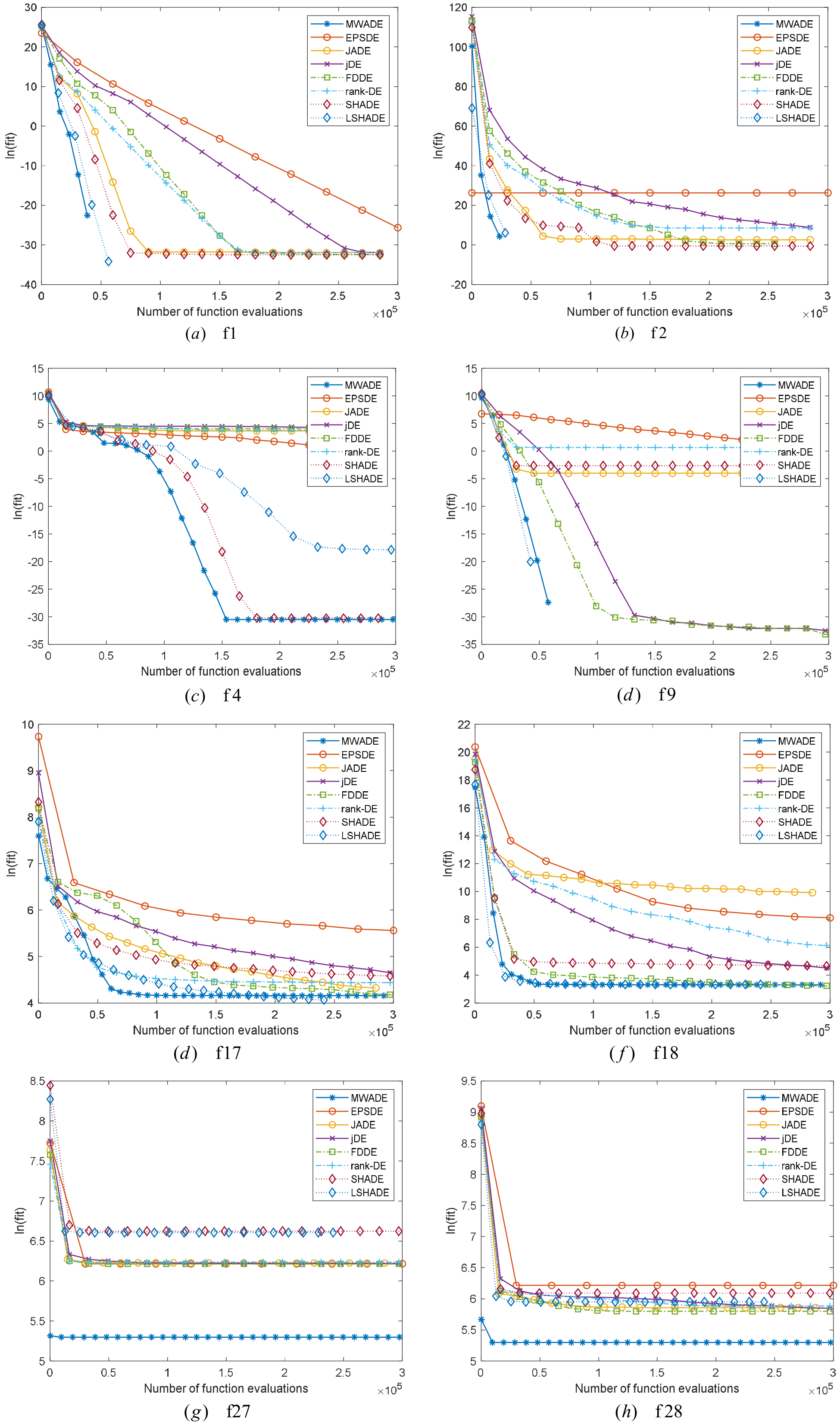

FIGURE 2. The evolution curves of MWADE and 7 typical DE variants on (a) f1, (b) f2, (c) f4, (d) f9, (e) f17, (f) f18, (g) f27 and (h) f28 with D=30.

TABLE 6. Comparison results of MWADE and 7 typical DE variants on IEEE CEC2017 test suite based on Wilcoxon signed-rank tests.

Comparisons	D=30				D=50				D=100			
	R+	R-	p-value	$\alpha = 0.05$	R+	R-	p-value	$\alpha = 0.05$	R+	R-	p-value	$\alpha = 0.05$
MWADE VS EPSDE	424	41	0.000	Yes	400	65	0.001	Yes	404	61	0.000	Yes
MWADE VS jDE	404	2	0.000	Yes	457	8	0.000	Yes	433	2	0.000	Yes
MWADE VS JADE	414	21	0.000	Yes	430	35	0.000	Yes	403	32	0.000	Yes
MWADE VS SHADE	407	28	0.000	Yes	325	80	0.005	Yes	310	68	0.004	Yes
MWADE VS FDDE	453	12	0.000	Yes	461	4	0.000	Yes	435	0	0.000	Yes
MWADE VS rank-DE	465	0	0.000	Yes	462	3	0.000	Yes	432	3	0.000	Yes
MWADE VS LSHADE	245	80	0.026	Yes	248	158	0.305	NO	266	140	0.151	NO

on 29, 30, 30, 27, 28 and 29 functions separately, and owns the best rank among them. So, MWADE shows better performance than them in these cases.

Furthermore, to further demonstrate the differences between MWADE and the chosen non-DE approaches, Table 9 also reports their comparison results based on Wilcoxon signed-rank tests with both D=30 and D=50. From Table 9, we can see that the significant differences between MWADE and its each compared method can always be found in all cases with D=50 and D=100 at significant level $\alpha = 0.05$, and higher R+ is always obtained by MWADE. Thereby, there is a significant difference between MWADE and its comparison algorithms. Thus, MWADE has a more promising performance than these compared methods.

D. EFFECTIVENESS OF THE PROPOSED STRATEGIES

To demonstrate the effectiveness of the proposed components in MWADE, four variants of MWADE, named MWADE-1, MWADE-2, MWADE-3 and MWADE-4, are designed and compared with MWADE on 30 problems from IEEE CEC2017 test suite [42] when D=30. In particular, MWADE-1 is the version of MWADE, where a single mutation operator, named DE/current-to-pbest/1, is just adopted to create the mutant individuals for all solutions, which is proposed in JADE [23]. MWADE-2 and MWADE-3 are the variants of MWADE, where the weighted control parameter setting and the random opposition learning mechanism are removed respectively. Meanwhile, MWADE-4 does not use the proposed adaptive population reduction mechanism, but let the population size be a fixed number (i.e., 100) during the whole evolutionary process. So, these variants can availably verify the benefits of the four developed strategies. Table 10 reports the numerical and statistical results of MWADE and its four variants based on Wilcoxon rank sum test and Friedman test.

From Table 10, it can be found that MWADE has a significantly better performance than its variants. In detail, according to Wilcoxon rank sum test, MWDAE gets the better results than MWADE-1, MWADE-2, MWADE-3 and MWADE-4 on 22, 24, 27 and 22 functions respectively, and owns the worse performance on 6, 0, 2 and 6 functions, respectively. Moreover, from the statistical results of Friedman test reported in Table 10, MWADE and MWADE-1, MWADE-2, MWADE-3 and MWADE-4 obtains 1.6, 2.97, 3.73, 3.0 and 3.7 in term of rank on the all problems, respectively. On the other hand, in order to further show the difference between MWADE and its four variants, Table 11 also lists the comparison results of MWADE and its four variants based on Wilcoxon signed-rank tests. From Table 11, one can see that MWADE always has larger R+ values in all cases, and there are significant divergences between MWADE and each its variant. So, the proposed methods can positively promote the performance improvement of algorithm. This might be due to the facts that the multi-schemes mutation strategy and the random opposition learning mechanism are helpful to balance the exploration and exploitation, and alleviate the risk of trapping into the local optima, respectively. Meanwhile, the weighted control parameter setting can effectively enlarge the exploration ability of algorithm, and the new adaptive population reduction scheme is capable of further improving the search efficiency of algorithm. Therefore, the proposed four strategies can effectively strengthen the performance of algorithm.

E. REAL APPLICATION

In this part, we further apply MWADE to the car side impact design problem [53], [54] to test its performance, which is on the foundation of the European Enhanced Vehicle-Safety Committee (EEVC). The aim of this problem is to minimize the weight of the door, which involves 11 parameters including thickness of the inner B-pillar plate x_1, B-pillar reinforcement x_2, thickness of the inner floor x_3, cross member x_4, door beam x_5, door beltline reinforcement x_6, roof longitudinal beam x_7, inner B-pillar x_8, inner floor x_9, guardrail height x_{10}, and crash position x_{11}. In detail, this problem can be mathematically modeled

TABLE 7. Numerical and statistic results of MWADE and 6 non-DE heuristic algorithms on IEEE CEC2017 test suite when D = 30.

Algorithms	TAPSO	HSOGA	HSSOGSA	EPSO	HSJOA	ExPSO	MWADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f1	9.06E+02/ 1.81E+03(+)	7.03E+03/ 4.26E+03(+)	4.43E+03/ 3.19E+03(+)	5.36E+00/ 5.51E+00(+)	4.01E+03/ 6.31E+03(+)	2.75E+03/ 4.57E+03(+)	**0.00E+00/ 0.00E+00**
f2	1.45E+33/ 7.77E+33(+)	2.74E+18/ 5.58E+18(+)	2.35E+38/ 5.99E+38(+)	1.57E+01/ 5.34E+01(+)	1.06E+16/ 2.09E+16(+)	6.28E+16/ 3.2E+17(+)	**0.00E+00/ 0.00E+00**
f3	7.46E+04/ 2.30E+04(+)	2.27E+03/ 6.57E+02(+)	5.99E+04/ 4.22E+04(+)	2.98E-05/ 1.81E-05(+)	7.95E+04/ 2.30E+04(+)	3.74E-01/ 3.23E-01(+)	**5.68E-15/ 1.73E-14**
f4	5.07E+01/ 2.94E+01(+)	4.75E+02/ 8.76E+00(+)	1.22E+02/ 3.96E+01(+)	1.01E+00/ 4.16E-01(+)	5.14E+01/ 3.61E+01(+)	7.75E+01/ 3.59E+01(+)	**3.60E-14/ 2.79E-14**
f5	1.31E+02/ 6.90E+01(+)	5.50E+02/ 1.37E+01(+)	2.42E+02/ 2.99E+01(+)	6.58E+01/ 4.18E+00(+)	8.49E+01/ 1.97E+01(+)	1.52E+02/ 4.07E+01(+)	**1.15E+01/ 3.89E+00**
f6	2.01E+01/ 7.28E+00(+)	6.00E+02/ 1.36E-03(+)	6.01E+01/ 4.20E+00(+)	1.67E-12/ 5.46E-13(+)	5.48E-01/ 8.00E-01(+)	4.19E+01/ 8.24E+00(+)	**1.48E-13/ 5.30E-14**
f7	2.43E+02/ 8.35E+01(+)	7.83E+02/ 1.27E+01(+)	1.05E+03/ 1.01E+02(+)	1.21E+02/ 6.69E+00(+)	1.26E+02/ 3.11E+01(+)	3.12E+02/ 8.72E+01(+)	**4.26E+01/ 4.09E+00**
f8	1.66E+02/ 6.76E+01(+)	8.43E+02/ 1.21E+01(+)	2.84E+02/ 3.79E+01(+)	4.49E+01/ 5.69E+00(+)	8.07E+01/ 1.95E+01(+)	1.55E+02/ 4.42E+01(+)	**1.04E+01/ 2.92E+00**
f9	2.85E+03/ 1.43E+03(+)	9.05E+02/ 4.31E+00(+)	7.82E+03/ 1.08E+03(+)	1.81E+01/ 6.60E+00(+)	4.57E+02/ 3.73E+02(+)	4.43E+03/ 1.86E+03(+)	**0.00E+00/ 0.00E+00**
f10	6.68E+03/ 4.40E+02(+)	3.55E+03/ 6.78E+02(+)	4.44E+03/ 3.56E+02(-)	**1.78E+03/ 1.88E+02(-)**	2.62E+03/ 4.78E+02(+)	4.72E+03/ 8.86E+02(+)	2.53E+03/ 4.27E+02
f11	1.08E+03/ 8.88E+02(+)	1.19E+03/ 3.41E+01(+)	4.40E+02/ 1.35E+02(+)	5.51E+01/ 8.48E+00(+)	7.75E+01/ 5.41E+01(+)	2.73E+02/ 9.88E+01(+)	**3.89E+00/ 2.43E+00**
f12	4.88E+06/ 2.36E+07(+)	1.06E+05/ 9.45E+04(+)	5.73E+03/ 2.84E+03(+)	2.46E+03/ 4.58E+02(+)	3.39E+04/ 3.57E+04(+)	2.02E+07/ 1.51E+07(+)	**4.05E+02/ 1.95E+02**
f13	5.12E+03/ 6.53E+03(+)	1.46E+03/ 3.99E+01(+)	6.67E+03/ 2.95E+03(+)	9.84E+01/ 2.44E+01(+)	1.30E+03/ 1.31E+03(+)	7.24E+04/ 1.07E+05(+)	**1.66E+01/ 1.15E+00**
f14	8.48E+04/ 1.59E+05(+)	1.47E+03/ 1.67E+01(+)	7.73E+02/ 2.55E+02(+)	1.01E+02/ 2.21E+01(+)	1.17E+03/ 7.88E+02(+)	2.17E+03/ 3.70E+03(+)	**2.19E+01/ 4.05E+00**
f15	2.54E+04/ 9.21E+04(+)	1.65E+03/ 5.37E+01(+)	6.05E+02/ 2.30E+02(+)	7.88E+01/ 1.62E+01(+)	9.45E+02/ 9.77E+02(+)	2.10E+04/ 1.81E+04(+)	**2.95E+00/ 1.36E+00**
f16	1.50E+03/ 3.08E+02(+)	2.23E+03/ 2.25E+02(+)	1.33E+03/ 1.66E+02(+)	4.93E+02/ 1.10E+02(+)	8.46E+02/ 2.58E+02(+)	1.15E+03/ 3.41E+02(+)	**1.07E+02/ 9.77E+01**
f17	6.14E+02/ 2.47E+02(+)	1.83E+03/ 7.26E+01(+)	1.08E+03/ 2.26E+02(+)	1.40E+02/ 2.45E+01(+)	2.32E+02/ 1.47E+02(+)	8.10E+02/ 2.77E+02(+)	**5.45E+01/ 1.40E+01**
f18	1.63E+06/ 1.39E+06(+)	1.68E+04/ 1.03E+04(+)	5.44E+04/ 2.00E+04(+)	3.22E+04/ 1.34E+04(+)	1.46E+05/ 9.79E+04(+)	4.00E+04/ 4.10E+04(+)	**2.21E+01/ 1.19E+00**
f19	6.63E+03/ 9.72E+03(+)	2.24E+03/ 4.67E+02(+)	6.28E+03/ 4.85E+03(+)	8.51E+01/ 3.06E+01(+)	1.71E+03/ 2.50E+03(+)	2.07E+04/ 1.91E+04(+)	**6.51E+00/ 1.89E+00**
f20	7.87E+02/ 1.39E+02(+)	2.31E+03/ 1.50E+02(+)	9.03E+02/ 1.01E+02(+)	2.80E+02/ 4.36E+01(+)	3.30E+02/ 9.01E+01(+)	6.07E+02/ 2.29E+02(+)	**6.05E+01/ 1.66E+01**
f21	**5.01E+01/ 2.89E+01(-)**	2.19E+03/ 1.49E+01(+)	1.32E+02/ 2.64E+01(+)	5.38E+01/ 2.62E+01(-)	1.00E+02/ 2.51E-10(≈)	1.03E+02/ 3.03E+01(+)	1.00E+02/ 0.00E+00
f22	1.64E+02/ 7.27E+01(+)	2.24E+03/ 1.13E+01(+)	2.60E+02/ 2.67E+01(+)	**4.53E+01/ 5.29E+00(-)**	1.00E+02/ 5.50E-12(≈)	1.52E+02/ 3.22E+01(+)	1.00E+02/ 0.00E+00
f23	7.14E+02/ 6.61E+01(+)	2.62E+03/ 1.23E+02(+)	1.19E+03/ 8.72E+01(+)	5.43E+02/ 6.28E+00(+)	4.56E+02/ 1.59E+01(+)	9.84E+02/ 1.79E+02(+)	**2.00E+02/ 0.00E+00**
f24	9.45E+02/ 1.70E+02(+)	2.60E+03/ 1.62E-02(+)	1.40E+03/ 1.23E+02(+)	2.00E+02/ 1.32E-12(+)	7.17E+02/ 2.76E+01(+)	1.41E+03/ 2.92E+02(+)	**2.00E+02/ 0.00E+00**
f25	2.12E+02/ 6.44E+01(+)	2.70E+03/ 1.35E-01(+)	6.11E+02/ 3.94E+01(+)	4.02E+02/ 2.51E+00(+)	4.72E+02/ 5.16E+01(+)	5.07E+02/ 1.16E+02(+)	**2.00E+02/ 0.00E+00**
f26	2.08E+03/ 1.91E+03(+)	2.80E+03/ 1.12E-01(+)	6.10E+03/ 1.55E+03(+)	2.00E+02/ 1.46E-10(+)	1.70E+03/ 2.70E+02(+)	4.16E+03/ 9.89E+02(+)	**2.00E+02/ 0.00E+00**
f27	5.21E+02/ 4.92E+01(+)	3.03E+03 /1.44E+02(+)	5.00E+02/ 6.39E-05(+)	8.82E+02/ 3.61E+01(+)	5.00E+02/ 3.15E-04(+)	1.18E+03/ 2.13E+02(+)	**2.00E+02/ 0.00E+00**
f28	4.10E+02/ 1.41E+02(+)	3.14E+03/ 1.37E+02(+)	5.00E+02/ 8.25E-05(+)	4.58E+02/ 1.72E+01(+)	4.80E+02/ 2.08E+01(+)	1.38E+03/ 1.14E+03(+)	**2.00E+02/ 0.00E+00**
f29	3.07E+02/ 2.62E+02(+)	3.11E+03/ 1.76E+01(+)	1.42E+03/ 2.75E+02(+)	4.60E+02/ 4.66E+01(+)	4.05E+02/ 1.24E+02(+)	1.42E+03/ 3.68E+02(+)	**2.00E+02/ 0.00E+00**
f30	1.85E+05/ 3.93E+05(+)	3.25E+03/ 2.85E+01(+)	5.57E+03/ 3.50E+03(+)	2.54E+03/ 5.87E+02(+)	4.61E+03/ 4.96E+03(+)	2.33E+06/ 7.91E+06(+)	**2.00E+02/ 0.00E+00**
+	29	30	30	27	28	30	/
≈	0	0	0	0	2	0	/
-	1	0	0	3	0	0	/
Rank	4.67	5.83	5.40	2.20	3.48	5.22	**1.20**

TABLE 8. Numerical and statistic results of MWADE and 6 non-DE heuristic algorithms on IEEE CEC2017 test suite when D = 50.

Algorithms	TAPSO	HSOGA	HSSOGSA	EPSO	HSJOA	ExPSO	MWADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f1	1.77E+04/ 3.21E+04(+)	1.61E+06/ 1.23E+06(+)	9.44E+03/ 6.10E+03(+)	4.14E+03/ 1.65E+02(+)	3.66E+03/ 4.73E+03(+)	1.27E+04/ 1.78E+04(+)	**1.85E-14/ 7.60E-15**
f2	5.39E+56/ 2.95E+57(+)	4.51E+36/ 1.55E+37(+)	9.81E+74/ 2.31E+75(+)	9.26E+02/ 2.62E+03(+)	2.14E+32/ 1.17E+33(+)	5.43E+27/ 2.19E+28(+)	**3.33E-01/ 6.06E-01**
f3	1.64E+05/ 4.88E+04(+)	1.54E+04/ 3.13E+03(+)	2.47E+05/ 1.13E+05(+)	4.39E+01/ 2.71E+01(+)	1.97E+05/ 3.44E+04(+)	2.13E+01/ 9.62E+00(+)	**1.72E-13/ 7.83E-14**
f4	1.13E+02/ 3.38E+01(+)	5.87E+02/ 3.13E+01(+)	5.24E+02/ 1.38E+02(+)	9.35E+01/ 9.19E+00(+)	9.59E+01/ 3.14E+01(+)	1.32E+02/ 3.18E+01(+)	**5.48E+01/ 4.71E+01**
f5	2.99E+02/ 1.48E+02(+)	5.99E+02/ 2.90E+01(+)	5.75E+02/ 4.72E+01(+)	1.64E+02/ 9.59E+00(+)	1.91E+02/ 1.93E+01(+)	3.33E+02/ 5.62E+01(+)	**4.53E+01/ 1.30E+01**
f6	3.20E+01/ 1.23E+01(+)	6.00E+02/ 1.13E-03(+)	6.76E+01/ 3.57E+00(+)	**2.76E-08/ 5.56E-08(-)**	1.40E+00/ 8.70E-01(+)	4.84E+01/ 5.90E+00(+)	2.60E-05/ 1.41E-04
f7	5.37E+02/ 1.59E+02(+)	8.70E+02/ 2.40E+01(+)	2.81E+03/ 2.29E+02(+)	2.81E+02/ 1.37E+01(+)	2.97E+02/ 4.32E+01(+)	7.64E+02/ 1.85E+02(+)	**8.74E+01/ 1.11E+01**
f8	3.18E+02/ 1.19E+02(+)	9.06E+02/ 2.37E+01(+)	6.14E+02/ 4.50E+01(+)	1.11E+02/ 1.02E+01(+)	2.04E+02/ 4.75E+01(+)	3.91E+02/ 6.71E+01(+)	**4.10E+01/ 1.23E+01**
f9	1.31E+04/ 5.48E+03(+)	9.51E+02/ 3.39E+01(+)	1.84E+04/ 2.18E+03(+)	2.77E+02/ 8.71E+01(+)	7.21E+03/ 2.16E+03(+)	9.51E+03/ 2.83E+03(+)	**2.98E-03/ 1.63E-02**
f10	1.30E+04/ 6.61E+02(+)	5.93E+03/ 8.03E+02(+)	7.31E+03/ 4.18E+02(+)	4.33E+03/ 2.29E+02(+)	4.92E+03/ 4.73E+02(+)	7.92E+03/ 1.82E+03(+)	**4.26E+03/ 1.25E+03**
f11	1.46E+04/ 1.42E+04(+)	1.31E+03/ 4.29E+01(+)	1.56E+03/ 6.77E+02(+)	2.73E+02/ 2.54E+01(+)	8.72E+01/ 1.78E+01(+)	5.59E+02/ 1.78E+02(+)	**4.96E+01/ 1.05E+01**
f12	1.21E+05/ 3.18E+05(+)	4.72E+05/ 2.93E+05(+)	1.49E+04/ 8.23E+03(+)	4.31E+03/ 4.33E+02(+)	2.69E+04/ 2.27E+04(+)	1.35E+07/ 6.01E+06(+)	**2.41E+03/ 5.09E+02**
f13	5.02E+04/ 5.02E+04(+)	3.36E+04/ 1.26E+04(+)	6.40E+04/ 3.06E+04(+)	8.26E+03/ 4.12E+03(+)	5.63E+04/ 4.83E+04(+)	1.39E+05/ 1.10E+05(+)	**4.99E+01/ 2.27E+01**
f14	1.26E+05/ 3.85E+05(+)	1.65E+03/ 9.02E+01(+)	9.96E+02/ 2.04E+02(+)	1.39E+02/ 2.91E+01(+)	3.83E+02/ 3.23E+02(+)	3.57E+04/ 5.01E+04(+)	**3.81E+01/ 6.42E+00**
f15	4.94E+03/ 4.87E+03(+)	6.63E+03/ 1.84E+03(+)	7.36E+03/ 4.56E+03(+)	3.06E+03/ 1.18E+02(+)	2.73E+03/ 2.42E+03(+)	1.09E+05/ 8.03E+04(+)	**6.14E+01/ 2.28E+01**
f16	2.83E+03/ 9.04E+02(+)	2.67E+03/ 3.66E+02(+)	2.33E+03/ 1.74E+02(+)	1.14E+03/ 1.27E+02(+)	1.54E+03/ 1.69E+02(+)	2.50E+03/ 5.60E+02(+)	**8.41E+02/ 3.01E+02**
f17	1.50E+03/ 3.96E+02(+)	2.38E+03/ 2.55E+02(+)	2.11E+03/ 1.93E+02(+)	7.76E+02/ 1.07E+02(+)	7.81E+02/ 2.10E+02(+)	1.58E+03/ 3.73E+02(+)	**4.09E+02/ 1.71E+02**
f18	6.66E+06/ 4.81E+06(+)	2.78E+05/ 2.71E+05(+)	1.48E+05/ 5.60E+04(+)	2.27E+05/ 2.06E+05(+)	1.09E+06/ 7.69E+05(+)	2.11E+05/ 3.16E+05(+)	**8.53E+01/ 2.77E+01**
f19	5.20E+04/ 1.69E+05(+)	2.32E+03/ 1.04E+03(+)	6.39E+03/ 3.40E+03(+)	2.58E+02/ 9.77E+01(+)	6.25E+03/ 8.03E+03(+)	5.12E+05/ 3.35E+05(+)	**2.81E+01/ 4.92E+00**
f20	1.59E+03/ 2.64E+02(+)	2.85E+03/ 2.66E+02(+)	1.62E+03/ 1.94E+02(+)	9.08E+02/ 9.04E+01(+)	6.17E+02/ 2.53E+02(+)	1.28E+03/ 3.25E+02(+)	**4.07E+02/ 1.89E+02**
f21	1.27E+02/ 4.87E+01(-)	2.35E+03/ 2.42E+01(+)	5.20E+02/ 1.07E+02(+)	**1.09E+02/ 2.44E+01(-)**	1.50E+02/ 2.37E-12(≈)	1.38E+02/ 3.26E+01(-)	1.50E+02/ 0.00E+00
f22	3.28E+02/ 1.35E+02(+)	2.31E+03/ 2.71E+01(+)	5.76E+02/ 4.53E+01(+)	**1.39E+02/ 1.63E+01(-)**	1.50E+02/ 1.45E-12(-)	3.56E+02/ 6.41E+01(+)	1.50E+02/ 0.00E+00
f23	1.27E+03/ 1.37E+02(+)	2.50E+03/ 1.55E-03(+)	2.31E+03/ 1.31E+02(+)	8.83E+02/ 1.93E+01(+)	7.40E+02/ 4.46E+01(+)	2.27E+03/ 4.44E+02(+)	**2.00E+02/ 0.00E+00**
f24	1.65E+03/ 3.08E+02(+)	2.60E+03/ 6.66E-02(+)	2.79E+03/ 1.83E+02(+)	2.00E+02/ 1.72E-10(+)	4.88E+02/ 5.39E+01(+)	2.50E+03/ 6.38E+02(+)	**2.00E+02/ 0.00E+00**
f25	2.11E+02/ 5.72E+01(+)	2.70E+03/ 5.04E-01(+)	7.57E+02/ 4.92E+01(+)	5.18E+02/ 1.81E+01(+)	5.54E+02/ 3.96E+01(+)	5.93E+02/ 5.47E+01(+)	**2.00E+02/ 0.00E+00**
f26	9.36E+02/ 1.97E+03(+)	2.80E+03/ 2.40E-01(+)	1.15E+04/ 1.73E+03(+)	2.23E+02/ 4.30E+01(+)	2.51E+03/ 1.98E+02(+)	7.23E+03/ 3.40E+03(+)	**2.00E+02/ 0.00E+00**
f27	6.97E+02/ 2.89E+02(+)	3.03E+03/ 1.46E+02(+)	5.00E+02/ 4.05E-05(+)	1.51E+03/ 5.81E+01(+)	5.00E+02/ 3.70E-04(+)	2.29E+03/ 4.57E+02(+)	**2.00E+02/ 0.00E+00**
f28	4.59E+02/ 2.89E+02(+)	3.21E+03/ 1.29E+02(+)	5.12E+02/ 6.32E+01(+)	4.79E+02/ 3.01E+01(+)	4.98E+02/ 6.15E+00(+)	9.41E+02/ 1.23E+03(+)	**2.00E+02/ 0.00E+00**
f29	2.75E+02/ 4.13E+02(+)	3.10E+03/ 4.54E+00(+)	2.67E+03/ 3.62E+02(+)	1.21E+03/ 9.24E+01(+)	8.94E+02/ 2.18E+02(+)	2.98E+03/ 6.17E+02(+)	**2.00E+02/ 0.00E+00**
f30	3.00E+04/ 1.63E+05(+)	2.86E+04/ 1.29E+04(+)	5.09E+03/ 2.37E+03(+)	3.02E+04/ 1.05E+04(+)	1.95E+03/ 1.96E+03(+)	4.03E+06/ 2.86E+06(+)	**2.00E+02/ 0.00E+00**
+	29	30	30	27	28	29	/
≈	0	0	0	0	1	0	/
-	1	0	0	3	1	1	/
Rank	4.55	5.79	5.36	2.53	3.29	5.24	**1.22**

TABLE 9. Comparison results of MWADE and 6 non-DE heuristic algorithms on IEEE CEC2017 test suite based on Wilcoxon signed-rank tests.

Comparisons	D=30				D=50			
	R+	R-	p-value	$\alpha = 0.05$	R+	R-	p-value	$\alpha = 0.05$
MWADE VS TAPSO	462	3	0.000	Yes	433	2	0.000	Yes
MWADE VS HSOGA	465	0	0.000	Yes	465	0	0.000	Yes
MWADE VS HSSOGSA	465	0	0.000	Yes	435	0	0.000	Yes
MWADE VS EPSO	362	44	0.000	Yes	427	8	0.005	Yes
MWADE VS HSJOA	406	0	0.000	Yes	406	0	0.000	Yes
MWADE VS ExPSO	465	0	0.000	Yes	464	1	0.000	Yes

TABLE 10. Numerical and statistical results of MWADE and MWADE-1, MWADE-2, MWADE-3 and MWADE-4 on IEEE CEC2017 test suite when D = 30.

Algorithms	MWADE-1	MWADE-2	MWADE-3	MWADE-4	MWADE
NO.	Mean/Std	Mean/Std	Mean/Std	Mean/Std	Mean/Std
f1	1.42E-14/0.00E+00(+)	1.61E-14/4.91E-15(+)	4.74E-16/2.59E-15(+)	3.08E-14/6.55E-15(+)	**0.00E+00/0.00E+00**
f2	5.60E+00/2.82E+01(+)	1.43E+00/6.75E+00(+)	**0.00E+00/0.00E+00(≈)**	1.67E+00/6.73E+00(+)	**0.00E+00/0.00E+00**
f3	6.44E-14/1.97E-14(+)	1.65E-13/4.80E-14(+)	1.89E-14/2.73E-14(+)	3.16E-13/1.03E-13(+)	**5.68E-15/1.73E-14**
f4	6.12E+01/4.60E+01(+)	5.08E+01/4.81E+01(+)	5.49E-14/2.79E-14(+)	6.43E+01/4.50E+01(+)	**3.60E-14/2.79E-14**
f5	1.43E+01/1.95E+00(+)	4.17E+01/7.65E+00(+)	2.58E+01/7.04E+00(+)	4.20E+01/1.36E+01(+)	**1.15E+01/3.89E+00**
f6	4.66E-09/2.32E-08(+)	3.40E-05/1.84E-04(+)	6.84E-09/2.60E-08(+)	1.50E-05/7.79E-05(+)	**1.48E-13/5.30E-14**
f7	6.37E+01/2.13E+00(+)	8.49E+01/1.14E+01(+)	5.33E+01/7.85E+00(+)	8.31E+01/7.25E+00(+)	**4.26E+01/4.09E+00**
f8	1.35E+01/2.60E+00(+)	4.06E+01/1.28E+01(+)	2.68E+01/7.44E+00(+)	3.99E+01/1.51E+01(+)	**1.04E+01/2.92E+00**
f9	3.79E-15/2.08E-14(+)	1.02E-13/3.47E-14(+)	**0.00E+00/0.00E+00(≈)**	1.81E-02/8.40E-02(+)	**0.00E+00/0.00E+00**
f10	2.94E+03/3.08E+02(+)	4.11E+03/1.24E+03(+)	**1.47E+03/4.39E+02(-)**	4.56E+03/1.18E+03(+)	2.53E+03/4.27E+02
f11	5.27E+01/9.80E+00(+)	5.31E+01/1.01E+01(+)	1.28E+01/5.25E+00(+)	5.25E+01/8.46E+00(+)	**3.89E+00/2.43E+00**
f12	2.51E+03/4.33E+02(+)	2.43E+03/6.08E+02(+)	1.01E+03/3.60E+02(+)	2.40E+03/6.11E+02(+)	**4.05E+02/1.95E+02**
f13	7.57E+01/3.17E+01(+)	5.33E+01/2.59E+01(+)	1.91E+01/2.82E+00(+)	5.69E+01/2.65E+01(+)	**1.66E+01/1.15E+00**
f14	3.11E+01/3.21E+00(+)	3.74E+01/6.02E+00(+)	2.68E+01/4.96E+00(+)	3.83E+01/7.63E+00(+)	**2.19E+01/4.05E+00**
f15	6.05E+01/2.04E+01(+)	5.39E+01/2.13E+01(+)	7.32E+00/3.02E+00(+)	4.79E+01/1.45E+01(+)	**2.95E+00/1.36E+00**
f16	3.14E+02/1.10E+02(+)	8.08E+02/3.35E+02(+)	2.74E+02/1.93E+02(+)	8.03E+02/3.28E+02(+)	**1.07E+02/9.77E+01**
f17	2.68E+02/4.73E+01(+)	3.64E+02/1.77E+02(+)	6.78E+01/2.37E+01(+)	4.71E+02/2.23E+02(+)	**5.45E+01/1.40E+01**
f18	9.33E+01/3.09E+01(+)	1.03E+02/3.43E+01(+)	2.61E+01/4.19E+00(+)	9.36E+01/3.02E+01(+)	**2.21E+01/1.19E+00**
f19	2.60E+01/3.83E+00(+)	2.61E+01/4.99E+00(+)	9.12E+00/2.20E+00(+)	2.64E+01/4.42E+00(+)	**6.51E+00/1.89E+00**
f20	1.63E+02/1.92E+01(+)	4.22E+02/1.86E+02(+)	1.03E+02/7.18E+01(+)	4.41E+02/2.12E+02(+)	**6.05E+01/1.66E+01**
f21	**5.17E+01/3.35E+01(-)**	1.50E+02/0.00E+00(+)	1.50E+02/2.89E-14(+)	5.83E+01/3.43E+01(-)	1.00E+02/0.00E+00
f22	**1.27E+01/2.31E+00(-)**	1.50E+02/0.00E+00(+)	1.50E+02/2.89E-14(+)	3.43E+01/1.07E+01(-)	1.00E+02/0.00E+00
f23	2.00E+02/0.00E+00(+)	2.00E+02/0.00E+00(+)	5.34E+02/1.56E+01(+)	2.00E+02/0.00E+00(+)	**2.00E+02/0.00E+00**
f24	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00(≈)**	5.18E+02/2.94E+02(+)	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00**
f25	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00(≈)**	4.11E+02/1.18E+01(+)	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00**
f26	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00(≈)**	1.99E+03/3.36E+02(+)	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00**
f27	2.00E+02/0.00E+00(+)	2.00E+02/0.00E+00(+)	7.35E+02/1.82E+01(+)	2.00E+02/0.00E+00(+)	**2.00E+02/0.00E+00**
f28	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00(≈)**	4.01E+02/2.34E+01(+)	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00**
f29	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00(≈)**	3.52E+02/3.88E+01(+)	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00**
f30	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00(≈)**	9.32E+02/6.12E+01(+)	**2.00E+02/0.00E+00(≈)**	**2.00E+02/0.00E+00**
+	22	24	27	22	/
≈	2	6	1	2	/
-	6	0	2	6	/
Rank	2.97	3.73	3.00	3.70	1.60

TABLE 11. Comparison results of MWADE and MWADE-1, MWADE-2, MWADE-3 and MWADE-4 on IEEE CEC2017 test suite based on Wilcoxon signed-rank tests when D=30.

Comparisons	R+	R-	p-value	$\alpha = 0.05$
MWADE VS MWADE-1	225	28	0.001	Yes
MWADE VS MWADE-2	253	0	0.000	Yes
MWADE VS MWADE-3	379	27	0.000	Yes
MWADE VS MWADE-4	225	28	0.001	Yes

as below [55].

$$min:\ f(x) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_6 \tag{20}$$

$$s.t\begin{cases} g_1(x) = 1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} \\ -0.484x_3x_9 + 0.01343x_6x_{10} - 1 \leq 0 \\ g_2(x) = 46.36 - 9.9x_2 - 12.9x_1x_2 + 0.1107x_3x_{10} \\ -32 \leq 0 \\ g_3(x) = 33.86 + 2.95x_3 + 0.1792x_3 - 5.057x_1x_2 \\ -11.0x_2x_8 - 0.0215x_5x_{10} \\ -9.98x_7x_8 + 22.0x_8x_9 - 32 \leq 0 \\ g_4(x) = -28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} \\ +6.63x_6x_9 - 7.7x_7x_8 \\ +0.32x_9x_{10} - 32 \leq 0 \\ g_5(x) = 0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.019x_2x_7 \\ +0.0144x_3x_5 \\ +0.0008757x_5x_{10} + 0.08045x_6x_9 + 0.00139x_8x_{11} \\ +0.00001575x_{10}x_{11} \leq 0 \\ g_6(x) = 0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 \\ +0.03099x_2x_6 - 0.018x_2x_7 \\ +0.0208x_3x_8 + 0.121x_3x_9 - 0.00364x_5x_6 \\ +0.0007715x_5x_{10} \\ -0.0005354x_6x_{10} + 0.00121x_8x_{11} + 0.00184x_9x_{10} \\ -0.02x_2^2 - 0.32 \leq 0 \\ g_7(x) = 0.74 - 0.61x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} \\ -0.166x_7x_9 + 0.227x_2^2 - 0.32 \leq 0 \\ g_8(x) = 4.72 - 0.5x_4 - 0.19x_2x_3 - 0.0122x_4x_{10} \\ +0.009325x_6x_{10} + 0.000191x_{11}^2 - 4 \leq 0 \\ g_9(x) = 10.58 - 0.674x_1x_2 - 1.95x_2x_8 \\ +0.02054x_3x_{10} \\ -0.0198x_4x_{10} + 0.028x_6x_{10} - 9.9 \leq 0 \\ g_{10}(x) = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 \\ +0.0432x_9x_{10} - 0.0556x_9x_{11} \\ -0.000786x_{11}^2 - 15.7 \leq 0 \end{cases} \tag{21}$$

Herein, $0.5 \leq x_1, x_2, x_3, x_4, x_5, x_6, x_7 \leq 1.5$, $0 \leq x_8, x_9 \leq 1$, $-30 \leq x_{10}, x_{11} \leq 30$.

To solve this problem, we transform this problem into an unconstrained problem by constructing a penalty term. Consequently, the original problem can be described as

$$min: F(X) = f(x) + M \cdot \left(\sum_{i=1}^{10} g_i^2(x) \right), \tag{22}$$

where M is a too large penalty factor, and we set it to 1000 in this paper.

Table 12 reports the numerical results of MWADE and other 7 DE variants, including EPSDE [26], jDE [46], JADE [23], SHADE [34], FDDE [25], rank-DE [24], and LSHADE [39], with 30 independent runs. From Table 12, we can find that MWADE gets the best results compared to its all counterparts. Thus, the proposed MWADE is a more promising optimizer for this practical problem.

TABLE 12. Numerical results of MWADE and other 7 DE variants on the side collision problem of automobile.

Algorithms	Mean	Std
EPSDE	20.98635321	0.191372
jDE	20.89338503	0.189503
JADE	20.99737374	0.170818
SHADE	20.96856533	0.184311
FDDE	20.88082172	0.187373
rank-DE	20.90623041	0.190492
LSHADE	20.95603053	0.187373
MWADE	**20.81814771**	**0.161768**

V. CONCLUSION

This paper proposed an enhanced adaptive differential evolution algorithm with multi-mutation schemes and weighted control parameter setting for solving global numerical optimization. In order to alleviate the defects of the existing DE variants, such as falling into local optima and the diversity degradation of population, the following four strategies were developed in this paper. First, a multi-schemes mutation strategy was devised to balance the exploration and exploitation of algorithm by dynamically dividing the whole population into three subpopulations based on the fitness values of individuals and the search performance of each operator. Then, a weighted control parameter setting scheme was developed to enlarge the exploration range of algorithm during the search process, and a random opposition learning strategy was introduced after the selection operation in DE to reduce the probability of falling into local optima and enhance the diversity of population. Moreover, an adaptive population size reduction mechanism was further presented to adaptively adjust the population size by making the best use of the distribution information of population, so as to strengthen the search capability of algorithm further. Thereby, MWADE was able to achieve a better balance between exploration and exploitation. Finally, a series of experiments were designed and conducted to demonstrate the benefit of MWADE on 30 benchmark functions from IEEE CEC 2017 test suite and a practical problem. Experimental results shown that MWADE had a better performance compared to both seven well-known DE variants and six other typical heuristic approaches.

Our future research will focus on designing a more reasonable multi-strategy adaptive mechanism for DE, and applying MWADE to other real and scientific applications, such as microgrid [56], neural networks [15], pattern recognition [57] and so on.

REFERENCES

[1] Y. Li, T. Han, H. Zhou, S. Tang, and H. Zhao, ''A novel adaptive L-SHADE algorithm and its application in UAV swarm resource configuration problem,'' *Inf. Sci.*, vol. 606, pp. 350–367, Aug. 2022.

[2] L. Liu, D. Zhao, F. Yu, A. A. Heidari, J. Ru, H. Chen, M. Mafarja, H. Turabieh, and Z. Pan, ''Performance optimization of differential evolution with slime mould algorithm for multilevel breast cancer image segmentation,'' *Comput. Biol. Med.*, vol. 138, Nov. 2021, Art. no. 104910.

[3] S. Chakraborty, A. K. Saha, A. E. Ezugwu, J. O. Agushaka, R. A. Zitar, and L. Abualigah, ''Differential evolution and its applications in image processing problems: A comprehensive review,'' *Arch. Comput. Methods Eng.*, vol. 30, pp. 985–1040, Mar. 2023.

[4] G.-G. Wang, D. Gao, and W. Pedrycz, ''Solving multiobjective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm,'' *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 8519–8528, Dec. 2022.

[5] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, ''Optimization by simulated annealing,'' *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[6] Z. Meng, Y. Chen, X. Li, C. Yang, and Y. Zhong, ''Enhancing quasi-affine transformation evolution (QUATRE) with adaptation scheme on numerical optimization,'' *Knowl.-Based Syst.*, vol. 197, Jun. 2020, Art. no. 105908.

[7] J. Kennedy and R. Eberhart, ''Particle swarm optimization,'' in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, 1995, pp. 1942–1948.

[8] R. Storn and K. Price, ''Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces,'' *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[10] T. Zhou, Z. Hu, Q. Su, and W. Xiong, ''A clustering differential evolution algorithm with neighborhood-based dual mutation operator for multimodal multiobjective optimization,'' *Expert Syst. Appl.*, vol. 216, Apr. 2023, Art. no. 119438.

[11] E. Zorarpacı, ''Data clustering using leaders and followers optimization and differential evolution,'' *Appl. Soft Comput.*, vol. 132, Jan. 2023, Art. no. 109838.

[12] M. Dai, X. Feng, H. Yu, and W. Guo, ''An opposition-based differential evolution clustering algorithm for emotional preference and migratory behavior optimization,'' *Knowl.-Based Syst.*, vol. 259, Jan. 2023, Art. no. 110073.

[13] F. Arce, E. Zamora, H. Sossa, and R. Barrón, ''Differential evolution training algorithm for dendrite morphological neural networks,'' *Appl. Soft Comput.*, vol. 68, pp. 303–313, Jul. 2018.

[14] M. S. AbouOmar, Y. Su, H. Zhang, B. Shi, and L. Wan, ''Observer-based interval type-2 fuzzy PID controller for PEMFC air feeding system using novel hybrid neural network algorithm-differential evolution optimizer,'' *Alexandria Eng. J.*, vol. 61, no. 9, pp. 7353–7375, Sep. 2022.

[15] Y. Xue, Y. Tong, and F. Neri, ''An ensemble of differential evolution and Adam for training feed-forward neural networks,'' *Inf. Sci.*, vol. 608, pp. 453–471, Aug. 2022.

[16] L. D. S. Coelho, R. C. T. Souza, and V. C. Mariani, ''Improved differential evolution approach based on cultural algorithm and diversity measure applied to solve economic load dispatch problems,'' *Math. Comput. Simul.*, vol. 79, no. 10, pp. 3136–3147, Jun. 2009.

[17] Z. Meng, Y. Zhong, and C. Yang, ''CS-DE: Cooperative strategy based differential evolution with population diversity enhancement,'' *Inf. Sci.*, vol. 577, pp. 663–696, Oct. 2021.

[18] M. Tian and X. Gao, ''Differential evolution with neighborhood-based adaptive evolution mechanism for numerical optimization,'' *Inf. Sci.*, vol. 478, pp. 422–448, Apr. 2019.

[19] M. Tian and X. Gao, ''An improved differential evolution with information intercrossing and sharing mechanism for numerical optimization,'' *Swarm Evol. Comput.*, vol. 50, Nov. 2019, Art. no. 100341.

[20] M. Tian, X. Gao, and X. Yan, ''Performance-driven adaptive differential evolution with neighborhood topology for numerical optimization,'' *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 105008.

[21] Z. Meng, J.-S. Pan, and L. Kong, ''Parameters with adaptive learning mechanism (PALM) for the enhancement of differential evolution,'' *Knowl.-Based Syst.*, vol. 141, pp. 92–112, Feb. 2018.

[22] X. Yan and M. Tian, ''Differential evolution with two-level adaptive mechanism for numerical optimization,'' *Knowl.-Based Syst.*, vol. 241, Apr. 2022, Art. no. 108209.

[23] J. Zhang and A. C. Sanderson, ''JADE: Adaptive differential evolution with optional external archive,'' *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[24] W. Gong and Z. Cai, ''Differential evolution with ranking-based mutation operators,'' *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2066–2081, Dec. 2013.

[25] J. Cheng, Z. Pan, H. Liang, Z. Gao, and J. Gao, ''Differential evolution algorithm with fitness and diversity ranking-based mutation operator,'' *Swarm Evol. Comput.*, vol. 61, Mar. 2021, Art. no. 100816.

[26] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, ''Differential evolution algorithm with ensemble of parameters and mutation strategies,'' *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1679–1696, Mar. 2011.

[27] Z. Meng and C. Yang, ''Two-stage differential evolution with novel parameter control,'' *Inf. Sci.*, vol. 596, pp. 321–342, Jun. 2022.

[28] A. Ghosh, S. Das, R. Mallipeddi, A. K. Das, and S. S. Dash, ''A modified differential evolution with distance-based selection for continuous optimization in presence of noise,'' *IEEE Access*, vol. 5, pp. 26944–26964, 2017.

[29] M. Wang, Y. Ma, and P. Wang, ''Parameter and strategy adaptive differential evolution algorithm based on accompanying evolution,'' *Inf. Sci.*, vol. 607, pp. 1136–1157, Aug. 2022.

[30] G. Sun, G. Yang, and G. Zhang, ''Two-level parameter cooperation-based population regeneration framework for differential evolution,'' *Swarm Evol. Comput.*, vol. 75, Dec. 2022, Art. no. 101122.

[31] Z. Meng, J.-S. Pan, and K.-K. Tseng, ''PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization,'' *Knowl.-Based Syst.*, vol. 168, pp. 80–99, Mar. 2019.

[32] Z. Meng and C. Yang, ''Hip-DE: Historical population based mutation strategy in differential evolution with parameter adaptive mechanism,'' *Inf. Sci.*, vol. 562, pp. 44–77, Jul. 2021.

[33] Z. Meng and J.-S. Pan, ''HARD-DE: Hierarchical archive based mutation strategy with depth information of evolution for the enhancement of differential evolution on numerical optimization,'' *IEEE Access*, vol. 7, pp. 12832–12854, 2019.

[34] R. Tanabe and A. Fukunaga, ''Success-history based parameter adaptation for differential evolution,'' in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 71–78.

[35] A. K. Qin, V. L. Huang, and P. N. Suganthan, ''Differential evolution algorithm with strategy adaptation for global numerical optimization,'' *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[36] Z. Meng, Y. Chen, and X. Li, ''Enhancing differential evolution with novel parameter control,'' *IEEE Access*, vol. 8, pp. 51145–51167, 2020.

[37] H. Zhang, J. Sun, K. C. Tan, and Z. Xu, ''Learning adaptive differential evolution by natural evolution strategies,'' *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 3, pp. 872–886, Jun. 2023.

[38] V. Stanovov, S. Akhmedova, and E. Semenkin, ''Biased parameter adaptation in differential evolution,'' *Inf. Sci.*, vol. 566, pp. 215–238, Aug. 2021.

[39] R. Tanabe and A. S. Fukunaga, ''Improving the search performance of SHADE using linear population size reduction,'' in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 1658–1665.

[40] Z. Zeng, M. Zhang, H. Zhang, and Z. Hong, ''Improved differential evolution algorithm based on the sawtooth-linear population size adaptive method,'' *Inf. Sci.*, vol. 608, pp. 1045–1071, Aug. 2022.

[41] C. Li, G. Sun, L. Deng, L. Qiao, and G. Yang, ''A population state evaluation-based improvement framework for differential evolution,'' *Inf. Sci.*, vol. 629, pp. 15–38, Jun. 2023.

[42] N. Awad, M. Ali, J. Liang, B. Qu, and P. Suganthan, ''Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization,'' Nanyang Technol. Univ., Jordan Univ. Sci. Technol., Zhenzhou Univ., 2016, pp. 1–34.

[43] W. Long, J. Jiao, X. Liang, S. Cai, and M. Xu, ''A random opposition-based learning grey wolf optimizer,'' *IEEE Access*, vol. 7, pp. 113810–113825, 2019.

[44] F. Wilcoxon, *Individual Comparisons by Ranking Methods*. Berlin, Germany: Springer, 1992.

[45] J. Derrac, S. García, D. Molina, and F. Herrera, ''A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,'' *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.

[46] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, ''Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems,'' *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.

[47] X. Xia, L. Gui, F. Yu, H. Wu, B. Wei, Y.-L. Zhang, and Z.-H. Zhan, "Triple archives particle swarm optimization," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4862–4875, Dec. 2020.

[48] Z.-Y. Jiang, Z.-X. Cai, and Y. Wang, "Hybrid self-adaptive orthogonal genetic algorithm for solving global optimization problems," *J. Softw.*, vol. 21, no. 6, pp. 1296–1307, Jun. 2010.

[49] H. A. Shehadeh, "A hybrid sperm swarm optimization and gravitational search algorithm (HSSOGSA) for global optimization," *Neural Comput. Appl.*, vol. 33, no. 18, pp. 11739–11752, Mar. 2021.

[50] N. Lynn and P. N. Suganthan, "Ensemble particle swarm optimizer," *Appl. Soft Comput.*, vol. 55, pp. 533–548, Jun. 2017.

[51] G. Sun, R. Han, L. Deng, C. Li, and G. Yang, "Hierarchical structure-based joint operations algorithm for global optimization," *Swarm Evol. Comput.*, vol. 79, Jun. 2023, Art. no. 101311.

[52] K. Kassoul, N. Zufferey, N. Cheikhrouhou, and S. B. Belhaouari, "Exponential particle swarm optimization for global optimization," *IEEE Access*, vol. 10, pp. 78320–78344, 2022.

[53] H. Bayzidi, S. Talatahari, M. Saraee, and C.-P. Lamarche, "Social network search for solving engineering optimization problems," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–32, Sep. 2021.

[54] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Comput. Ind.*, vol. 41, no. 2, pp. 113–127, Mar. 2000.

[55] B. D. Youn and K. K. Choi, "A new response surface methodology for reliability-based design optimization," *Comput. Struct.*, vol. 82, nos. 2–3, pp. 241–256, Jan. 2004.

[56] L. Zhang and R. Tang, "Dispatch for a continuous-time microgrid based on a modified differential evolution algorithm," *Mathematics*, vol. 11, no. 2, p. 271, Jan. 2023.

[57] M. G. Omran, A. P. Engelbrecht, and A. Salman, "Differential evolution methods for unsupervised image classification," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, Sep. 2005, pp. 966–973.

MENGNAN TIAN received the Ph.D. degree from Shaanxi Normal University, in 2020. He is currently an Associate Professor with the School of Sciences, Xi'an Polytechnic University. His current research interests include intelligent computing and global optimization.

YANHUI MENG is currently pursuing the master's degree with the School of Sciences, Xi'an Polytechnic University. His current research interests include evolutionary computing, intelligent optimization, and intelligent algorithms.

XINGSHI HE received the M.S. degree in science from Shaanxi Normal University, in 1989. He is currently a Professor with the School of Sciences, Xi'an Polytechnic University. His current research interests include intelligent algorithms and data mining.

QINGQING ZHANG received the Ph.D. degree from Northwestern Polytechnical University, in 2016. She is currently a Staff Member with the School of Sciences, Xi'an Polytechnic University. Her current research interests include intelligent computing and text mining.

YANGHAN GAO is currently pursuing the master's degree with the School of Sciences, Xi'an Polytechnic University. His current research interests include intelligent algorithms, machine learning, and global optimization.