Self-Adaptive Differential Evolution Algorithm With Zoning Evolution of Control Parameters and Adaptive Mutation Strategies

Qinqin Fan and Xuefeng Yan

***Abstract*—The performance of the differential evolution (DE) algorithm is significantly affected by the choice of mutation strategies and control parameters. Maintaining the search capability of various control parameter combinations throughout the entire evolution process is also a key issue. A self-adaptive DE algorithm with zoning evolution of control parameters and adaptive mutation strategies is proposed in this paper. In the proposed algorithm, the mutation strategies are automatically adjusted with population evolution, and the control parameters evolve in their own zoning to self-adapt and discover near optimal values autonomously. The proposed algorithm is compared with five state-of-the-art DE algorithm variants according to a set of benchmark test functions. Furthermore, seven nonparametric statistical tests are implemented to analyze the experimental results. The results indicate that the overall performance of the proposed algorithm is better than those of the five existing improved algorithms.**

***Index Terms*—Control parameter adaptation, differential evolution (DE) algorithm, mutation strategy adaptation, zoning evolution.**

I. INTRODUCTION

DIFFERENTIAL evolution (DE) algorithm proposed by Storn and Price [1] is a competitive and reliable evolutionary computing technique for solving a wide variety of complex optimization problems. The optimization performance [2]–[5] of the DE algorithm not only depends on the choice of three control parameters (i.e., mutation control parameter F, crossover control parameter CR, and population size NP), but also on the choice of trial vector generation strategies (i.e., mutation and crossover strategies). To improve the algorithm's performance, several useful empirical guidelines for selecting control parameter settings and mutation strategies have been introduced by many researchers during the past decade. Eiben *et al.* [6] and Brest *et al.* [7] proposed three parameter control techniques (i.e., deterministic, adaptive, and self-adaptive) and an encoding technique (i.e., encoding control parameters F and CR into an individual), respectively. However, different optimization problems or particular evolution stages often require chosen mutation strategies and suitable control parameters [8] during the evolution process; an inappropriate choice of mutation strategies and control parameters may directly influence the algorithm's performance [9]–[11]. Different control parameter combinations may also significantly affect algorithm's search capability. Wang *et al.* [5] and Mallipeddi *et al.* [8] employed constant combinations of control parameters (i.e., F and CR values are determined before actual DE usage) to maintain the searching capability of the control parameter combinations; however, their proposed methods may be unable to adapt quickly in complex optimization environments. On the basis of these considerations, obtaining the most suitable mutation strategy and reasonable combinations of control parameters at different evolution phases is important to improve DE performance. In this paper, a self-adaptive DE (SDE) algorithm with zoning evolution of control parameters and adaptive mutation strategies (ZEPDE) is proposed. In ZEPDE, the number of each mutation strategy can be gradually adjusted via a roulette wheel, and real-time optimal control parameter combinations can be obtained by zoning evolution, which can maintain the control parameter distribution. ZEPDE was compared with five improved DE variants according to 25 CEC2005 and 28 CEC2013 test functions.

The remainder of this paper is organized as follows. Section II introduces the basic DE algorithm. Section III reviews the related researches on DE algorithm. Section IV presents the proposed ZEPDE algorithm. Section V reports the experimental results and sensitive analysis of ZEPDE parameters. Finally, the conclusion is summarized in Section VI. To gain better understanding, interested readers can refer to the supplementary file.

II. DE ALGORITHM

In the evolution process of DE, mutation, crossover, and selection operators are performed. The vector containing D optimized variables $x_1, x_2, \ldots, x_D$ is denoted by $\boldsymbol{x}$. $\boldsymbol{x}_i^G = [x_{i,1}^G, x_{i,2}^G, \ldots, x_{i,D}^G]$ denotes the ith solution (or individual) in the Gth generation. The population of the Gth generation is denoted by $\boldsymbol{X}^G = [\boldsymbol{x}_1^G, \boldsymbol{x}_2^G, \ldots, \boldsymbol{x}_{\mathrm{NP}}^G]$, which

Manuscript received December 10, 2013; revised November 16, 2014 and January 21, 2015; accepted January 27, 2015. Date of publication March 10, 2015; date of current version December 14, 2015. This work was supported in part by the 973 project of China under Grant 2013CB733600, in part by the National Natural Science Foundation of China under Grant 21176073, in part by the Program for New Century Excellent Talents in University under Grant NCET-09-0346, and in part by the Fundamental Research Funds for the Central Universities. This paper was recommended by Y. Jin.

The authors are with the Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, East China University of Science and Technology, Shanghai 200237, China (e-mail: xfyan@ecust.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2015.2399478

contains NP individuals. NP generally is fixed during the evolution process. The minimum problem is described as follows:

$$\min_x f(x_1, x_2, \ldots, x_D)$$
$$x_j \in \left(x_j^{\text{low}}, x_j^{\text{high}}\right), j = 1, 2, \ldots, D \tag{1}$$

where f denotes the function that is subject to optimization. A D-dimensional space $\boldsymbol{P}_0$ is defined within the region $\{(x_j^{\text{low}}, x_j^{\text{high}})\, j = 1, 2, \ldots, D\}$.

The procedure of executing DE is as follows [9].

1) *Initialization Operation:* The initial individuals $\boldsymbol{x}_i^0, i = 1, 2, \ldots,$ NP are randomly generated in $\boldsymbol{P}_0$. Mutation F, crossover CR, and maximum number of generations $G_{\max}$ are determined. The current generation is set as $G = 0$.
2) For each individual $\boldsymbol{x}_i^G, i = 1, 2, \ldots, NP$, perform steps 3)–5) to derive the population for the next generation.
3) *Mutation Operation [12]:* For each $\boldsymbol{x}_i^G$ in the parent population, the mutant individual $\hat{\boldsymbol{x}}_i^{G+1}$ is generated as follows:
$$\hat{\boldsymbol{x}}_i^{G+1} = \boldsymbol{x}_{r_1}^G + F \cdot \left(\boldsymbol{x}_{r_2}^G - \boldsymbol{x}_{r_3}^G\right) \tag{2}$$
where $r_1, r_2, r_3 \in \{1, 2, \ldots, \text{NP}\}$ are randomly chosen and are different from the running index i. F is a real constant scaling factor within [0, 1], which controls the amplification of the differential variation $(\boldsymbol{x}_{r_2}^G - \boldsymbol{x}_{r_3}^G)$.
4) *Crossover Operation, [9]:* For each $\boldsymbol{x}_i^G$, a trial individual $\bar{\boldsymbol{x}}_i^{G+1}$ is generated as follows:
$$\bar{x}_{ij}^{G+1} = \begin{cases} \hat{x}_{ij}^{G+1}, & R_j \leq \text{CR} \\ x_{ij}^G, & \text{otherwise} \end{cases} \quad j = 1, 2, \ldots, D \tag{3}$$
where R_j is a uniform random number in the range of [0, 1].
5) *Evaluation Operation, [9]:* Offspring $\bar{\boldsymbol{x}}_i^{G+1}$ competes one-to-one with its parent $\boldsymbol{x}_i^G$. The evaluation operation is expressed as
$$\boldsymbol{x}_i^{G+1} = \begin{cases} \bar{\boldsymbol{x}}_i^{G+1}, & f\left(\bar{\boldsymbol{x}}_i^{G+1}\right) \leq f(\boldsymbol{x}_i^G) \\ \boldsymbol{x}_i^G, & \text{otherwise.} \end{cases} \tag{4}$$
6) $G = G + 1$.
7) Steps 2)–6) are repeated as long as the number of generations is smaller than the allowable maximum number $G_{\max}$.

Other useful strategies are as follows:

"DE/rand/2" [13]: $\hat{\boldsymbol{x}}_i^{G+1} = \boldsymbol{x}_{r1}^G + F\cdot(\boldsymbol{x}_{r2}^G - \boldsymbol{x}_{r3}^G) + F\cdot(\boldsymbol{x}_{r4}^G - \boldsymbol{x}_{r5}^G)$

"DE/best/2" [12]: $\hat{\boldsymbol{x}}_i^{G+1} = \boldsymbol{x}_{\text{best}}^G + F\cdot(\boldsymbol{x}_{r1}^G - \boldsymbol{x}_{r2}^G) + F\cdot(\boldsymbol{x}_{r3}^G - \boldsymbol{x}_{r4}^G)$

"DE/current-to-best/1" [12]:

$$\hat{\boldsymbol{x}}_i^{G+1} = \boldsymbol{x}_i^G + F \cdot \left(\boldsymbol{x}_{\text{best}}^G - \boldsymbol{x}_i^G\right) + F \cdot \left(\boldsymbol{x}_{r1}^G - \boldsymbol{x}_{r2}^G\right)$$

"DE/current-to-best/2" [13]:

$$\hat{\boldsymbol{x}}_i^{G+1} = \boldsymbol{x}_{\text{best}}^G + F \cdot \left(\boldsymbol{x}_{\text{best}}^G - \boldsymbol{x}_i^G\right) + F \cdot \left(\boldsymbol{x}_{r1}^G - \boldsymbol{x}_{r2}^G + \boldsymbol{x}_{r3}^G - \boldsymbol{x}_{r4}^G\right)$$

where $\boldsymbol{x}_{\text{best}}^G$ is the individual vector with the best fitness value in the population at generation G.

III. Literature Review

Although the DE algorithm has attracted the attention of many researchers because of its high search accuracy, robustness, and good convergence speed, selecting suitable mutation strategies and control parameters at different evolution stages is difficult, particularly when the optimization problems are complex and require different search capabilities.

To improve DE performance, DE researchers have proposed several empirical guidelines for choosing control parameters and mutation strategies over the past decade. For instance, Storn and Price [14] suggested that the suitable population size should be between $5D$ and $10D$; F can be within the range of [0.4–1], and CR can be 0.1 or 0.9. Based on the experimental results for three test functions, Gamperle *et al.* [15] indicated that the appropriate population size should be between $3D$ and $8D$; $F = 0.6$ can be selected, and CR in the range of [0.3–0.9] is a good choice. Ronkkonen *et al.* [16] recommended that the range of population size be between $2D$ and $4D$. $F = 0.9$ would be a good initial choice, but a suitable F should be selected within the range of [0.4–0.95]. CR should be within [0–0.2] for separable test functions and within [0.9–1] for multimodal and parameter dependent test functions. Zielinski *et al.* [17] suggested that the settings $F \geq 0.6$ and CR ≥ 0.6 are beneficial to algorithm convergence in most cases. Moreover, several researchers [14], [15], [18]–[21] have proposed and investigated various mutation strategies in consideration of the fact that different optimization problems require different search capabilities. Although these guidelines are useful to improve DE performance, they may cause confusion to DE users when the properties of optimization problems are not sufficiently clear and lack sufficient justifications because these guidelines were designed based on specific experiments (i.e., lack of universality) [13]. Constant control parameter settings and single mutation strategy also violate the nature of Darwinian evolution and cannot adapt to different optimization problems and evolution phases.

Many adaptive or self-adaptive DE variants have been proposed to avoid manual tuning of DE control parameters and the use of a constant mutation strategy during the evolution process. For example, Abbass [22] introduced a self-adaptive pareto differential evolution algorithm (SPDE) wherein the control parameters F and CR are generated by using a Gaussian distribution. On the basis of the DE algorithm, Vu *et al.* [23] recently introduced a direction-guided evolution algorithm (DEAL) wherein two types of directions (i.e., convergence and spreading directions) are utilized to guide the population evolution. Their experimental results show that DEAL can perform better than other well-known algorithms. Liu and Lampinen [11] proposed a fuzzy adaptive DE (FADE) algorithm wherein appropriate control parameters F and CR can be generated by using fuzzy logic controllers. Their results indicate that the performance of FADE in a set of benchmark set functions is better than that of standard DE algorithms. Zhang and Sanderson [21] proposed a new DE algorithm (JADE), in which a mutation strategy with an optional external archive is used and control parameters F and CR can be automatically adjusted based

TABLE I
INDIVIDUAL WITH CONTROL PARAMETERS AND MUTATION STRATEGY

Individual	Control parameters and strategies
$\boldsymbol{x}_1^G$	$(F_1^G,\ CR_1^G)$, Strategy_1^G
$\boldsymbol{x}_2^G$	$(F_2^G,\ CR_2^G)$, Strategy_2^G
...	...
$\boldsymbol{x}_{NP}^G$	$(F_{NP}^G,\ CR_{NP}^G)$, Strategy_{NP}^G

on their previous record of success. Zaharie [24] introduced a parameter adaptation for DE (ADE) in which the adjustment of F and CR is based on population diversity and a multipopulation approach is used. Brest *et al.* [7] proposed a self-adaptive jDE, in which each individual has its own control parameters (i.e., F and CR). The control parameters were adjusted by two new parameters. Omran *et al.* [25] introduced a SDE wherein F is adaptive and CR is generated from a normal distribution $N(0.5, 0.15)$. Qin *et al.* [13] also introduced a SDE wherein both mutation strategies and their associated control parameters can gradually self-adapt by learning from their previous successful experiences. Results indicate that SaDE outperforms several DE variants on a set of 26 test functions. Pan *et al.* [26] and Mallipeddi *et al.* [8] proposed a DE algorithm with a self-adaptive trial vector generation strategy and control parameters (SspDE) and an ensemble of mutation strategies and control parameters with DE (EPSDE), respectively. In both DE variants, mutation strategies and control parameters can automatically adapt to population evolution. Ghosh *et al.* [27] introduced an improved DE (FiADE) wherein the adaptation of F and CR is based on the fitness function values of individuals. Their experimental results show that FiADE is a competitive optimization tool for solving various optimization problems. Wang *et al.* [5] introduced a composite DE (CoDE), in which each mutation strategy is randomly combined with a fixed control parameter setting. CoDE employs three mutation strategies and three control parameter settings. Gong *et al.* [28] introduced an enhanced DE in which a new strategy adaptation mechanism is used. Meanwhile, several researchers have applied an adaptive or self-adaptive population size in DE. Teo [29], [30] proposed the first a self-adaptive population size approach; absolute and relative encoding methodologies are employed in the population size of the proposed algorithm. Teng *et al.* [31] also utilized two different encoding methodologies to implement a self-adaptive population size. Tirronen and Neri [32] proposed an adaptive population size wherein a measurement of the fitness diversity was used. Zhu *et al.* [33] introduced an adaptive population tuning scheme to adjust population size. Various other population size reduction schemes have been proposed by [34], [35].

The DE algorithm has been successfully applied to solve various real-world engineering optimization problems due to its effectiveness, robustness, and simplicity. For example, Zhong and Zhang [36] proposed an adaptive DE algorithm to solve subpixel mapping. Chen *et al.* [37] introduced a modified differential evolution which is applied to obtain the optimal controller parameters of an adaptive neural fuzzy network. Neri and Mininno [38] proposed a memetic compact differential evolution for the control of commercial robots. Zamuda *et al.* [39] proposed an approach based on the DE algorithm for procedural 3-D models reconstruction of woody plants. Salvatore *et al.* [40] employed the DE algorithm to optimize an algorithm of sensorless control of induction motors. Amin *et al.* [41] employed the DE algorithm to adjust the air traffic controllers taskload in real time.

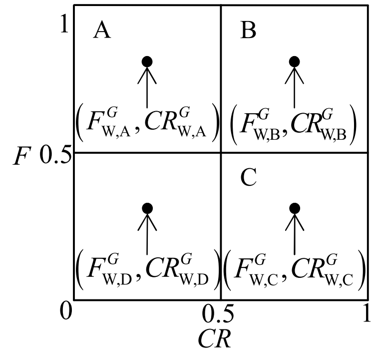


Fig. 1. Zoning type of ZEPDE.

IV. SELF-ADAPTIVE DIFFERENETIAL EVOLUTION ALGORITHM WITH ZEPDE

Although many self-adaptive DE algorithms have been introduced, how to maintain the search capabilities of control parameter combinations of different zoning (i.e., zoning evolution of control parameters) during the evolution process has not been considered by other DE researchers. For example, EPSDE and CoDE employ fixed combinations of control parameters throughout the entire search process. Self-adaptive control parameters are generated in SaDE and JADE, but the performance of control parameter combinations in different zoning is not considered. Based on these observations, we propose the ZEPDE algorithm, in which the appropriate mutation strategy can be gradually adjusted by a roulette wheel, and suitable combinations of F and CR can be generated by zoning evolution. Each individual has its own control parameter combination and mutation strategy (Table I).

A. Self-Adaptive Control Parameters Based on Zoning Evolution

To achieve reasonable combinations of control parameters during different evolution stages in ZEPDE, the total region of control parameters (i.e., F and CR) is divided into four same-size areas (Fig. 1). The zoning evolution operations of control parameters of the proposed algorithm can be described by the following steps.

1) Count the number of control parameters combinations in each region. The number of control parameter combinations in the hth region is assumed to be N_h^G, $h = 1, 2, \ldots, H$, where H is the total number of zoning (NZ).
2) In each zoning, if offspring individual fitness is better than that of its parent, then its control parameter combination is regarded as the elite control

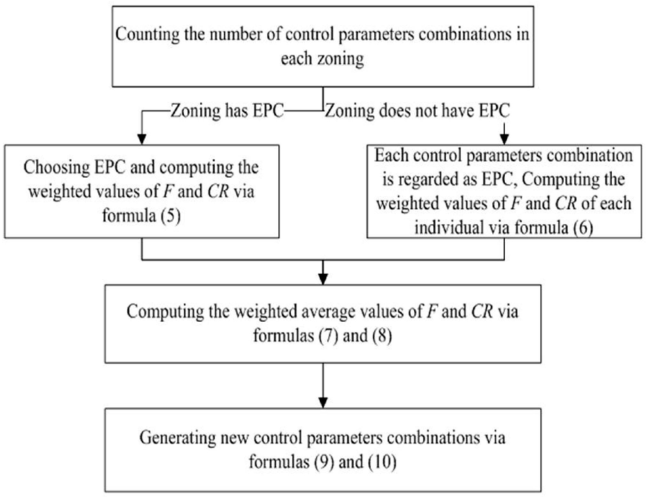


Fig. 2. Procedure of zoning evolution.

parameter combination (EPC). If the hth region has EPC, assuming that the number of EPC in the hth region is $N^G_{h,\text{elite}}$, $0 \le N^G_{h,\text{elite}} \le N^G_h$, then the weighted value of each EPC in the hth region is computed as follows:

$$w^G_{N^G_{h,e}} = \frac{f\left(\bar{\boldsymbol{x}}^{G+1}_{N^G_{h,e}}\right) - f\left(\boldsymbol{x}^G_{N^G_{h,e}}\right)}{\sum_{N^G_{h,e}=1}^{N^G_{\text{h,elite}}} f\left(\bar{\boldsymbol{x}}^{G+1}_{N^G_{h,e}}\right) - f\left(\boldsymbol{x}^G_{N^G_{h,e}}\right)} \tag{5}$$

where $N^G_{h,e} = 1, 2, \ldots, N^G_{h,\text{elite}}$.

If the hth region does not have EPC, then each control parameter combination is regarded as EPC. We let

$$w^G_{N^G_{h,e}} = 1/N^G_h. \tag{6}$$

3) The weighted average values of F and CR in the hth region are computed as follows:

$$F^G_{W,h} = \sum_{N^G_{h,e}=1}^{N^G_{h,\text{elite}}} w^G_{N^G_{h,e}} \times F^G_{N^G_{h,e}} \tag{7}$$

$$\text{CR}^G_{w,h} = \sum_{N^G_{h,e}=1}^{N^G_{h,\text{elite}}} w^G_{N^G_{h,e}} \times \text{CR}^G_{N^G_{h,e}}. \tag{8}$$

4) The control parameters are generated in the hth region as the following way:

$$F^{G+1}_{h,l} = Cauchy\left(F^G_{W,h}, \sigma\right) \quad l = 1, 2, \ldots, N^G_h \tag{9}$$

$$\text{CR}^{G+1}_{h,l} = N\left(\text{CR}^G_{w,h}, \sigma\right) \quad l = 1, 2, \ldots, N^G_h \tag{10}$$

where Cauchy and N are the Cauchy and normal distribution functions, respectively. $\sigma = 0.55 - 0.3\times(1 - G/G_{\max})$.

The zoning evolution of control parameters is shown in Fig. 2.

B. Self-Adaptive Mutation Strategies

The DE/rand/1 mutation strategy has good exploration capability and is widely utilized in DE. Meanwhile, self-adaptive mutation strategies (which contain greedy mutation strategies) may lead to decreased population diversity during the early evolution process. To obtain good balance between global and local search capabilities in the proposed algorithm, the DE/rand/1 strategy is used to optimize the population during the early stages of the evolution process. Subsequently, self-adaptive mutation strategies are utilized to search for the optimal solution in consideration of the fact that different evolution stages require different search capabilities. On the basis of the mutation strategies selected in SaDE [13] and CoDE [5], five mutation strategies are selected for the proposed algorithm. These five strategies are DE/rand/1, DE/rand/2, DE/best/2, DE/current-to-best/1, and DE/current-to-best/2.

C. Boundary Operation of Control Parameters

An extremely large value of F can increase population diversity and lead to slow convergence speed, whereas an extremely small value of F may lead to stagnation or premature convergence [4]. Therefore, the range of F in this paper is within [0.1–1] (similar to jDE in [7]) during the early evolution process and within [0–1] during the later stages of evolution. Meanwhile, the range of CR is from [0–1] throughout the entire search process. Infeasible control parameters are reset via the weighted average values of F and CR in the early stages of evolution and regenerated on the boundary in the middle and late stages because Cauchy and normal distribution functions may result in the loss of boundary distribution of the control parameters.

D. Overall Implementation of ZEPDE

The proposed algorithm is described as follows.

1) *Initialization Operation:* The values of parameters such as the number of individuals in the population NP, maximum variation of selective probability (Msp) in each generation, and maximum generations $G_{\max}$ are determined. Set Setp = 0.175 (i.e., only use DE/rand/1 mutation strategy before G_s = Setp $\times$ $G_{\max}$ generations), current generation G = 0, and Bset = 0.35 (i.e., a boundary operation of control parameters is used before G_b = Bset $\times$ $G_{\max}$ generations, whereas the other boundary operation is employed after G_b = Bset $\times$ $G_{\max}$ generations). The initial population $\boldsymbol{P}^0_1$ is generated, and the number of individuals belonging to each mutation strategy is set (i.e., $N^{G_s}_{\text{str\_rand/1}} = N^{G_s}_{\text{str\_rand/2}} = N^{G_s}_{\text{str\_best/2}} = N^{G_s}_{\text{str\_ current-to-best/1}} = N^{G_s}_{\text{str\_ current to-best/2}} = \text{NP}/5$).
2) *Population Evolution (Mutation Operation):* For each individual $\boldsymbol{x}^G_i$, $i = 1, 2, \ldots$, NP, if $G < G_s$, the mutation strategy DE/rand/1 is used to produce the candidate population for the next generation $G+1$. Otherwise, a new trial individual $\hat{x}^{G+1}_i$ is generated by using the corresponding mutation strategy, which is randomly selected from the mutation strategy pool.

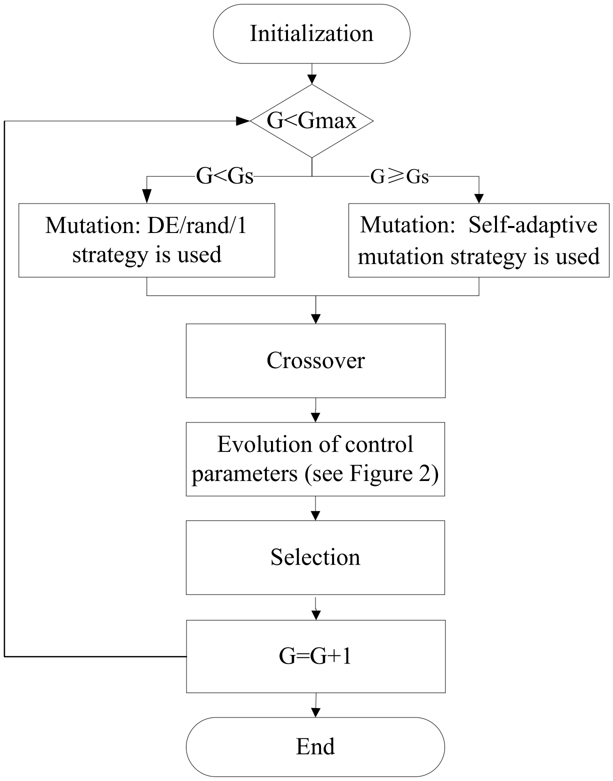


Fig. 3. Framework of ZEPDE.

Boundary operation: If $\hat{x}_{ij}^{G+1} < x_j^{\text{low}}$ or $\hat{x}_{ij}^{G+1} > x_j^{\text{high}}$, then $\hat{\boldsymbol{x}}_{ij}^{G+1} = x_{r_1 j}^{G}$.

Crossover operation

$$\bar{x}_{ij}^{G+1} = \begin{cases} \hat{x}_{ij}^{G+1}, & R_j \le \text{CR} \\ x_{ij}^{G}, & \text{otherwise} \quad j = 1, 2, \ldots, D. \end{cases}$$

Selection operation

$$\boldsymbol{x}_i^{G+1} = \begin{cases} \bar{\boldsymbol{x}}_i^{G+1}, & f\left(\bar{\boldsymbol{x}}_i^{G+1}\right) \le f\left(\boldsymbol{x}_i^{G}\right) \\ \boldsymbol{x}_i^{G}, & \text{otherwise.} \end{cases}$$

3) *Self-Adaptive Mutation Strategy:* If $G = G_s$, five mutation operations, i.e., DE/rand/1, DE/current-to-best/2, DE/current-to-best/1, DE/best/2, and DE/rand/2, are randomly allocated for all individuals. Each individual has its own mutation operation, and each mutation operation has $N_{\text{str\_rand/1}}^{G_s} = N_{\text{str\_rand/2}}^{G_s} = N_{\text{str\_best/2}}^{G_s} = N_{\text{str\_current-to-best/1}}^{G_s} = N_{\text{str\_current-to-best/2}}^{G_s} = \text{NP}/5$ individuals. If $G \ge G_s$, the maximum objective function value is obtained as follows:

$$f_{\max} = \max\left(f\left(\bar{\boldsymbol{x}}_i^{G+1}\right), i = 1, 2, \ldots, \text{NP}\right). \tag{11}$$

The difference between the objective function value of each individual and $f_{\max}$ is calculated in the following way:

$$\Delta f_i^{G+1} = \left| f\left(\bar{\boldsymbol{x}}_i^{G+1}\right) - f_{\max} \right|, \quad i = 1, 2, \ldots, \text{NP}. \tag{12}$$

For each mutation strategy, the sum of difference is computed as

$$S_{\text{str\_name}}^{G+1} = \sum_{k=1}^{N_{\text{str\_name}}^{G}} \Delta f_{\text{str\_name},k}^{G+1}, \quad k = 1, 2, \ldots, N_{\text{str\_name}}^{G} \tag{13}$$

where str_name denotes one of the five mutation strategies and $N_{\text{str\_name}}^{G}$ denotes the number of individuals for a special mutation strategy.

The sum of difference of the five mutation strategies is calculated as follows:

$$S^{G+1} = S_{\text{rand/1}}^{G+1} + S_{\text{rand/2}}^{G+1} + S_{\text{current-to-best/1}}^{G+1} + S_{\text{current-to-best/2}}^{G+1} + S_{\text{best/2}}^{G+1}. \tag{14}$$

The selective probability (sp) of each mutation strategy is computed in the following:

$$sp_{\text{str\_name}}^{G+1} = S_{\text{str\_name}}^{G+1} / S^{G+1}. \tag{15}$$

The sp of each mutation strategy can be formulated as follows:

$$sp_{\text{str\_name}}^{G+1} = \begin{cases} sp_{\text{str\_name}}^{G} + \text{Msp}, & \text{if } sp_{\text{str\_name}}^{G+1} - sp_{\text{str\_name}}^{G} > \text{Msp} \\ sp_{\text{str\_name}}^{G} - \text{Msp}, & \text{if } sp_{\text{str\_name}}^{G+1} - sp_{\text{str\_name}}^{G} < -\text{Msp} \\ sp_{\text{str\_name}}^{G+1}, & \text{otherwise.} \end{cases} \tag{16}$$

The cumulative probability (cp) of each mutation strategy is computed as follows:

$$cp_{\text{rand/1}}^{G+1} = sp_{\text{rand/1}}^{G+1} \tag{17}$$

$$cp_{\text{rand/2}}^{G+1} = sp_{\text{rand/1}}^{G+1} + sp_{\text{rand/2}}^{G+1} \tag{18}$$

$$cp_{\text{current-to-best/1}}^{G+1} = sp_{\text{rand/1}}^{G+1} + sp_{\text{rand/2}}^{G+1} + sp_{\text{current-to-best/1}}^{G+1} \tag{19}$$

$$cp_{\text{current-to-best/2}}^{G+1} = sp_{\text{rand/1}}^{G+1} + sp_{\text{rand/2}}^{G+1} + sp_{\text{current-to-best/1}}^{G+1} + sp_{\text{current-to-best/2}}^{G+1} \tag{20}$$

$$cp_{\text{best/2}}^{G+1} = sp_{\text{rand/1}}^{G+1} + sp_{\text{rand/2}}^{G+1} + sp_{\text{current-to-best/1}}^{G+1} + sp_{\text{current-to-best/2}}^{G+1} + sp_{\text{best/2}}^{G+1}. \tag{21}$$

Based on the cp of each mutation strategy, the number of each mutation strategy (i.e., $N_{\text{str\_rand/1}}^{G+1}$, $N_{\text{str\_rand/2}}^{G+1}$, $N_{\text{str\_ current-to-best/1}}^{G+1}$, $N_{\text{str\_ current-to-best/2}}^{G+1}$, and $N_{\text{str\_best/2}}^{G+1}$) is generated by a roulette wheel.

4) *Self-Adaptive Control Parameters:* The detailed descriptions of the zoning evolution operations of control parameters are presented in Section IV-A.

If $G < 0.35 \times G_{\max}$, $F_i^{G+1} > 1$, or $F_i^{G+1} < 0.1$, the mutation control parameter is updated as follows:

$$F_i^{G+1} = N(\mu_F, 0.2)$$

where $\mu_F = (F_{W,A}^{G} + F_{W,B}^{G} + F_{W,C}^{G} + F_{W,D}^{G})/4$.

If $G < 0.35 \times G_{\max}$, $\text{CR}_i^{G+1} > 1$ or $\text{CR}_i^{G+1} < 0$, the crossover control parameter can be updated as follows:

$$\text{CR}_i^{G+1} = N(\mu_{\text{CR}}, 0.2)$$

TABLE II
RESULTS OF WILCOXON TEST ON 30-DIMENSIONAL CEC2005 FUNCTIONS

ZEPDE v.s.	+	−	≈
jDE	16	1	8
SaDE	15	0	10
JADE	9	5	11
EPSDE	15	5	5
CoDE	7	4	14

where $\mu_{\mathrm{CR}} = (\mathrm{CR}_{W,A}^{G} + \mathrm{CR}_{W,B}^{G} + \mathrm{CR}_{W,C}^{G} + \mathrm{CR}_{W,D}^{G})/4$. If $G \geq 0.35 \times G_{\max}$, the mutation and crossover control parameters can be updated as follows:

$$F_i^{G+1} = \begin{cases} 1, & \text{if } F_i^{G+1} > 1, \\ \left|N\left(0, 0.15 \times \left(1 - (G/G_{\max})^2\right)\right)\right|, & \text{if } F_i^{G+1} < 0. \end{cases} \tag{22}$$

$$\mathrm{CR}_i^{G+1} = \begin{cases} 1, & \text{if } \mathrm{CR}_i^{G+1} > 1, \\ \left|N\left(0, 0.15 \times \left(1 - (G/G_{\max})^2\right)\right)\right|, & \text{if } \mathrm{CR}_i^{G+1} < 0. \end{cases} \tag{23}$$

5) $G = G + 1$.
6) Repeat steps 2)–5) as long as the number of generations is equal to the allowable maximum number $G_{\max}$.

The framework of ZEPDE is shown in Fig. 3.

V. EXPERIMENTAL STUDY ON ZEPDE

To demonstrate the overall performance of the proposed algorithm, 25 CEC2005 [42] and 28 CEC2013 [43] benchmark functions were used in the experiments. The optimization performance of ZEPDE was then compared with that of five state-of-the-art DE variants, i.e., jDE [7], SaDE [13], JADE [21], EPSDE [8], and CoDE [5]. All these DE variants were programmed in MATLAB (MATLAB R2012a) and run on a Windows 7 operating system (64 bit). The maximum number of function evaluations (FEs) is set to 300 000 for 30-dimensional test functions and 500 000 for 50-dimensional test functions in all algorithms. The number of independent runs is set to 30 and the accuracy level of optimization results is set to be 1E-8 (i.e., the optimization result less than 1E-8 as zero). The population size settings of these DE variants are different, 100 for ZEPDE, and the population size settings of the other algorithms are similar to those in their original papers, namely, 100 for jDE and JADE; 50 for EPSDE and SaDE; 30 for CoDE. To effectively analyze the experimental results obtained by the compared algorithms, seven nonparametric statistical tests with a significance level of 0.05 are utilized in the experiments. These tests are Wilcoxon's rank sum test [44], Kruskal–Wallis test with multiple comparisons [45], Friedman's test [46], Bonferroni–Dunn's test [47], Holm's procedure, Iman–Davenport test, and Hochberg's procedure [48]. For the Wilcoxon's rank sum test, the "+," "−," and "≈" marks indicate that ZEPDE performs significantly better than, worse than, and almost similar to the compared algorithms, respectively. For the Kruskal–Wallis test with multiple comparisons, "K+" denotes that the result is the best among all compared results, and "K−" or "K≈" indicate that the result is significantly worse than or almost the same as the best result, respectively.

TABLE III
RANKING OBTAINED BY FRIEDMAN'S TEST ON 30-DIMENSIONAL CEC2005 FUNCTIONS

Algorithms	Ranking
jDE	4.12
SaDE	4.18
JADE	3.04
EPSDE	4.26
CoDE	2.96
ZEPDE	**2.44**

TABLE IV
p-VALUES OBTAINED BY BONFERRONI–DUNN'S, HOLM'S, AND HOCHBERG'S PROCEDURES FOR THE COMPARED DE ALGORITHMS ON 30-DIMENSIONAL CEC2005 FUNCTIONS

ZEPDE v.s.	z	Unadjusted p	Bonferroni-Dunn p	Holm p	Hochberg p
jDE	3.174	**0.0014**	**0.0074**	**0.0044**	**0.0044**
SaDE	3.2882	**0.0010**	**0.0050**	**0.004**	**0.004**
JADE	1.1338	0.2568	1	0.5136	0.3257
EPSDE	3.439	**0.00058**	**0.0029**	**0.0029**	**0.0029**
CoDE	0.9827	0.3257	1	0.5136	0.3257

A. Comparison With Five Improved DE on 30-Dimensional Problems (CEC2005)

In this experiment, ZEPDE is compared with five famous DE variants (i.e., jDE, SaDE, JADE, EPSDE, and CoDE) according to 25 30-dimensional CEC2005 benchmark functions to evaluate its overall performance. The experimental results are given in supplementary (Table A1) due to pages limit.

Wilcoxon test is employed to analyze the optimization results shown in Table II. It can be seen from Table II that ZEPDE performs better than jDE, SaDE, JADE, EPSDE, and CoDE on 16, 15, 9, 15, and 7 test functions, respectively. jDE, JADE, EPSDE, and CoDE perform significantly better than ZEPDE on one, five, five, and four test functions, respectively. The performance of SaDE cannot be significantly better than that of ZEPDE on any test functions.

Friedman's test is also employed to evaluate the performances of all the compared algorithms. The rankings obtained by Friedman's test are shown in Table III. Table III indicates that the overall performance of ZEPDE performs better than those of the other compared DE algorithms. To detect the differences among the compared algorithms, p-values are obtained by Bonferroni–Dunn's, Holm's, and Hochberg's procedures shown in Table IV. Table IV indicates that the average performance of ZEPDE is significantly better than those of jDE, EPSDE, and SaDE and is similar to those of JADE and CoDE. However, the results obtained by Wilcoxon test and Friedman's test, it is observed that the overall performance of ZEPDE is better than those of JADE and CoDE.

B. Comparison With Five Improved DE on 50-Dimensional Problems (CEC2005)

In this experiment, 25 50-dimensional CEC2005 benchmark functions are used to test the performances of all

TABLE V
OPTIMIZATION RESULTS FOR 25 50-DIMENSIONAL CEC2005 TEST FUNCTIONS

ZEPDE v.s.	+	−	≈
jDE	16	4	5
SaDE	20	1	4
JADE	11	6	8
EPSDE	18	5	2
CoDE	13	5	7

TABLE VI
RANKING OBTAINED BY FRIEDMAN'S TEST ON 50-DIMENSIONAL CEC2005 FUNCTIONS

Algorithms	Ranking
jDE	3.64
SaDE	5.06
JADE	2.7
EPSDE	4.28
CoDE	2.98
ZEPDE	**2.34**

TABLE VII
p-VALUES OBTAINED BY BONFERRONI–DUNN'S, HOLM'S, AND HOCHBERG'S PROCEDURES FOR THE COMPARED DE ALGORITHMS ON 50-DIMENSIONAL CEC2005 FUNCTIONS

ZEPDE v.s.	z	Unadjusted p	Bonferroni-Dunn p	Holm p	Hochberg p
jDE	2.456	**0.0140**	0.070	**0.042**	**0.042**
SaDE	5.140	**2.742E-07**	**1.371E-06**	**1.371E-06**	**1.371E-06**
JADE	0.680	0.496	1	0.496	0.496
EPSDE	3.666	**0.00024**	**0.00123**	**0.00098**	**0.00098**
CoDE	1.209	0.2264	1	0.452	0.452

compared algorithms. The experimental results are given in supplementary (Table A2) due to pages limit.

The Wilcoxon test and Friedman's test are employed to demonstrate the differences between ZEPDE and the other compared algorithms. Table V indicates that ZEPDE significantly outperforms jDE, SaDE, JADE, EPSDE, and CoDE on 16, 20, 11, 18, and 30 test functions, respectively. Table VI shows that the overall performance of the proposed algorithm is the best among these compared algorithms.

Bonferroni–Dunn's, Holm's, and Hochberg's procedures are employed to detect the global differences among the different DE variants. Table VII shows that the average performance of ZEPDE is significantly better than those of jDE, EPSDE, and SaDE and is similar to those of JADE and CoDE. However, it can be observed from the statistical analysis results that the average performance of the proposed algorithm is the best among the compared algorithms on 25 50-dimensional test functions.

C. Comparison With Five Improved DE on 30-Dimensional Problems (CEC2013)

In this experiment, 28 30-dimensional CEC2013 benchmark functions are used to test the performances of all compared DE variants. The experimental results are given in supplementary (Table A3) due to pages limit.

TABLE VIII
RESULTS OF WILCOXON TEST ON 30-DIMENSIONAL CEC2013 FUNCTIONS

ZEPDE v.s.	+	−	≈
jDE	16	5	7
SaDE	19	3	6
JADE	10	9	9
EPSDE	17	3	8
CoDE	11	7	10

TABLE IX
RANKING OBTAINED BY FRIEDMAN'S TEST ON 30-DIMENSIONAL CEC2013 FUNCTIONS

Algorithms	Ranking
jDE	3.607
SaDE	4.410
JADE	2.642
EPSDE	4.857
CoDE	2.982
ZEPDE	**2.500**

TABLE X
p-VALUES OBTAINED BY BONFERRONI–DUNN'S, HOLM'S, AND HOCHBERG'S PROCEDURES FOR THE COMPARED DE ALGORITHMS ON 30-DIMENSIONAL CEC2013 FUNCTIONS

ZEPDE v.s.	z	Unadjusted p	Bonferroni-Dunn p	Holm p	Hochberg p
jDE	2.2142	**0.0268**	0.1340	0.0804	0.0804
SaDE	3.8214	**0.0001326**	**0.00066**	**0.00053**	**0.00053**
JADE	0.2857	0.7750	1	0.7750	0.7750
EPSDE	4.7142	**2.4256E-6**	**1.2128E-5**	**1.2128E-5**	**1.2128E-5**
CoDE	0.9642	0.3349	1	0.6698	0.6698

The Wilcoxon test and Friedman's test are employed to demonstrate the differences between ZEPDE and the five improved DE variants. The results of statistical analysis are shown in Tables VIII and IX. Table VIII shows that ZEPDE can significantly outperforms jDE, SaDE, JADE, EPSDE, and CoDE on 16, 19, 10, 17, and 11 test functions, respectively. Meanwhile, Table IX shows that the overall performance of the proposed algorithm is the best among these six DE variants.

Bonferroni–Dunn's, Holm's, and Hochberg's procedures are also employed to detect the global differences among the different DE variants. Table X indicates that the average performance of ZEPDE is significantly better than those of EPSDE and SaDE. Although the overall performance of ZEPDE is not significantly better than those of jDE, JADE, and CoDE, it can be seen from the results of the Wilcoxon test and Friedman's test that the average performance of ZEPDE is better than those of compared DE variants.

D. Comparison With Five Improved DE on 50-Dimensional Problems (CEC2013)

In this section, 28 50-dimensional CEC2013 test functions are employed to test the performances of all compared DE algorithms. The experimental results are given in supplementary (Table A4) due to pages limit.

From the statistical analysis results of the Wilcoxon test presented in Table XI, it can be seen from Table XI that ZEPDE

TABLE XI
RESULTS OF WILCOXON TEST ON 50-DIMENSIONAL CEC2013 FUNCTIONS

ZEPDE v.s.	+	−	≈
jDE	16	6	6
SaDE	19	1	8
JADE	10	7	11
EPSDE	19	3	6
CoDE	12	6	10

TABLE XII
RANKING OBTAINED BY FRIEDMAN'S TEST ON 50-DIMENSIONAL CEC2013 FUNCTIONS

Algorithms	Ranking
jDE	3.517
SaDE	4.375
JADE	2.75
EPSDE	5.071
CoDE	3.017
ZEPDE	**2.267**

TABLE XIII
p-VALUES OBTAINED BY BONFERRONI–DUNN'S, HOLM'S, AND HOCHBERG'S PROCEDURES FOR THE COMPARED DE ALGORITHMS ON 30-DIMENSIONAL CEC2013 FUNCTIONS

ZEPDE v.s.	z	Unadjusted p	Bonferroni-Dunn p	Holm p	Hochberg p
jDE	2.5	**0.0124**	0.0620	**0.0372**	**0.0372**
SaDE	4.2142	**2.5056E-5**	**1.2528E-4**	**1.0022E-4**	**1.0022E-4**
JADE	0.9642	0.3349	1	0.3349	0.3349
EPSDE	5.6071	**2.0569E-8**	**1.0284E-7**	**1.0284E-7**	**1.0284E-7**
CoDE	1.5	0.1336	0.6680	0.2672	0.2672

significantly performs better than jDE, SaDE, JADE, EPSDE, and CoDE on 16, 19, 10, 19, and 12 test functions. However, the performance of ZEPDE is significantly worse than those of jDE, SaDE, JADE, EPSDE, and CoDE on six, one, seven, three, and six test functions.

The results of Friedman's test are shown in Table XII, it can be seen from Table XII that the average performance of ZEPDE is the best among these compared DE algorithms. Furthermore, the p-values obtained by Bonferroni–Dunn's, Holm's, and Hochberg's procedures shown in Table XIII. Table XIII indicates that the overall performance of ZEPDE is significantly better than those of jDE, SaDE, and EPSDE on 28 50-dimensional CEC2013 test functions. Overall, the average performance of ZEPDE is the best among all compared algorithms.

The above four experimental comparisons indicate that for 25 30-dimensional and 50-dimensional CEC2005 test functions, the overall performance of ZEPDE is significantly better than those of jDE, SaDE, and EPSDE and is better than those of JADE and CoDE. However, the exploitation capability of JADE is better than that of ZEPDE because JADE uses a greedy mutation strategy. The overall performance (i.e., exploration and exploitation capacities) of ZEPDE is better than that of CoDE. For 28 30-dimensional and 50-dimensional CEC2013 test functions, exploitation capacity of algorithm may be important because the shifted global optimum of all test functions are randomly distributed in $[-80, 80]^D$ (search range is $[-100, 100]^D$). The experimental

TABLE XIV
RESULTS OF THE FRIEDMAN AND IMAN–DAVENPORT TEST WITH DIFFERENT NP

Friedman value	χ^2 value	p-value	Iman-Davenport value	F_F value	p-value
5.256	9.488	**0.262**	1.331	2.466	**0.264**

TABLE XV
RANKING OBTAINED BY FRIEDMAN'S TEST UNDER DIFFERENT NP

	$NP=50$	$NP=75$	$NP=100$	$NP=125$	$NP=150$
Ranking	3.3	2.88	**2.46**	3	3.36

TABLE XVI
RESULTS OF KRUSKAL–WALLIS TEST UNDER DIFFERENT NP

	$NP=50$	$NP=75$	$NP=100$	$NP=125$	$NP=150$
K+	13	9	11	8	12
K−	6	**2**	**2**	5	7
K≈	6	14	12	12	6

TABLE XVII
RESULTS OF THE FRIEDMAN AND IMAN–DAVENPORT TEST WITH DIFFERENT SETP

Friedman value	χ^2 value	p-value	Iman-Davenport value	F_F value	p-value
9.853	15.507	**0.275**	1.244	1.986	**0.276**

results shown in Sections V-C and V-D indicate that the overall performance of ZEPDE is significantly better than those of jDE, SaDE, and EPSDE. The results also show that the exploitation capability of ZEPDE is not significantly worse than that of JADE and is better than that of CoDE on most of the test functions. Overall, the average performance of ZEPDE is better than those of the five compared DE variants, especially when the test problems have high dimensions.

E. Analysis of Parameter Settings

In this section, 25 30-dimensional CEC2005 test functions are utilized to assess the robustness of the proposed algorithm and three nonparametric statistical tests (i.e., Friedman's test [46], Iman–Davenport's test [48], and Kruskal–Wallis test with multiple comparisons) are employed to analyze the sensitivity of population size, Setp, Msp of each mutation strategy in each generation, Bset, sigma combination, and NZ. The maximum number of FEs is set to be 300 000 and all obtained results are based on 30 independent runs for each compared DE variant in all experiments.

1) Impact of Population Size: In this experiment, the impact of population size is investigated due to the fact that appropriate population size can achieve good balance between exploration and exploitation capabilities during the entire evolution process. Some of the parameter settings are the same as in Section V-A except for NP, which is set to be 50, 75, 100, 125, and 150 in this experiment. The mean and standard deviation values have been provided in the supplementary attachment (Table A5) due to space limitation.

Friedman's [46] and Iman–Davenport's tests [48] are employed to test whether the performance of ZEPDE is significantly affected by different population size settings. Additionally, the Friedman's test [46] and Kruskal–Wallis test

TABLE XVIII
RANKING OBTAINED BY FRIEDMAN'S TEST UNDER DIFFERENT SETP

	Setp=0	Setp=0.05	Setp=0.1	Setp=0.15	Setp=0.2	Setp=0.25	Setp=0.3	Setp=0.35	Setp=0.4
Ranking	5.96	5.1	5.18	4.34	**4.26**	4.54	4.58	5.32	5.72

TABLE XIX
RESULTS OF KRUSKAL–WALLIS TEST UNDER DIFFERENT SETP

	Setp=0	Setp=0.05	Setp=0.1	Setp=0.15	Setp=0.2	Setp=0.25	Setp=0.3	Setp=0.35	Setp=0.4
K+	10	5	5	5	5	7	7	5	8
K−	6	3	**2**	**2**	**2**	3	4	4	4
K≈	9	17	18	18	18	15	14	16	13

with multiple comparisons are used to determine the suitable population size setting for the proposed algorithm.

Table XIV shows that the performance of ZEPDE is not extremely sensitive to different NP. Tables XV and XVI indicate that NP = 100 is the most appropriate parameter setting for the proposed algorithm.

2) Impact of Setp: To investigate the impact of Setp, which varies from 0 to 0.4 with a step of 0.05 in this experiment. The mean and standard deviation values have been given in the supplementary attachment (Table A6) due to space limitation. Friedman's [46] and Iman–Davenport's tests [48] are employed to test whether the performance of ZEPDE is significantly affected by different settings of Setp. Friedman's test [46] and Kruskal–Wallis test with multiple comparisons are used to find the suitable parameter setting for the proposed algorithm. The other parameter settings of ZEPDE are the same as in Section V-A.

The results obtained by Friedman's [46] and Iman–Davenport's tests [48] are reported in Tables XVII and XVIII. Table XVII indicates that the performance of ZEPDE is significantly insensitive to different Setp. Meanwhile, it can be seen from Table XVIII that the performance of ZEPDE is ideal when Setp = 0.2. However, Table XIX shows that the performance of ZEPDE is similar when Setp is within [0.1–0.2]. Based on the results obtained by the above statistical analysis, Setp = 0.175, which is a median value between 0.15 and 0.2, is suitable for the proposed algorithm.

3) Impact of Msp: In this experiment, the parameter settings of the proposed algorithm are the same as those used in Section V-A, except for Msp, which varies from 0.01 to 0.09 with a step equal to 0.01. The mean and standard deviation values have been given in the supplementary attachment (Table A7) due to space limitation. Friedman's test [46] and Iman–Davenport's test [48] are used to demonstrate whether the performance of ZEPDE is significantly sensitive to different Msp. Additionally, the Friedman's test [46] and Kruskal–Wallis test with multiple comparisons are employed to identify the most suitable Msp in ZEPDE.

Table XX reveals no significant performance differences under different Msp in the proposed algorithm. Tables XXI and XXII indicate that Msp = 0.01 is a suitable choice.

TABLE XX
RESULTS OF THE FRIEDMAN AND IMAN–DAVENPORT TEST WITH DIFFERENT MSP

Friedman value	χ^2 value	p-value	Iman-Davenport value	F_F value	p-value
16.906	15.507	**0.031**	2.216	1.986	**0.028**

4) Impact of Bset: In this experiment, to demonstrate the effect of Bset, which varies from 0 to 1 in a step of 0.1. The other parameter settings of ZEPDE are the same as in Section V-A. The mean and standard deviation values have been given in the supplementary attachment (Table A8) due to space limitation. Friedman's test [46] and Iman–Davenport's test [48] are used to demonstrate whether the performance of ZEPDE is significantly different under different Bset. Table XXIII shows that the performance of ZEPDE is significantly sensitive to different Bset. Friedman's test [46] and Kruskal–Wallis test with multiple comparisons are employed to identify the suitable Bset in ZEPDE. From Tables XXIV and XXV, it can be seen that Bset = 0.35 (the median value of 0.3 and 0.4) may be an appropriate choice in the proposed algorithm.

5) Impact of Sigma Combinations: In this experiment, the influence of the combinations of σ_1, σ_2 on ZEPDE performance is investigated. The proposed algorithm employs the same parameter settings as those used in Section V-A, except for σ_1, σ_2, which varies from 0 to 0.8 in a step of 0.05. The mean and standard deviation values have been given in the supplementary attachment (Table A9) due to space limitation. Friedman's test [46] and Iman–Davenport's test [48] are used to demonstrate whether the performance of ZEPDE differs significantly in different sigma combinations. Table XXVI indicates that the performance of ZEPDE is not significantly influenced by different sigma combinations. Friedman's test [46] and Kruskal–Wallis test with multiple comparisons are also employed to analyze the suitable sigma combination for ZEPDE algorithm. From Table XXVII, it can be seen that the combination (0.5, 0.5) provides the best performance; however, Table XXVIII shows that the overall performance of combination (0.35, 0.35) is also excellent. Therefore, in consideration of the statistical analysis results (i.e., Friedman's and Kruskal–Wallis tests), the sigma of ZEPDE could be selected within the range of [0.25, 0.55].

6) Impact of the NZ: Change in NZ may influence the performance of ZEPDE because different zonings may result in different combinations of control parameters. To investigate the effect of NZ, NZ is set to be 1, 2, 4 (Fig. 1), 6, and 9. The zoning implementations are shown in Fig. 4. In the experiment, the proposed algorithm employs the same parameter settings

TABLE XXI
RANKING OBTAINED BY FRIEDMAN'S TEST UNDER DIFFERENT MSP

	Msp=0.01	Msp=0.02	Msp=0.03	Msp=0.04	Msp=0.05	Msp=0.06	Msp=0.07	Msp=0.08	Msp=0.09
Ranking	**3.72**	3.92	5	5.78	4.98	5.62	5.9	5.52	4.56

TABLE XXII
RESULTS OF KRUSKAL–WALLIS TEST UNDER DIFFERENT MSP

	Msp=0.01	Msp=0.02	Msp=0.03	Msp=0.04	Msp=0.05	Msp=0.06	Msp=0.07	Msp=0.08	Msp=0.09
K+	11	8	6	5	6	5	6	5	6
K−	**0**	1	1	4	5	3	3	3	4
K≈	14	16	18	16	14	17	16	17	15

TABLE XXIII
RESULTS OF THE FRIEDMAN AND IMAN–DAVENPORT TEST WITH DIFFERENT BSET

Friedman value	χ^2 value	p-value	Iman-Davenport value	F_F value	p-value
43.876	18.307	**3.46E-6**	5.109	1.870	**9.16E-7**

TABLE XXIV
RANKING OBTAINED BY FRIEDMAN'S TEST UNDER DIFFERENT BSET

	Bset=0	Bset=0.1	Bset=0.2	Bset=0.3	Bset=0.4	Bset=0.5	Bset=0.6	Bset=0.7	Bset=0.8	Bset=0.9	Bset=1
Ranking	8.16	6.26	5.62	**4.42**	4.58	4.58	5.24	5.24	6.26	7.66	7.98

TABLE XXV
RESULTS OF KRUSKAL–WALLIS TEST UNDER DIFFERENT BSET

	Bset=0	Bset=0.1	Bset=0.2	Bset=0.3	Bset=0.4	Bset=0.5	Bset=0.6	Bset=0.7	Bset=0.8	Bset=0.9	Bset=1
K+	3	5	4	7	**9**	5	6	8	5	4	6
K−	18	12	8	**1**	2	3	5	6	10	14	14
K≈	4	8	13	17	14	17	14	11	10	7	5

TABLE XXVI
RESULTS OF THE FRIEDMAN AND IMAN–DAVENPORT TEST WITH DIFFERENT SIGMA COMBINATIONS

Friedman value	χ^2 value	p-value	Iman-Davenport value	F_F value	p-value
18.181	24.996	**0.253**	1.223	1.694	**0.25**

TABLE XXVII
RANKING OBTAINED BY FRIEDMAN'S TEST UNDER DIFFERENT SIGMA COMBINATIONS

	(0.05,0.05)	(01,0.1)	(0.15,0.15)	(0.2,0.2)	(0.25,0.25)	(0.3,0.3)	(0.35,0.35)	(0.4,0.4)	(0.45,0.45)	(05,0.5)	(0.55,0.55)	(0.6,0.6)	(0.65,0.65)	(0.7,0.7)	(0.75,0.75)	(0.8,0.8)
Ranking	9.94	10.84	9.62	8.56	8.08	8	8.7	7.22	7.38	**6.92**	7.52	8.7	8.42	8.34	8.38	9.38

TABLE XXVIII
RESULTS OF KRUSKAL–WALLIS TEST UNDER DIFFERENT SIGMA COMBINATIONS

	(0.05,0.05)	(01,0.1)	(0.15,0.15)	(0.2,0.2)	(0.25,0.25)	(0.3,0.3)	(0.35,0.35)	(0.4,0.4)	(0.45,0.45)	(05,0.5)	(0.55,0.55)	(0.6,0.6)	(0.65,0.65)	(0.7,0.7)	(0.75,0.75)	(0.8,0.8)
K+	**7**	6	6	6	6	5	5	6	4	5	**7**	6	5	4	4	5
K−	12	9	8	4	4	3	**2**	3	4	4	4	4	4	4	4	4
K≈	6	10	11	15	15	17	18	16	17	16	14	15	16	17	17	16

as those used in Section V-A, expect for NZ. NZ (a)–(f) denote six ZEPDE variants that use the zoning types [Fig. 4(a)–(f)]. Moreover, 17 30-dimensional CEC2005 test (i.e., F1–F17) functions are used. The mean and standard deviation values have been given in the supplementary attachment (Table A10) due to space limitation. Friedman's test [46] and Iman–Davenport's test [48] are used to demonstrate whether the performance of ZEPDE is significantly influenced by different NZ values. Table XXIX indicates that the performance of ZEPDE is significantly affected by different NZ values and can be improved by increasing NZ. Moreover, from the results obtained by the Friedman's test [46] and Kruskal–Wallis test with multiple comparisons shown in Tables XXX and XXXI. Table XXX shows that NZe provides the best performance. Table XXXI indicates that ZEPDE is not significantly worse than other ZEPDE variants on any test functions. Therefore, to achieve balance between algorithm performance and run time, NZ = 4 is selected for the proposed algorithm.

TABLE XXIX
RESULTS OF THE FRIEDMAN AND IMAN–DAVENPORT TEST WITH DIFFERENT NZ

Friedman	χ^2	p-value	Iman-Davenport	F_F	p-value
18.202	12.592	**0.0057**	3.475	2.195	**0.0037**

TABLE XXX
RANKING OBTAINED BY FRIEDMAN'S TEST UNDER DIFFERENT NZ

	NZa	NZb	NZc	ZEPDE	NZd	NZe	NZf
Ranking	5.647	4.470	4.5	3.235	3.500	**3.147**	3.500

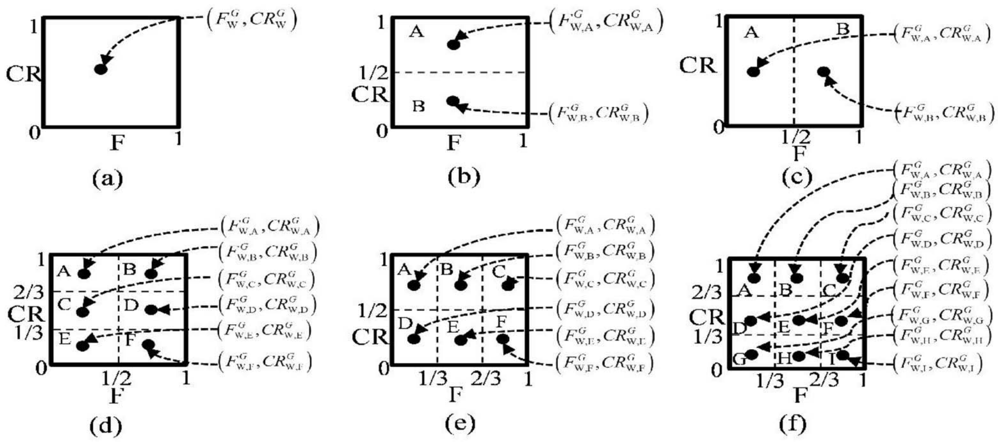


Fig. 4. Zoning types of control parameters for NZs is 1 (a), 2 (b and c), 6 (d and e), and 9 (f).

TABLE XXXI
RESULTS OF KRUSKAL–WALLIS TEST UNDER DIFFERENT NZ

	NZa	NZb	NZc	ZEPDE	NZd	NZe	NZf
K+	1	1	6	1	4	4	**7**
K−	6	4	5	**0**	3	2	2
K≈	10	12	6	16	10	11	8

TABLE XXXII
RESULTS OF THE FRIEDMAN AND IMAN IMAN–DAVENPORT TESTS WITH DIFFERENT ZEPDE VARIANTS

Friedman value	χ^2 value	p-value	Iman-Davenport value	F_F Value	p-value
15.7226	11.070	**0.00768**	3.631	2.328	**0.00518**

TABLE XXXIII
RANKING OBTAINED BY FRIEDMAN'S TEST UNDER DIFFERENT ZEPDE VARIANTS

	ZEPDE1	ZEPDE2	ZEPDE3	ZEPDE4	ZEPDE5	ZEPDE
Ranking	3.588	3.705	4.647	2.941	3.823	**2.294**

TABLE XXXIV
RESULTS OF KRUSKAL–WALLIS TEST UNDER DIFFERENT ZEPDE VARIANTS

	ZEPDE1	ZEPDE2	ZEPDE3	ZEPDE4	ZEPDE5	ZEPDE
K+	0	3	0	5	3	8
K−	6	8	12	6	11	2
K≈	11	6	5	6	3	7

TABLE XXXV
RESULTS OF KRUSKAL–WALLIS TEST UNDER DIFFERENT ZEPDE VARIANTS ON THREE CEC2005 FUNCTIONS

Function	ZEPDE1	ZEPDE2	ZEPDE3	ZEPDE4	ZEPDE5
F4	K−	K−	K−	**K+**	K−
F8	K−	K−	K≈	**K+**	K−
F9	**K+**	K≈	K−	K−	K−

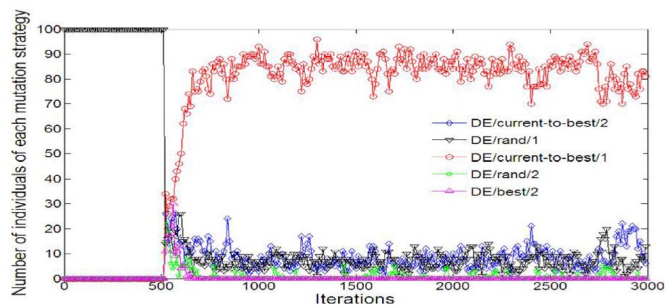


Fig. 5. Evolution curves of the mutation strategies of ZEPDE for F4.

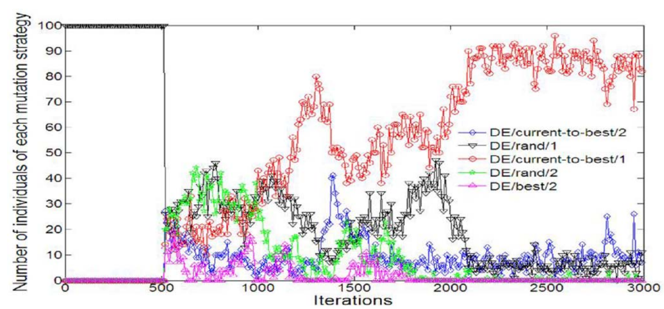


Fig. 6. Evolution curves of the mutation strategies of ZEPDE for F8.

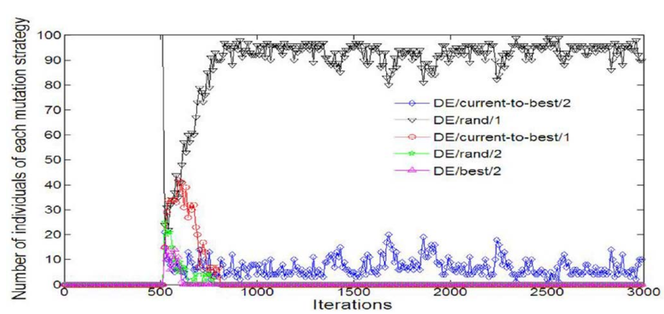


Fig. 7. Evolution curves of the mutation strategies of ZEPDE for F9.

F. Self-adaptive mutation strategy of ZEPDE

To demonstrate the effect of the self-adaptive mutation strategies of ZEPDE, five ZEPDE variants (i.e., ZEPDE1, ZEPDE2, ZEPDE3, ZEPDE4, and ZEPDE5 using only DE/rand/1, DE/rand/2, DE/best/2, DE/current-to-best/1, or DE/current-to-best/2 mutation strategy) with a single mutation strategy are employed to solve 17 30-dimensional CEC2005 test functions (i.e., F1–F17). The parameter settings of these variants are the same as ZEPDE in Section V-A. The mean and standard deviation values have been given in the supplementary attachment (Table A11) due to space limitation.

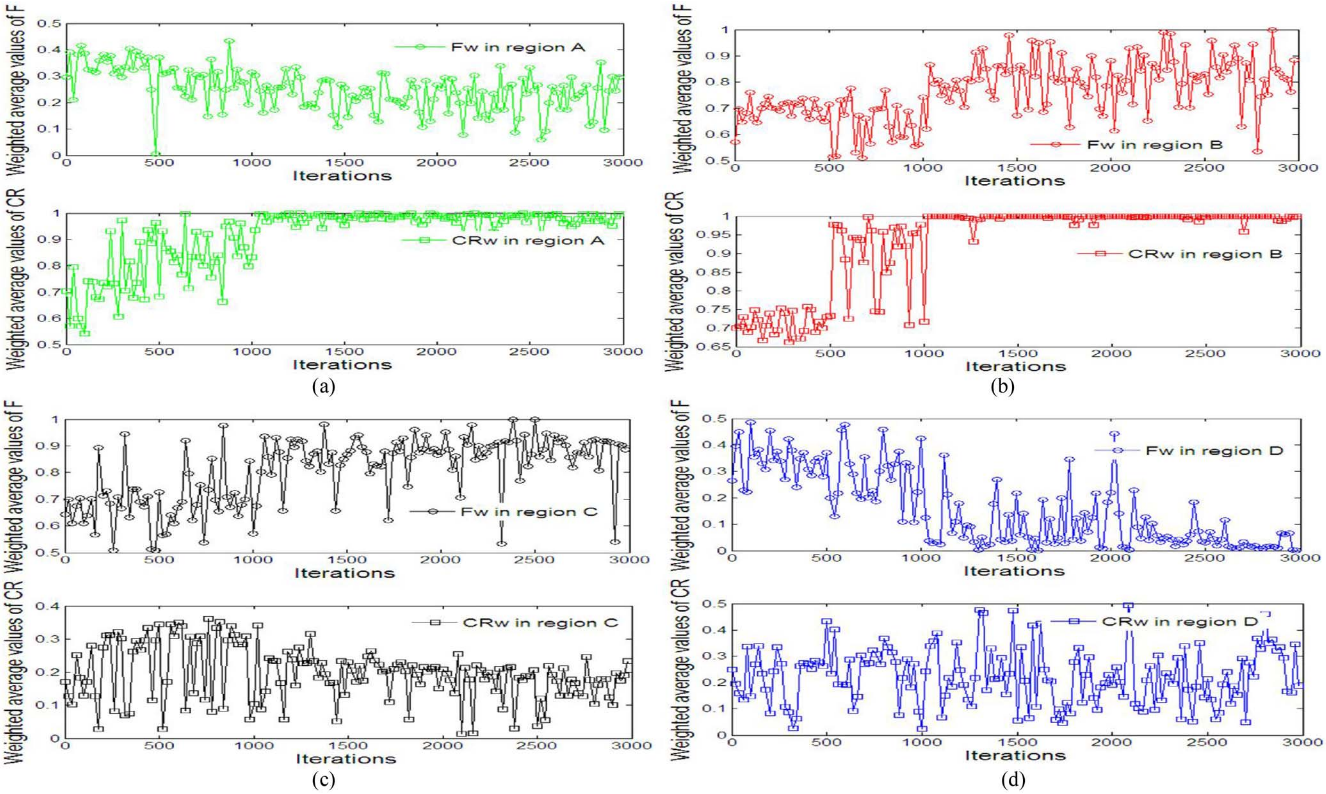


Fig. 8. Evolution curves of the weighted average values of F and CR in four regions. (a) Evolution curve of the weighted average values of F and CR in region A. (b) Evolution curve of the weighted average values of F and CR in region B. (c) Evolution curve of the weighted average values of F and CR in region C. (d) Evolution curve of the weighted average values of F and CR in region D.

Friedman's test [46] and Iman–Davenport's test [48] are employed to test whether the performance of ZEPDE is significantly affected by different mutation strategies. It can be seen from Table XXXII that the performance of the proposed algorithm is significantly influenced by the choice of mutation strategy. Meanwhile, from the results of statistical analysis obtained by Friedman's test [46] and Kruskal–Wallis test with multiple comparisons shown in Tables XXXIII and XXXIV, it can be seen that the performance of ZEPDE is better than those of ZEPDE variants with a single mutation strategy. In addition, to further demonstrate the effectiveness of the self-adaptive mutation strategies of the proposed algorithm, three test functions (i.e., F4, F8, and F9 selected from CEC2005) are used to test the adaptation of mutation strategies, and three typical evolution curves of mutation strategies of ZEPDE are shown in Figs. 5–7. As shown in Figs. 5–7 and Table XXXV, it can be observed that the evolution of the mutation strategies of ZEPDE conforms to the results presented in Table XXXV.

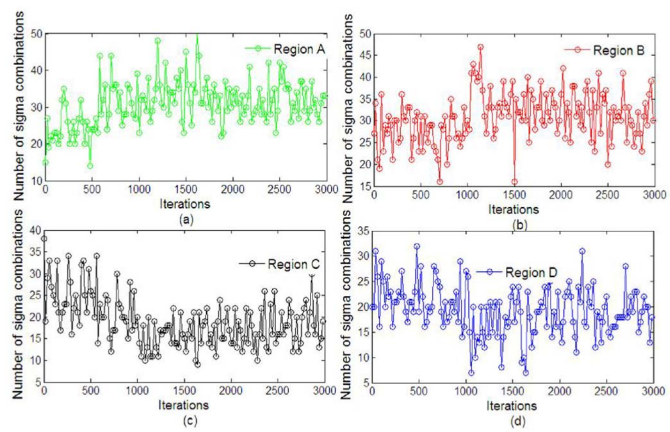


Fig. 9. Curves of the number of sigma combinations in each region. (a) Curve of the number of sigma combinations in region A. (b) Curve of the number of sigma combinations in region B. (c) Curve of the number of sigma combinations in region C. (d) Curve of the number of sigma combinations in region D.

G. Self-Adaptive Control Parameters of ZEPDE

In this experiment, a CEC2005 test function (i.e., F3) is employed to analyze the self-adaptive properties of control parameters. some of the parameter settings of ZEPDE are the same as in Section V-A. The evolution curves of the weighted average of F and CR (i.e., $F^G_{W,A}$, $\mathrm{CR}^G_{W,A}$, $F^G_{W,B}$, $\mathrm{CR}^G_{W,B}$, $F^G_{W,C}$, $F^G_{W,D}$, and $\mathrm{CR}^G_{W,D}$) in each region are shown in Fig. 8. Fig. 8(a) indicates that the weighted average value of F gradually decreases during the entire evolution process, and the weighted average value of CR gradually increases when generation is less than 1000. Fig. 8(b) shows that the weighted average value of F gradually increases throughout the entire evolution process, and the weighted average value of CR gradually increases when generation is less than 1000. Fig. 8(c) indicates that the weighted average value of F gradually increases during the evolution process, and the weighted average value of CR gradually decreases. Fig. 8(d) shows that the weighted average value of F gradually decreases throughout the entire evolution process and the weighted average value of CR fluctuates in the evolution process. Overall, it is observed that control parameter combinations of each zoning gradually play a different role throughout the entire evolution process. Therefore, ZEPDE can well balance between the exploitation and exploration capabilities.

H. Number of Control Parameter Combinations in Each Zoning

To show the changes in the number of control parameter combinations in each zoning during the evolution process, one 30-dimensional CEC2005 test function (i.e., F3) is used. Fig. 9 shows that the number of control parameter combinations can be self-adaptively adjusted with the population evolution in each zoning.

VI. Conclusion

A self-adaptive differential evolution algorithm with ZEPDE is proposed in this paper. The performance of ZEPDE is compared with that of five famous DE variants on 25 30-dimensional and 50-dimensional CEC2005 test functions and 28 30-dimensional and 50-dimensional CEC2013 test functions. The experimental results show that the overall performance of ZEPDE is better than those of the other compared algorithms, especially when the test function has high dimensions. Several nonparametric statistical tests are also employed to evaluate the performances of all compared DE algorithms and analyze the reasonability of key parameter choice of ZEPDE. Several self-adaptive properties of the mutation strategies and control parameters of the proposed algorithm and the variations in the number of control parameter combinations in each zoning are discussed in this paper. From the experimental results, it is observed that the mutation strategies and control parameters of ZEPDE can automatically adapt during the evolution process, and the performance of ZEPDE is significantly affected by the NZs. Therefore, ZEPDE can quickly adapt to complex and unknown optimization environments and can maintain the search capabilities of control parameter combinations of different zonings during the entire evolution process.

References

[1] R. Storn and K. Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," International Computer Science Institute, Berkeley, CA, USA, Tech. Rep. TR-95-012, 1995.

[2] J. Liu and J. Lampinen, "On setting the control parameter of the differential evolution method," in *Proc. 8th Int. Soft Comput. MENDEL*, Brno, Czech Republic, 2002, pp. 11–18.

[3] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[4] F. Neri and V. Tirronen, "Recent advances in differential evolution: A survey and experimental analysis," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 61–106, 2010.

[5] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.

[6] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 124–141, Jul. 1999.

[7] J. Brest, V. Zumer, and M. S. Maucec, "Self-adaptive differential evolution algorithm in constrained real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 215–222.

[8] R. Mallipeddi, P. Suganthan, and Q. Pan, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1679–1696, 2011.

[9] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution—A Practical Approach to Global Optimization*. Berlin, Germany: Springer, 2005.

[10] J. Lampinen and I. Zelinka, "On stagnation of the differential evolution algorithm," in *Proc. 6th Int. Mendel Conf. Soft Comput.*, Brno, Czech Republic, 2002, pp. 76–83.

[11] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Comput.*, vol. 9, no. 6, pp. 448–462, 2005.

[12] R. Storn, "On the usage of differential evolution for function optimization," in *Proc. Biennial Conf. North Amer. Fuzzy Inf. Process. Soc. (NAFIPS)*, Berkeley, CA, USA, 1996, pp. 519–523.

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

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

[15] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," *Adv. Intell. Syst. Fuzzy Syst. Evol. Comput.*, vol. 10, pp. 293–298, Aug. 2002.

[16] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Edinburgh, Scotland, 2005, pp. 506–513.

[17] K. Zielinski, P. Weitkemper, and R. Laur, "Parameter study for differential evolution using a power allocation problem including interference cancellation," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 1857–1864.

[18] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Seattle, WA, USA, 2006, pp. 485–492.

[19] A. W. Iorio and X. Li, "Solving rotated multiobjective optimization problems using differential evolution," in *AI 2004: Advances in Artificial Intelligence*. Berlin, Germany: Springer, 2005, pp. 861–872.

[20] S. Das, A. Abraham, and U. K. Chakraborty, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.

[21] 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.

[22] H. A. Abbass, "The self-adaptive pareto differential evolution algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI, USA, 2002, pp. 831–836.

[23] C. C. Vu, L. T. Bui, and H. A. Abbass, "DEAL: A direction-guided evolutionary algorithm," in *Simulated Evolution and Learning.* Berlin, Germany: Springer, 2012, pp. 148–157.

[24] D. Zaharie, "Parameter adaptation in differential evolution by controlling the population diversity," in *Proc. Int. Workshop Symb. Numeric Algorithms Sci. Comput.*, Timisoara, Romania, 2002, pp. 385–397.

[25] M. G. Omran, A. Salman, and A. P. Engelbrecht, "Self-adaptive differential evolution," in *Computational Intelligence and Security*. Berlin, Germany: Springer, 2005, pp. 192–199.

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

[27] A. Ghosh, S. Das, and A. Chowdhury, "An improved differential evolution algorithm with fitness-based adaptation of the control parameters," *Inf. Sci.*, vol. 181, no. 18, pp. 3749–3765, 2011.

[28] W. Gong, Z. Cai, and C. X. Ling, "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 2, pp. 397–413, Apr. 2011.

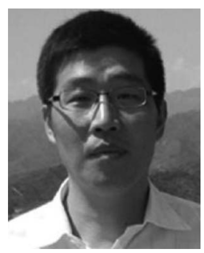
[29] J. Teo, "Differential evolution with self-adaptive populations," in *Knowledge-Based Intelligent Information and Engineering Systems*. Berlin, Germany: Springer, 2005, pp. 1284–1290.

[30] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," *Soft Comput.*, vol. 10, no. 8, pp. 673–686, 2006.

[31] N. S. Teng, J. Teo, and M. H. A. Hijazi, "Self-adaptive population sizing for a tune-free differential evolution," *Soft Comput.*, vol. 13, no. 7, pp. 709–724, 2009.
[32] V. Tirronen and F. Neri, "Differential evolution with fitness diversity self-adaptation," in *Nature-Inspired Algorithms for Optimisation*. Berlin, Germany: Springer, 2009, pp. 199–234.
[33] W. Zhu, Y. Tang, and J.-A. Fang, "Adaptive population tuning scheme for differential evolution," *Inf. Sci.*, vol. 223, pp. 164–191,Feb. 2012.
[34] J. Brest and M. S. Maučec, "Population size reduction for the differential evolution algorithm," *Appl. Intell.*, vol. 29, no. 3, pp. 228–247, 2008.
[35] A. Zamuda and J. Brest, "Population reduction differential evolution with multiple mutation strategies in real world industry challenges," in *Swarm and Evolutionary Computation*. Berlin, Germany: Springer, 2012, pp. 154–161.
[36] Y. Zhong and L. Zhang, "Remote sensing image subpixel mapping based on adaptive differential evolution," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 5, pp. 1306–1329, Oct. 2012.
[37] C.-H. Chen, C.-J. Lin, and C.-T. Lin, "Nonlinear system control using adaptive neural fuzzy networks based on a modified differential evolution," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 4, pp. 459–473, Jul. 2009.
[38] F. Neri and E. Mininno, "Memetic compact differential evolution for Cartesian robot control," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 54–65, May 2010.
[39] A. Zamuda, J. Brest, and B. Bošković, "Differential evolution for parameterized procedural woody plant models reconstruction," *Appl. Soft Comput.*, vol. 11, no. 8, pp. 4904–4912, 2011.
[40] N. Salvatore, A. Caponio, and F. Neri, "Optimization of delayed-state Kalman-filter-based algorithm via differential evolution for sensorless control of induction motors," *IEEE Trans. Ind. Electron.*, vol. 57, no. 1, pp. 385–394, Jan. 2010.
[41] R. Amin, J. Tang, M. Ellejmi, S. Kirby, and H. A. Abbass, "Trading-off simulation fidelity and optimization accuracy in air-traffic experiments using differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Beijing, China, 2014, pp. 475–482.
[42] P. N. Suganthan, N. Hansen, and J. J. Liang, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Dept. Comput. Sci., Kanpur Genet. Algorithms Lab., Indian Inst. Technol., Kanpur, India, KanGAL Rep. 2005005, 2005.
[43] J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Dept. Econ., Nanyang Technol. Univ., Singapore, Tech. Rep. 201212, 2013.
[44] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometr. Bull.*, vol. 1, no. 6, pp. 80–83, 1945.
[45] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *J. Amer. Stat. Assoc.*, vol. 47, no. 260, pp. 583–621, 1952.
[46] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937.
[47] O. J. Dunn, "Multiple comparisons among means," *J. Amer. Stat. Assoc.*, vol. 56, no. 293, pp. 52–64, 1961.
[48] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

Qinqin Fan received the B.S. degree from Wuhan Institute of Technology, Wuhan, China, in 2007, and the M.S. degree from East China University of Science and Technology, Shanghai, China, in 2011, where he is currently pursuing the Ph.D. degree from the Department of Automation.

His current research interests include differential evolution algorithm, particle swarm optimization, constrained optimization, multiobjective optimization, and their real-world applications.



Xuefeng Yan received the Ph.D. degree from Zhejiang University, Hangzhou, China.

He is currently a Professor with East China University of Science and Technology, Shanghai, China. His current research interests include complex chemical process modeling, optimizing and controlling, process monitoring, fault diagnosis, and intelligent information processing.