

Research Article

Multiscale Cooperative Differential Evolution Algorithm

Yongzhao Du,^{1,2} Yuling Fan,¹ Xiaofang Liu,¹ Yanmin Luo ,³ Jianeng Tang ,¹ and Peizhong Liu ^{1,2}

¹College of Engineering, Huaqiao University, Quanzhou 362021, China

³College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China

²Research Center of Apply Statistics and Big Data, Huaqiao University, Xiamen 361021, China

Correspondence should be addressed to Peizhong Liu; pzliu@hqu.edu.cn

Received 20 June 2019; Revised 2 August 2019; Accepted 26 August 2019; Published 17 December 2019

Academic Editor: Roman Bartak

Copyright © 2019 Yongzhao Du et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A multiscale cooperative differential evolution algorithm is proposed to solve the problems of narrow search range at the early stage and slow convergence at the later stage in the performance of the traditional differential evolution algorithms. Firstly, the population structure of multipopulation mechanism is adopted so that each subpopulation is combined with a corresponding mutation strategy to ensure the individual diversity during evolution. Then, the covariance learning among populations is developed to establish a suitable rotating coordinate system for cross operation. Meanwhile, an adaptive parameter adjustment strategy is introduced to balance the population survey and convergence. Finally, the proposed algorithm is tested on the CEC 2005 benchmark function and compared with other state-of-the-art evolutionary algorithms. The experiment results showed that the proposed algorithm has better performance in solving global optimization problems than other compared algorithms.

1. Introduction

The differential evolution (DE) is a bionic intelligence method proposed by American scholars Rainer Storn and Kenneth Price in 1995, simulating survival of the fittest [1, 2]. The algorithm adopts mutation, crossover, and selection operations to mimic genetic mutations during biological evolution and retains highly adaptable individuals for optimal solutions. Aiming at the problems of both population convergence stagnation and premature convergence, researchers mainly focus on three aspects of control parameter setting and mutation strategy selection [3–6], crossover operation [7–9], and population structure [10–12] to improve the algorithm performance. The DE has been widely concerned by researchers because of its simple coding, convergence, and strong robustness. It has been applied in many fields such as industrial control [13], antenna design [14], power system [15], image processing [16], and so on.

The parameter control and evolutionary strategy selection are mainly discussed in two aspects of DE. On the one hand, control parameter settings for the scaling factor

F , crossover probability CR , and population size NP [17]. On the other hand, different strategies for different optimization problems [18], we need to choose the most suitable strategy. The parameter setting affects the population diversity [19], the development ability of the early period, and the convergence of the later period [20]. The choice of evolutionary strategy is the key step to determine the balance between exploration and convergence of DE, and different evolution strategies will show different surveying capabilities and the convergence tendencies. At the same time, diverse crossover operations have diverse effects on seeking global optimization. Although the traditional binomial crossover operation has a certain role, it is more dependent on the cross coordinate system and is widely used. In addition, the population structure is also an important indicator for the algorithm performance. If the population size is too small, it will easily lead to the loss of effective alleles, thereby reducing the generation of competitive individuals. In contrast, if the population size is too large, the possibility of correct search direction by the algorithm will be reduced [11].

Due to premature convergence, parameter control, and strategy improvement, crossover operation and population structure attract increasing attention to improve the performances of DEs [17, 21–24]. Therefore, numerous DE-improvement algorithms [25, 26] have been proposed constantly based on parameters and strategies, such as parameter-adaptive jDE [21], JADE using “current-to-pbest/1” strategy and adaptive parameters [24], SaDE using adaptive difference strategy [23], CoDE suiting for experimental individual algebra strategy and control parameters [27], EPSDE for mutation strategy and control parameter [17], TDE for triangular mutation strategy [28], and super-fit multistandard adaptive SMADE [29]. Moreover, the DE is improved based on the crossover operation, such as ODE [30] adopting orthogonal crossover operators and CoBiDE [31] using covariance learning and bimodal distribution of parameters. And DE is improved based on population structure, such as SPSRDEMMS [32] for multivariation strategies, MPEDE [33] for multipopulations and strategy sets, master-slave model [34], island model [35], cellular model [36], level model [37], and pool model [38]. In recent years, population segregation techniques have been used to improve evolutionary algorithms, including particle swarm optimization, genetic algorithm, and evolutionary algorithm [39–44].

To further improve the convergence and reduce the population stagnation, a multiscale cooperative differential evolution (MCDE) algorithm is proposed. In terms of parameter setting, the scaling factor F and the crossover probability CR are mainly adjusted based on the literature [24]. In the selection of mutation strategy, the MCDE selects “current-to-pbest/1,” “current-to-rand/1,” and “rand/1” as mutation strategy groups. In the initial phase, the evolutionary population was divided into multiple subpopulations, and one subpopulation was selected as the experimental population to test the mutation strategy with better evolutionary results. In the evolutionary phase, the global search capability is continuously promoted by establishing a constant rotation of the cross coordinate system and coordinating among multiple subpopulations. In the end, the best individual that remains is used as the optimal solution. In CEC 2005, 30-dimension and 50-dimension simulation tests were conducted and compared with contemporary evolutionary algorithms; MCDE was found to have more significant effects.

The paper is organized as follows. Section 2 briefly introduces the standard DE algorithm. Section 3 elaborates on the algorithm improvement. Section 4 analyzes the significance of the proposed algorithm through experimental data. Section 5 gives a summary.

2. Standard Differential Evolution Algorithm

DE can be regarded as greedy evolution algorithm based on real number coding and global optimization. In the evolutionary phase, three iteration processes of mutation, crossover, and selection are performed until the stop condition is satisfied. The fitness function $f(x)$ is utilized to evaluate the quality and the best individual is recorded.

2.1. Initialization. Assuming that the population size is NP and the dimension of the feasible solution space is D , x^G is employed to represent the evolution population of G generation. Each individual is composed of D -dimensional parameters, which can be expressed as

$$x_i^G = \{x_{i,1}^G, x_{i,2}^G, \dots, x_{i,D}^G\}, \quad i \in \{1, 2, \dots, NP\}, \quad (1)$$

where $x_{i,j}^G \in (x^L, x^H)$ and x^L and x^H represent the upper and lower bounds of the individual, respectively.

2.2. Mutation Operation. The individual x_i^G in the parent population generates a variant individual v_i^G by a mutation strategy. “DE/rand/1” indicates that the DE chooses a random perturbation individual to mutate. The expression is as follows:

$$v_i^G = x_{r1}^G + F \cdot (x_{r2}^G - x_{r3}^G), \quad r1, r2, r3 \in \{1, 2, \dots, NP\}, \quad (2)$$

where $r1 \neq r2 \neq r3$ and $r1, r2$, and $r3$ are a randomly generated mutation individual. The scaling factor F is chosen from $[0, 1]$.

2.3. Crossover Operation. The main function of the crossover operation is that the generated variant individuals cross with individuals in the original population to generate new crossover individuals. The DE adopts binomial crossover scheme. The crossover operation is as follows:

$$u_{i,j}^G = \begin{cases} v_{i,j}^G, & \text{if } \text{rand}_j \leq CR \text{ or } j = j_{\text{rand}}, \\ x_{i,j}^G, & \text{otherwise, } j = 1, \dots, D, \end{cases} \quad (3)$$

where $\text{rand}_j \in [0, 1]$, j_{rand} is chosen from $\{1, 2, \dots, D\}$, and crossover probability CR is $[0, 1]$.

2.4. Selection Operation. The selection operation mainly adopts the greedy selection mode of the survival of the fittest, making the offspring always superior to or equal to the parent individual x_i . When the fitness value of the new individual u_i is better than that of the objective individual, the new individuals u_i will be accepted by the population. Otherwise, x_i still remains in the next generation population and continues to perform mutation and crossover operations as the objective individual in the next iterative calculation so that the population will always evolve toward the optimal solution. The selection operation is for minimization fitness value as follows:

$$x_i^{G+1} = \begin{cases} u_i^G, & \text{if } f(u_i^G) \leq f(x_i^G), \\ x_i^G, & \text{otherwise,} \end{cases} \quad (4)$$

where $f(x)$ is the objective function to be optimized.

3. Multiscale Cooperative Differential Evolution Algorithm

In the proposed algorithm, we divide the whole population into multiple subpopulations and give corresponding

mutation strategies. Then, the cross coordinate system of each subpopulation is established by covariance learning and parameter adaptation of evolutionary subpopulation. Finally, the obtained crossover individual is selected and the individuals with better fitness are retained to make the whole population search forward to the global optimal solution.

3.1. Multiscale Mutation Strategy Integration Method. In recent years, because different mutation strategies are suitable for solving different optimization functions, some researchers mainly focus on multiple mutation strategies method [23, 24]. Even for a specific optimization problem, the most appropriate mutation strategy may be different at different stages of evolution. Therefore, mutation strategy is an important indicator to ensure significant results in the DE. During evolution, this paper selects “current-to-pbest/1,” “current-to-rand/1,” and “rand/1” as the multiscale mutation strategy set because of the different performance requirements for the mutation strategy. The individuals of “current-to-rand/1” and “rand/1” involved in mutation are all selected in a random manner so that global optimization can be performed in the early stages of evolution. “current-to-pbest/1” seeks the global optimal solution through the current best population individual. During evolution, the search range can be reduced to the vicinity of the optimal solution and the convergence speed can be accelerated.

Current-to-pbest/1:

$$v_i^G = x_i^G + F \cdot (x_{\text{pbest}}^G - x_i^G + x_{r1i}^G - x_{r2i}^G). \quad (5)$$

Current-to-rand/1:

$$v_i^G = x_i^G + K \cdot (x_{r1i}^G - x_i^G) + F \cdot (x_{r2i}^G - x_{r3i}^G). \quad (6)$$

rand/1:

$$v_i^G = x_{r1i}^G + F \cdot (x_{r2i}^G - x_{r3i}^G), \quad (7)$$

where x_{pbest}^G is uniformly chosen as one of the top p individuals in the current population with $\text{pbest} \in (0, 1]$. Because the three mutation strategies have their own advantages, there are some differences. Therefore, the population multiscale mechanism is introduced in this paper. The whole population Pop is divided into three subpopulation Pop₁, Pop₂, and Pop₃. Pop₁ with the largest population size is determined as the experimental population and combined with the corresponding mutation strategy. During evolution, the experimental population is allocated to mutation strategy with better evolution results. The population structure is expressed as follows:

$$\begin{aligned} \text{Pop} &= \sum_{i=1,2,3} \text{Pop}_i, \\ \text{NP}_i &= \sigma_i * \text{NP}, \\ \sum_{i=1,2,3} \sigma_i &= 1, \end{aligned} \quad (8)$$

where we assume that Pop₁ is an experimental population, NP is the population size, σ represents population size ratio, $\sigma_1 > \sigma_2 = \sigma_3$ and $\sigma_i \in [0, 1]$.

After the population structure is well designed, the distribution rules of the subpopulations should be given. First, subpopulation Pop₁, Pop₂, and Pop₃ incorporate corresponding mutation strategies. Then, the population undergoes mutation, crossover, and selection operations. Finally, the total number bd_i of superior individuals retained after each subpopulation evolution is counted. That is, the superior rate br_i of the subpopulation can be expressed as

$$\text{br}_i = \frac{\text{bd}_i}{\text{NP}_i}, \quad i = 1, 2, 3. \quad (9)$$

The superior rate br_i of each generation of subpopulation is calculated, and the subpopulation is reallocated for three mutation strategies according to the superior rate br_i in the next generation initialization stage. The multiscale mutation strategy set method makes full use of the advantages of the three mutation strategies to regulate and balance the contradiction between the population diversity and the convergence speed, which can be seen from the experimental results of the latter. In the first generation, the subpopulations randomly assign a mutation strategy. At the end of the first generation, we calculate the subpopulations superior rate by equation (9). The maximum superior rate stands for the best mutation strategy in this generation. Assume that the first-generation mutation strategy “current-to-pbest/1” has the highest superior rate, and the second generation assigns Pop₁ to it. The remaining subpopulations Pop₂ and Pop₃ randomly assign a mutation strategy (“current-to-rand/1” or “rand/1”).

3.2. Covariance Learning. The aforementioned crossover operators mainly depend on the coordinate system, while the distribution information of the population reflects the direction of evolution to some extent [20]. During evolution, the distribution of population is often neglected, leading to the possibility of the population falling into local optimum and premature convergence. In this paper, variance and covariance are utilized to analyze population distribution and form covariance matrix to reflect population diversity information. Therefore, the systematic use of covariance matrix can reduce the dependence on coordinate system and the interaction between variables. Covariance matrix learning includes two related technologies: the feature decomposition and coordinate transformation of covariance matrix. The covariance matrix learning steps are as follows:

Step 1. Calculate covariance matrix C of subpopulations.

Step 2. Get the eigenvalue λ and feature vector matrix R of covariance.

Step 3. Update the objective individual and the variant individual through the feature-based cooperative system.

$$\begin{cases} x_i^G = R^{-1} x_i^G = R^T x_i^G, \\ v_i^G = R^{-1} v_i^G = R^T v_i^G. \end{cases} \quad (10)$$

Step 4. Populations with better fitness for crossover and selection operations are retained and rotated back to the original coordinate system.

$$u_i^G = R \cdot u_i^G. \quad (11)$$

Based on the above four steps, we establish the population feature coordinate system. Figure 1(a) shows the initial coordinate system of population evolution, and Figure 1(b) shows the feature coordinate system. By analyzing the population feature, we obtain the ox_1x_2 coordinate system and discover that we can find the global optimum faster.

3.3. Adaptive Control Parameter Settings. At present, researchers have proposed many effective parameter adaptation methods [21, 23, 24]. The combination of different control parameters and mutation strategies for the optimization problem will yield different results. In this paper, each scale strategy has its own control parameters, and different technologies are applied to the algorithm. The method in [24] is more suitable for the algorithm, so it adapts to the algorithm by improving its technology.

During evolution, scaling factor F plays a decisive role in the search range of base vectors. In standard DE algorithm, the value of F is a fixed value, which cannot be applied to solve all global optimization functions. In this paper, the scalar factor F mainly adopts the Cauchy inverse cumulative distribution function, assuming that $F_{i,j}$ represents the scale factor of each dimension in the individual. $F_{i,j}$ is expressed as follows:

$$F_{i,j} = \text{Cauchy}_{i,j}(Fm_j, 0.1), \quad (12)$$

where Fm_j is the position parameter of the Cauchy inverse cumulative distribution function and the scale factor of current individual and the initial value of Fm_j is set to 0.5. 0.1 indicates the scale parameter of the Cauchy inverse cumulative distribution function. To better apply to population evolution, the weighting factor c is introduced to combine the parent factor and the next generation factor. The current Fm_j is expressed as follows:

$$Fm_j = (1 - c) \cdot Fm_j + c \cdot \text{mean}_F(S_{F,j}), \quad (13)$$

where $c \in [0, 1]$ and parental scalar factor $S_{F,j}$ is calculated using the power mean. The power mean is expressed as follows:

$$\text{mean}_F(S_{F,j}) = \left(\frac{1}{NP} \cdot \sum_{NP} F^n \right)^{1/n}, \quad (14)$$

where n is the index value of the power mean, which is quantified to the influence of the parent's scaling factor on the offspring.

In the DE algorithm, the crossover probability CR determines the possibility that an objective individual inherits gene from variant individual v_i^G . In this paper, the crossover probability CR mainly adopts the normal distribution function, assuming that $CR_{i,j}$ represents the crossover probability of each dimension in the individual. $CR_{i,j}$ is expressed as follows:

$$CR_{i,j} = \text{rand } n_{i,j}(CRm_j, 0.1), \quad (15)$$

where CRm_j is the mean of individual crossover probability and the initial value is set to 0.5. The standard deviation of normal distribution is set to 0.1. To better inherit the parent gene, a weighting factor c is introduced to combine the parent crossover probability with the next generation crossover probability. CRm_j is expressed as follows:

$$CRm_j = (1 - c) \cdot CRm_j + c \cdot \text{mean}_{CR}(S_{CR,j}), \quad (16)$$

where $c \in [0, 1]$ and parental crossover probability $S_{CR,j}$ is calculated using the Lehmer mean. The Lehmer mean is as follows:

$$\text{mean}_{CR}(S_{CR,j}) = \frac{\sum_D CR^2}{\sum_D CR}. \quad (17)$$

The Lehmer mean method can flexibly adjust the value of CR according to the parent cross probability.

3.4. Algorithm Framework. The proposed algorithm combines multiscale strategy and covariance learning and introduces adaptive control parameters to lead the population to keep close to the global optimum. Based on the above analysis, the basic flow of MCDE is summarized as Algorithm 1.

4. Experimental Results and Analysis

The MCDE is tested on 25 benchmark functions of IEEE CEC 2005. The 25 benchmark functions mainly include unimodal function F_1 – F_5 , basic multimodal function F_6 – F_{12} , extended multimodal function F_{13} – F_{14} , and complex function F_{15} – F_{25} . For details, please refer to [45]. In this paper, the parameter setting of the MCDE is as follows: population size NP = 250 and subpopulation ratio $\sigma_1 = 0.6$, $\sigma_2 = \sigma_3 = 0.2$. Experimental environment: the operating system is win7 Professional 64 bit, CPU is core i7 (3.40 GHz), RAM is 8 GB, and the compiler is MATLAB R2014b.

4.1. Comparison with Improved DE Algorithm. To verify the performance of the MCDE, it is compared with the six classic DE-improvement algorithms: JADE [24], jDE [21], SaDE [23], EPSDE [17], CoDE [18], CoBiDE [31], and LSHADE [46]. JADE and jDE are representative algorithms and are heavily referenced. SaDE and EPSDE are based on multi-strategy improved algorithms. CoDE and CoBiDE are improved algorithms based on population structure. The experimental results of the above seven algorithms are shown in Table 1, where $D = 30$ and MaxFES = 300000. The form of the numerical values in the table is the mean error \pm standard deviation. “ $-/+/\approx$ ” means that the comparison algorithm is obviously better than, worse than, and similar to MCDE. Based on the data given in Table 1 and Figure 2, we can draw the following conclusions: (1) unimodal function F_1 – F_5 : among the comparison algorithms, JADE and LSHADE have the best effect. Because of the greedy strategy “current-to-pbest/1,” the algorithm can achieve fast convergence and high precision. However, the MCDE multiscale strategy achieves better results than JADE

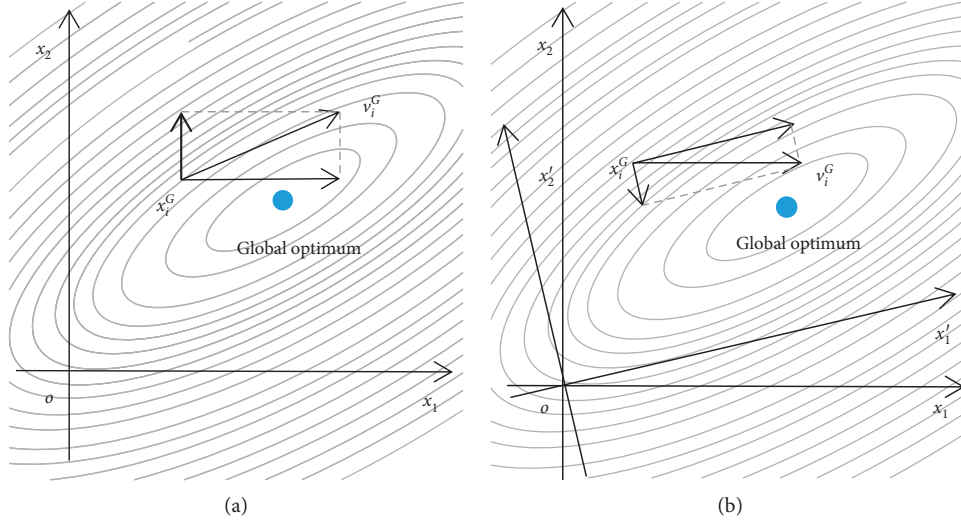


FIGURE 1: Population initial coordinate system and population feature coordinate system: (a) the initial coordinate system of population evolution; (b) the Eigen coordinate system of population evolution.

```

Input: population size NP, dimension D, Maximum number of evaluations MaxFES; the initial scale factor  $Fm_j$ , the initial crossover probability  $CRm_j$ ; weighting factor  $c = 0.1$ , power mean  $n = 4$ ; initialize the population  $Pop_1$ ,  $Pop_2$ , and  $Pop_3$  according to equation (8); let  $\sigma_1 + \sigma_2 + \sigma_3 = 1$ ;
Begin:
  Evolutionary generation  $G = 0$ , current evaluation times  $FES = 0$ ;
  While  $FES \leq \text{MaxFES}$ 
    For  $j = 1 \rightarrow 3$ 
      If  $G > 0$ 
        Calculate  $Fm_j$ ,  $CRm_j$  according to equations (13), (14), (16), and (17);
      End if
      For  $i = 1 \rightarrow NP_j$ 
        Calculate the subpopulation  $F_{i,j}$ ,  $CR_{i,j}$  according to equations (12) and (15);
        Perform mutation strategies according to equations (5), (6), and (7), respectively;
        Perform covariance learning according to equations (10) and (11);
      End for
      For  $i = 1 \rightarrow NP_j$ 
        If  $f(u_i^G) \leq f(x_i^G)$ 
           $x_i^{G+1} = u_i^G$ ;
           $S_{F,j} = F_{i,j}$ ,  $SCR_j = CR_{i,j}$ ;
        Else
           $x_i^{G+1} = x_i^G$ ;
        End if
         $FES = FES + NP_j$ ;
      End for
    End for
     $Pop = \sum_{j=1,2,3} Pop_j$ ,  $G = G + 1$ ;
    According to the equation (9), the proportion of good individuals and the redistribution of subpopulations were counted;
  End while
End
Output: the population of individual minimum objective function.

```

ALGORITHM 1: Pseudocode of the MCDE algorithm.

in the accuracy of the benchmark functions F_3 (a), F_4 (b), and F_5 (c). (2) Basic multimodal function F_6 – F_{12} : the best performing algorithm is CoBiDE, which is better than MCDE on benchmark functions F_6 , F_8 , F_9 , and F_{11} . Test results on F_7 (d), F_{10} (e), and F_{12} (f) are worse than MCDE.

Overall, the MCDE is similar to the CoBiDE on this type of benchmark function. (3) Extended multimodal function F_{13} – F_{14} (g): the average error of the seven algorithms is at an order of magnitude, but the effects of JADE, CoDE, and MCDE are slightly better than the other four algorithms. (4)

TABLE 1: Comparison results of MCDE with JADE, jDE, SaDE, EPSDE, CoDE, and CoBiDE ($D = 30$).

Function	JADE	jDE	SaDE	EPSDE	CoDE	CoBiDE	LSHADE	MCDE
F_1	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$
F_2	$1.49E-28 \pm 1.32E-28+$	$1.06E-06 \pm 1.92E-06-$	$1.21E-05 \pm 1.39E-05-$	$4.25E-10 \pm 2.12E-09-$	$2.39E-15 \pm 5.11E-15-$	$1.52E-12 \pm 3.08E-12-$	$3.08E-29 \pm 6.89E-29+$	$8.52E-28 \pm 3.79E-28$
F_3	$8.36E+03 \pm 6.57E+03-$	$1.67E+05 \pm 1.06E+05-$	$3.71E+05 \pm 1.79E+05-$	$1.37E+06 \pm 4.97E+06-$	$1.27E+05 \pm 7.91E+04-$	$8.86E+04 \pm 6.74E+04-$	$7.70E+03 \pm 6.16E+03-$	$3.82E-12 \pm 1.64E-11$
F_4	$1.88E-15 \pm 8.53E-15-$	$2.06E-02 \pm 3.14E-02-$	$1.14E+01 \pm 1.39E+01-$	$2.30E+05 \pm 6.26E-05-$	$5.24E-03 \pm 1.12E-02-$	$1.35E-03 \pm 2.34E-03-$	$5.05E-15 \pm 2.44E-14-$	$7.12E-22 \pm 3.13E-21$
F_5	$3.86E-08 \pm 1.41E-07-$	$3.27E+03 \pm 2.94E+02-$	$2.19E+03 \pm 7.10E+02-$	$9.15E+02 \pm 5.45E+02-$	$5.52E+02 \pm 4.20E+02-$	$1.04E+02 \pm 8.07E+01-$	$7.09E-10 \pm 3.21E-09-$	$4.54E-10 \pm 6.03E-10$
F_6	$1.73E+01 \pm 5.14E+01-$	$1.96E+01 \pm 2.22E+01-$	$5.61E+01 \pm 2.78E+01-$	$3.18E-01 \pm 1.10E+00-$	$3.18E-01 \pm 1.10E+00-$	$3.73E-02 \pm 4.66E-02+$	$1.59E-01 \pm 7.97E-01+$	$3.18E-01 \pm 1.10E+00$
F_7	$8.17E-03 \pm 8.57E-03-$	$1.18E-02 \pm 9.30E-03-$	$1.19E-02 \pm 5.83E-03-$	$1.22E-02 \pm 1.34E-02-$	$6.99E-03 \pm 7.47E-03-$	$3.25E-03 \pm 5.49E-03-$	$7.88E-03 \pm 2.82E-03-$	$1.67E-03 \pm 4.06E-03$
F_8	$2.08E+01 \pm 2.43E-01 \approx$	$2.09E+01 \pm 3.91E-02 \approx$	$2.09E+01 \pm 5.11E-02 \approx$	$2.09E+01 \pm 6.24E-02 \approx$	$2.02E+01 \pm 1.41E-01+$	$2.07E+01 \pm 3.84E-01+$	$2.04E+01 \pm 3.47E-01+$	$2.09E+01 \pm 4.25E-02$
F_9	$0.00E+00 \pm 0.00E+00+$	$0.00E+00 \pm 0.00E+00+$	$0.00E+00 \pm 0.00E+00+$	$0.00E+00 \pm 0.00E+00+$	$0.00E+00 \pm 0.00E+00+$	$0.00E+00 \pm 0.00E+00+$	$0.00E+00 \pm 0.00E+00+$	$2.56E-07 \pm 5.94E-07$
F_{10}	$2.37E+01 \pm 3.91E+00-$	$5.45E+01 \pm 9.77E+00-$	$3.86E+01 \pm 8.61E+01-$	$6.14E+01 \pm 9.02E+00-$	$4.23E+01 \pm 1.12E+01-$	$4.28E+01 \pm 1.44E+01-$	$1.53E+01 \pm 2.74E+00+$	$2.23E+01 \pm 6.39E+00$
F_{11}	$2.55E+01 \pm 1.62E+00-$	$2.82E+01 \pm 1.71E+00-$	$1.97E+01 \pm 7.77E+00-$	$3.57E+01 \pm 2.87E+00-$	$1.31E+01 \pm 3.10E+00+$	$6.05E+00 \pm 2.24E+00+$	$2.68E+01 \pm 1.69E+00-$	$1.45E+01 \pm 7.73E+00$
F_{12}	$5.88E+03 \pm 3.63E+03-$	$1.17E+04 \pm 9.44E+03-$	$5.15E+03 \pm 4.33E+03-$	$5.75E+04 \pm 1.03E+04-$	$2.81E+03 \pm 3.19E+03-$	$3.86E+03 \pm 5.02E+03-$	$2.44E+03 \pm 2.44E+03-$	$1.71E+03 \pm 2.33E+03$
F_{13}	$1.45E+00 \pm 1.16E-01+$	$1.71E+00 \pm 1.37E-01+$	$4.55E+00 \pm 3.10E-01-$	$2.52E+00 \pm 1.77E-01-$	$1.61E+00 \pm 4.13E-01+$	$2.69E+00 \pm 9.92E-01-$	$1.17E+00 \pm 1.11E-01+$	$1.78E+00 \pm 2.04E-01$
F_{14}	$1.23E+01 \pm 3.21E-01 \approx$	$1.29E+01 \pm 2.88E-01 \approx$	$1.28E+01 \pm 2.24E-01-$	$1.34E+01 \pm 1.98E-01-$	$1.23E+01 \pm 5.90E-01 \approx$	$1.22E+01 \pm 5.23E-01 \approx$	$1.23E+01 \pm 4.08E-01 \approx$	$1.23E+01 \pm 2.53E-01$
F_{15}	$3.26E+02 \pm 1.27E+02+$	$3.56E+02 \pm 6.46E+01+$	$3.84E+02 \pm 7.48E+01+$	$2.36E+02 \pm 4.25E+01+$	$4.00E+02 \pm 5.00E-01 \approx$	$4.12E+02 \pm 6.00E+01-$	$3.56E+02 \pm 7.11E+01+$	$4.00E+02 \pm 1.15E+02$
F_{16}	$9.14E+01 \pm 9.89E+01-$	$8.04E+01 \pm 2.51E+01-$	$5.99E+01 \pm 2.16E+01-$	$1.18E+02 \pm 8.78E+01-$	$6.89E+01 \pm 2.23E+01+$	$6.50E+01 \pm 1.26E+01-$	$9.07E+01 \pm 1.32E+02-$	$5.48E+01 \pm 2.92E+01$
F_{17}	$1.44E+02 \pm 1.34E+02-$	$1.40E+02 \pm 3.24E+01-$	$8.38E+01 \pm 7.68E+01-$	$1.74E+02 \pm 7.24E+01-$	$7.24E+01 \pm 2.54E+01-$	$8.03E+01 \pm 7.42E+01-$	$1.08E+02 \pm 1.11E+02-$	$6.41E+01 \pm 7.58E+01$
F_{18}	$9.04E+02 \pm 7.97E-01 \approx$	$9.04E+02 \pm 9.87E-01 \approx$	$8.69E+02 \pm 5.78E+01+$	$8.16E+02 \pm 5.16E-01+$	$9.04E+02 \pm 1.31E+00 \approx$	$9.04E+02 \pm 7.16E-01 \approx$	$9.03E+02 \pm 6.22E-01 \approx$	$9.03E+02 \pm 5.59E-01$
F_{19}	$9.04E+02 \pm 1.02E+00 \approx$	$9.04E+02 \pm 9.35E-01 \approx$	$8.51E+02 \pm 5.76E+01+$	$8.16E+02 \pm 5.59E-01+$	$9.04E+02 \pm 8.03E-01 \approx$	$9.04E+02 \pm 8.61E-01 \approx$	$9.03E+02 \pm 7.09E-01 \approx$	$9.03E+02 \pm 2.01E-01$
F_{20}	$9.04E+02 \pm 9.61E-01 \approx$	$9.04E+02 \pm 1.04E+00 \approx$	$8.51E+02 \pm 5.77E+02+$	$8.16E+02 \pm 4.03E-01+$	$9.04E+02 \pm 7.11E-01 \approx$	$9.04E+02 \pm 7.38E-01 \approx$	$9.03E+02 \pm 7.21E-01 \approx$	$9.03E+02 \pm 2.40E-01$
F_{21}	$5.00E+02 \pm 7.33E-14 \approx$	$5.00E+02 \pm 8.83E-14 \approx$	$5.00E+02 \pm 9.84E-14 \approx$	$8.58E+02 \pm 1.37E+02-$	$5.00E+02 \pm 2.84E-14 \approx$	$5.00E+02 \pm 7.36E-14 \approx$	$5.00E+02 \pm 1.25E-13 \approx$	$5.00E+02 \pm 6.76E-14$
F_{22}	$8.63E+02 \pm 2.06E+01 \approx$	$8.81E+02 \pm 1.46E+01 \approx$	$9.16E+02 \pm 1.54E+01-$	$5.04E+02 \pm 3.63E+00+$	$8.62E+02 \pm 2.22E+01 \approx$	$8.69E+02 \pm 3.42E+01 \approx$	$8.75E+02 \pm 1.69E+01-$	$8.69E+02 \pm 1.92E+01$
F_{23}	$5.66E+02 \pm 1.12E+02-$	$5.34E+02 \pm 2.14E-04 \approx$	$5.34E+02 \pm 2.74E-03 \approx$	$8.66E+02 \pm 1.45E+00-$	$5.34E+02 \pm 4.13E-04 \approx$	$5.34E+02 \pm 2.43E-04 \approx$	$5.34E+02 \pm 2.28E-13 \approx$	$5.34E+02 \pm 3.40E-13$
F_{24}	$2.00E+02 \pm 2.90E-14 \approx$	$2.00E+02 \pm 2.90E-14 \approx$	$2.00E+02 \pm 2.90E-14 \approx$	$2.11E+02 \pm 1.16E+00-$	$2.00E+02 \pm 2.90E-14 \approx$	$2.00E+02 \pm 2.90E-14 \approx$	$2.29E+02 \pm 1.48E+02-$	$2.00E+02 \pm 2.90E-14$
F_{25}	$2.11E+02 \pm 8.77E-01-$	$2.11E+02 \pm 7.33E+02-$	$2.11E+02 \pm 1.23E+00-$	$2.11E+02 \pm 2.52E+00-$	$2.10E+02 \pm 7.22E-01-$	$2.10E+02 \pm 8.03E-01-$	$2.11E+02 \pm 7.91E-01-$	$2.09E+02 \pm 2.84E-01$
$-/+/\approx$	13/3/9	13/3/9	15/5/5	18/6/1	10/5/10	13/4/8	11/7/7	

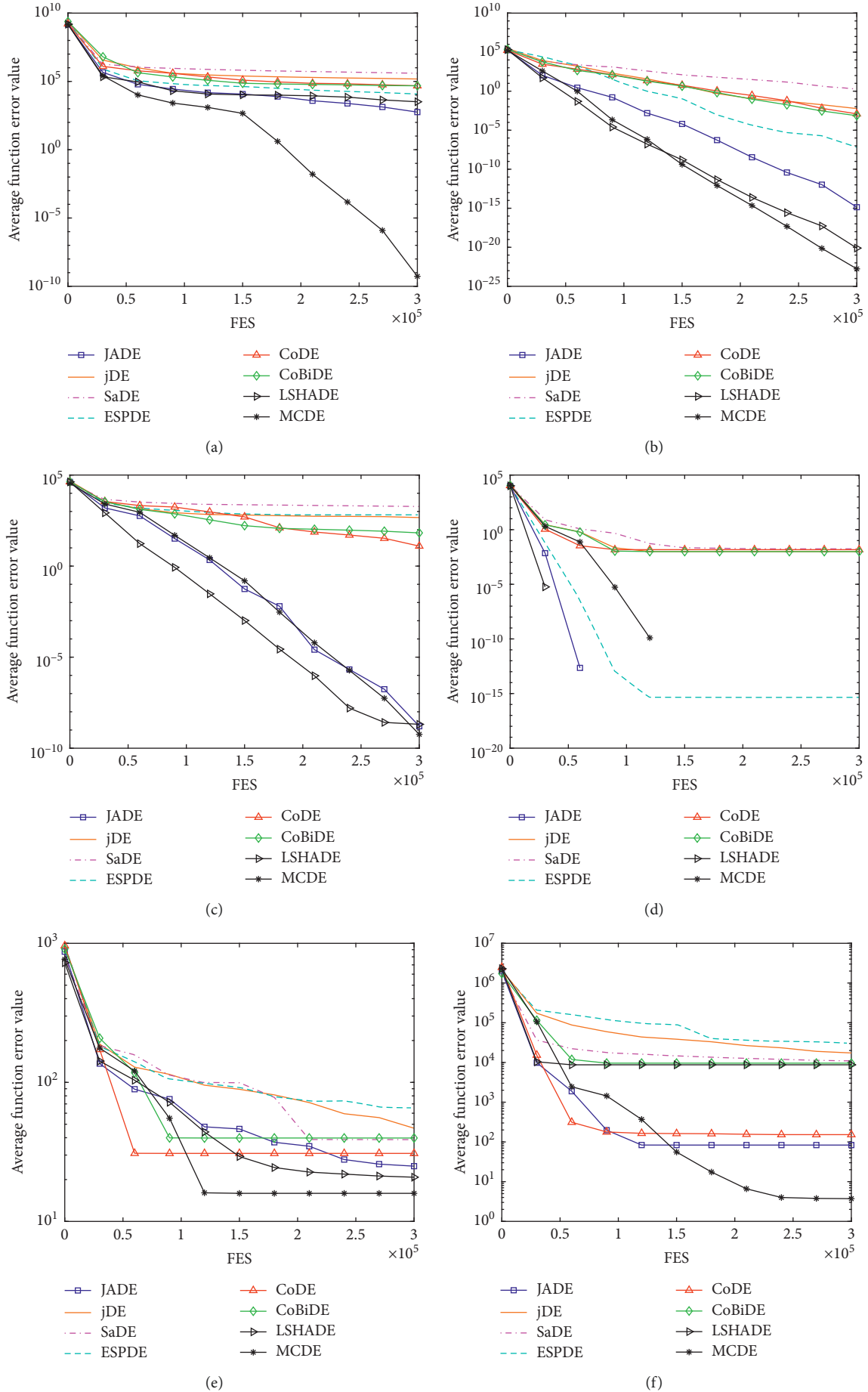


FIGURE 2: Continued.

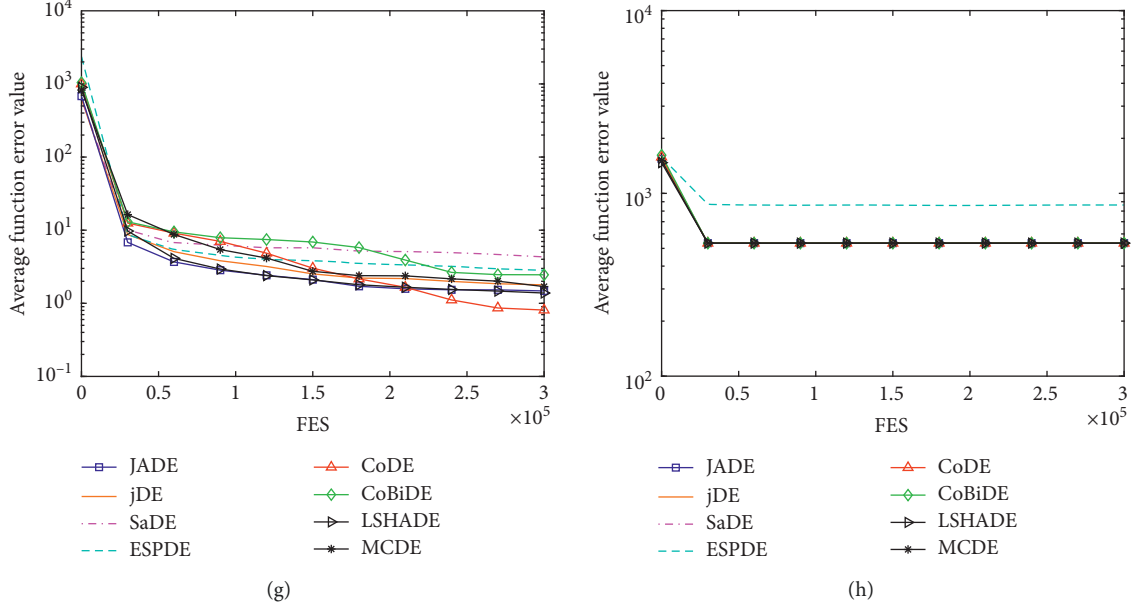


FIGURE 2: Evolution of the mean function error values derived from JADE, jDE, SaDE, CoDE, CoBiDE, LSHADE, and MCDE versus the number of FES on $F_3, F_4, F_5, F_7, F_{10}, F_{12}, F_{13}$, and F_{23} . (a) $D = 30 F_3$. (b) $D = 30 F_4$. (c) $D = 30 F_5$. (d) $D = 30 F_7$. (e) $D = 30 F_{10}$. (f) $D = 30 F_{12}$. (g) $D = 30 F_{13}$. (h) $D = 30 F_{23}$.

Complex function F_{15} – F_{25} (h): MCDE is significantly better than JADE, jDE, CoDE, CoBiDE and LSHADE and slightly better than SaDE and EPSDE.

Based on the above four conclusions and Wilcoxon's test, it can be concluded that the MCDE shows significant effects in the four types of benchmark functions. Finally, the experimental results of the MCDE in $D = 30$ are better than those of JADE, jDE, SaDE, EPSDE, CoDE, CoBiDE, and LSHADE on 13, 13, 15, 18, 10, 13, and 11 benchmark functions, respectively, worse than other comparison algorithms on 3, 3, 5, 6, 5, 4, and 7 benchmark functions, and similar to other comparison algorithms on 9, 9, 5, 1, 10, 8, and 7 benchmark functions.

From Table 2, it can be seen that when Wilcoxon's detection is at $\alpha = 0.05$ and $\alpha = 0.1$, MCDE is more effective than JADE, jDE, SaDE, EPSDE, CoDE, CoBiDE, and LSHADE. According to Friedman's average ranking ($D = 30$) in Table 3, MCDE performed well in all types of benchmark functions and achieved the best ranking.

The experimental results of the above seven algorithms are shown in Table 4, where $D = 50$ and $\text{MaxFES} = 500000$. Based on the data given in Table 4 and Figure 3, we can draw the following conclusions. (1) Unimodal function F_1 – F_5 : in $D = 50$, the effect of MCDE is higher than the other algorithms on F_4 (b) and F_5 (c). It shows that the search scope of mutation strategy is wider and the parameter-adaptive control performance is better. The test results of F_2 and F_3 are only inferior to the JADE and LSHADE. (2) Basic multimodal function F_6 – F_{12} : CoBiDE and LSHADE perform significantly better in this type of benchmark functions, whereas MCDE is slightly better than that on F_6 (d), F_7 (e), F_{10} (f), and F_{12} (g). (3) Extended multimodal function F_{13} – F_{14} (h): the accuracy of MCDE is better than that of the other six algorithms, mainly because the proposed algorithm

TABLE 2: Wilcoxon's detection ($D = 30$).

Algorithm	R^+	R^-	p value	$\alpha = 0.05$	$\alpha = 0.1$
MCDE vs JADE	249.0	76.0	0.017653	Yes	Yes
MCDE vs jDE	262.0	38.0	0.001111	Yes	Yes
MCDE vs SaDE	224.0	76.0	0.032663	Yes	Yes
MCDE vs EPSDE	219.5	80.5	0.027602	Yes	Yes
MCDE vs CoDE	225.5	74.5	0.027602	Yes	Yes
MCDE vs CoBiDE	243.0	57.0	0.006725	Yes	Yes
MCDE vs LSHADE	196.5	103.5	0.19244	No	No

TABLE 3: Friedman's average ranking ($D = 30$).

Algorithm	Ranking	Final rank
JADE	4.48	5
jDE	5.76	7
SaDE	5.08	6
EPSDE	5.68	8
CoDE	3.92	3
CoBiDE	4.22	4
LSHADE	3.62	2
MCDE	3.24	1

adopts a multiscale strategy to search for the best results. (4) Complex function F_{15} – F_{25} (i) (j): such functions are the most complex problems on benchmark functions. There are 11 benchmark functions in all. The result of JADE, jDE, SaDE, CoDE, and LSHADE is not significant. The result of EPSDE is not stable, while MCDE, JADE, and CoBiDE are better than them.

Based on the above four conclusions and Wilcoxon's test, it can be concluded that MCDE shows significant effects in the four types of benchmark functions. Finally, the experimental results of the MCDE in $D = 50$ are better than the

TABLE 4: Comparison results of MCDE with JADE, jDE, SaDE, EPSDE, CoDE, and CoBiDE ($D = 50$).

Function	JADE	jDE	SaDE	EPSDE	CoDE	CoBiDE	LSHADE	MCDE
F_1	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$	$5.25E-29 \pm 1.28E-28$	$8.07E-30 \pm 4.04E-29$	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$
F_2	$3.57E-27 \pm 1.67E-27+$	$2.22E-02 \pm 3.21E-02$	$6.55E-02 \pm 4.34E-02$	$5.77E-23 \pm 8.87E-23+$	$5.87E-09 \pm 6.51E-09$	$1.03E-06 \pm 1.08E-06$	$6.85E-27 \pm 7.92E-27+$	$5.76E-13 \pm 1.01E-12$
F_3	$1.41E+05 \pm 6.61E+03$	$4.48E+05 \pm 2.21E+05$	$8.01E+05 \pm 5.44E+05$	$7.82E+06 \pm 1.35E+07$	$3.22E+05 \pm 8.66E+04$	$2.31E+05 \pm 7.87E+04$	$1.97E+04 \pm 1.07E+04+$	$4.31E+04 \pm 3.02E+04$
F_4	$2.79E+00 \pm 1.26E+01$	$3.88E+02 \pm 3.13E+02$	$7.28E+02 \pm 4.56E+02$	$3.42E+03 \pm 3.51E+04$	$4.98E+02 \pm 5.78E+02$	$1.52E+02 \pm 9.61E+01$	$2.24E+01 \pm 5.73E+01$	$8.51E-01 \pm 8.82E-01$
F_5	$1.45E+03 \pm 4.38E+02$	$3.65E+03 \pm 6.29E+02$	$8.34E+03 \pm 1.29E+03$	$4.66E+03 \pm 8.85E+02$	$3.60E+03 \pm 5.69E+02$	$2.71E+03 \pm 6.16E+02$	$1.12E+03 \pm 4.72E+02$	$2.09E+02 \pm 4.11E+02$
F_6	$7.71E+00 \pm 3.03E+01$	$4.43E+02 \pm 2.78E+01$	$4.39E+02 \pm 2.52E+01$	$1.43E+02 \pm 1.95E+00$	$1.23E+02 \pm 2.12E+00$	$2.36E+01 \pm 1.88E+01$	$7.37E-01 \pm 1.49E+00$	$7.21E-01 \pm 1.46E+00$
F_7	$7.57E-03 \pm 1.09E-02$	$2.85E-03 \pm 6.10E-03+$	$9.23E-03 \pm 5.45E-02$	$1.08E-02 \pm 1.79E-02$	$6.54E-03 \pm 9.36E-03$	$4.01E-03 \pm 3.09E-03$	$2.16E-03 \pm 4.04E-03+$	$3.05E-03 \pm 6.32E-03$
F_8	$2.11E+01 \pm 5.93E-02$	$2.11E+01 \pm 3.72E-02$	$2.11E+01 \pm 4.33E-02$	$2.11E+01 \pm 3.35E-02$	$2.01E+01 \pm 1.09E-01+$	$2.08E+01 \pm 4.45E-01+$	$2.07E+01 \pm 2.89E-01+$	$2.11E+01 \pm 5.06E-02$
F_9	$0.00E+00 \pm 0.00E+00+$	$0.00E+00 \pm 0.00E+00+$	$9.94E-01 \pm 2.21E-01$	$6.39E-16 \pm 1.01E-15+$	$2.38E+00 \pm 4.33E-01$	$3.21E-13 \pm 9.03E-13+$	$7.11E-17 \pm 3.55E-16+$	$9.55E-01 \pm 2.92E+00$
F_{10}	$6.55E+01 \pm 6.93E+00$	$9.98E+01 \pm 1.32E+01$	$1.14E+02 \pm 1.54E+01$	$1.54E+02 \pm 2.53E+01$	$8.29E+01 \pm 1.93E+01$	$9.24E+01 \pm 2.04E+01$	$4.60E+01 \pm 5.30E+00$	$4.31E+01 \pm 1.15E+01$
F_{11}	$5.24E+01 \pm 2.15E+00$	$5.42E+01 \pm 2.10E+00$	$4.45E+01 \pm 1.89E+00$	$7.04E+01 \pm 3.21E+00$	$3.14E+01 \pm 5.35E+00$	$1.80E+01 \pm 4.49E+00+$	$5.39E+01 \pm 3.26E+00$	$3.03E+01 \pm 8.38E+00$
F_{12}	$1.54E+04 \pm 1.27E+04$	$1.57E+04 \pm 1.54E+04$	$5.65E+04 \pm 2.04E+04$	$3.16E+05 \pm 3.85E+04$	$1.54E+04 \pm 1.73E+04$	$1.35E+04 \pm 1.22E+04$	$1.21E+04 \pm 1.04E+04$	$1.16E+04 \pm 1.01E+04$
F_{13}	$2.78E+00 \pm 1.99E-01$	$2.94E+00 \pm 2.41E-01$	$7.27E+00 \pm 7.34E-01$	$6.17E+00 \pm 6.03E-01$	$3.23E+00 \pm 4.15E-01$	$4.28E+00 \pm 1.32E+00$	$2.14E+00 \pm 1.03E-01+$	$2.72E+00 \pm 4.54E-01$
F_{14}	$2.16E+01 \pm 4.76E-01$	$2.25E+01 \pm 2.26E-01$	$2.23E+01 \pm 2.42E-01$	$2.34E+01 \pm 2.63E-01$	$2.19E+01 \pm 4.39E-01$	$2.20E+01 \pm 3.87E-01$	$2.18E+01 \pm 3.12E-01$	$2.12E+01 \pm 3.24E-01$
F_{15}	$3.77E+02 \pm 9.03E+01$	$3.62E+02 \pm 1.21E+02+$	$3.86E+02 \pm 7.62E+01$	$2.64E+02 \pm 6.45E+01+$	$3.88E+02 \pm 6.00E+01$	$3.92E+02 \pm 4.00E+01$	$2.97E+02 \pm 9.67E+01+$	$3.68E+02 \pm 6.23E+01$
F_{16}	$8.42E+01 \pm 7.92E+01$	$8.35E+01 \pm 1.03E+01$	$8.78E+01 \pm 6.57E+01$	$1.50E+02 \pm 4.25E+01$	$9.35E+01 \pm 7.01E+01$	$7.64E+01 \pm 2.11E+01$	$9.82E+02 \pm 1.35E+02$	$7.56E+01 \pm 1.01E+02$
F_{17}	$9.39E+01 \pm 2.64E+01$	$1.81E+02 \pm 2.31E+01$	$9.81E+01 \pm 1.01E+02$	$2.38E+02 \pm 7.01E+01$	$7.21E+01 \pm 2.58E+01+$	$7.62E+01 \pm 1.73E+01+$	$1.18E+02 \pm 1.08E+02$	$8.84E+01 \pm 1.26E+02$
F_{18}	$9.21E+02 \pm 4.38E+00$	$9.20E+02 \pm 3.35E+00$	$9.78E+02 \pm 8.37E+01$	$8.53E+02 \pm 2.42E+01+$	$9.21E+02 \pm 5.42E+00$	$9.12E+02 \pm 2.39E+01$	$9.18E+02 \pm 4.25E+00$	$9.14E+02 \pm 9.46E+00$
F_{19}	$9.19E+02 \pm 1.07E+01$	$9.20E+02 \pm 2.88E+00$	$9.78E+02 \pm 7.81E+01$	$8.59E+02 \pm 1.54E+01$	$9.21E+02 \pm 4.64E+00$	$9.14E+02 \pm 2.39E+01$	$9.18E+02 \pm 6.62E+00$	$9.15E+02 \pm 2.90E+00$
F_{20}	$9.21E+02 \pm 3.38E+00$	$9.20E+02 \pm 3.14E+00$	$9.55E+02 \pm 3.54E+01$	$8.56E+02 \pm 3.03E+00+$	$9.11E+02 \pm 3.38E+01$	$9.18E+02 \pm 3.40E+00$	$9.17E+02 \pm 7.36E+00$	$9.16E+02 \pm 3.94E+00$
F_{21}	$5.52E+02 \pm 1.49E+02+$	$7.21E+02 \pm 2.55E+02$	$5.66E+02 \pm 2.32E+02+$	$7.29E+02 \pm 2.82E+00$	$6.83E+02 \pm 2.49E+02$	$5.40E+02 \pm 1.41E+02+$	$6.82E+02 \pm 2.48E+02$	$6.21E+02 \pm 2.21E+02$
F_{22}	$9.05E+02 \pm 2.48E+01$	$9.05E+02 \pm 1.23E+00$	$9.82E+02 \pm 8.21E+01$	$5.00E+02 \pm 6.61E-02+$	$9.01E+02 \pm 2.18E+01$	$8.84E+02 \pm 2.53E+01+$	$8.97E+02 \pm 1.94E+01$	$8.97E+02 \pm 1.65E+01$
F_{23}	$5.82E+02 \pm 1.31E+02+$	$8.60E+02 \pm 2.25E+02$	$5.98E+02 \pm 7.29E+00+$	$7.33E+02 \pm 4.48E+00$	$7.10E+02 \pm 2.33E+02$	$6.91E+02 \pm 2.25E+02$	$8.04E+02 \pm 2.40E+02$	$6.90E+02 \pm 2.25E+02$
F_{24}	$2.00E+02 \pm 1.49E-12$	$2.00E+02 \pm 1.62E-12$	$2.89E+02 \pm 6.55E+01$	$2.38E+02 \pm 1.34E+01$	$2.00E+02 \pm 5.81E-14$	$2.00E+02 \pm 1.08E-12$	$2.32E+02 \pm 1.57E+02$	$2.00E+02 \pm 1.66E-12$
F_{25}	$2.18E+02 \pm 1.71E+00$	$2.16E+02 \pm 1.41E+00$	$2.24E+02 \pm 1.25E+01$	$2.47E+02 \pm 1.87E+01$	$2.17E+02 \pm 1.97E+00$	$2.15E+02 \pm 1.69E+00$	$2.18E+02 \pm 1.69E+00$	$2.14E+02 \pm 1.76E-01$
$-/+/\approx$	18/4/3	20/2/3	21/2/2	17/7/1	22/2/1	13/6/6	15/7/3	

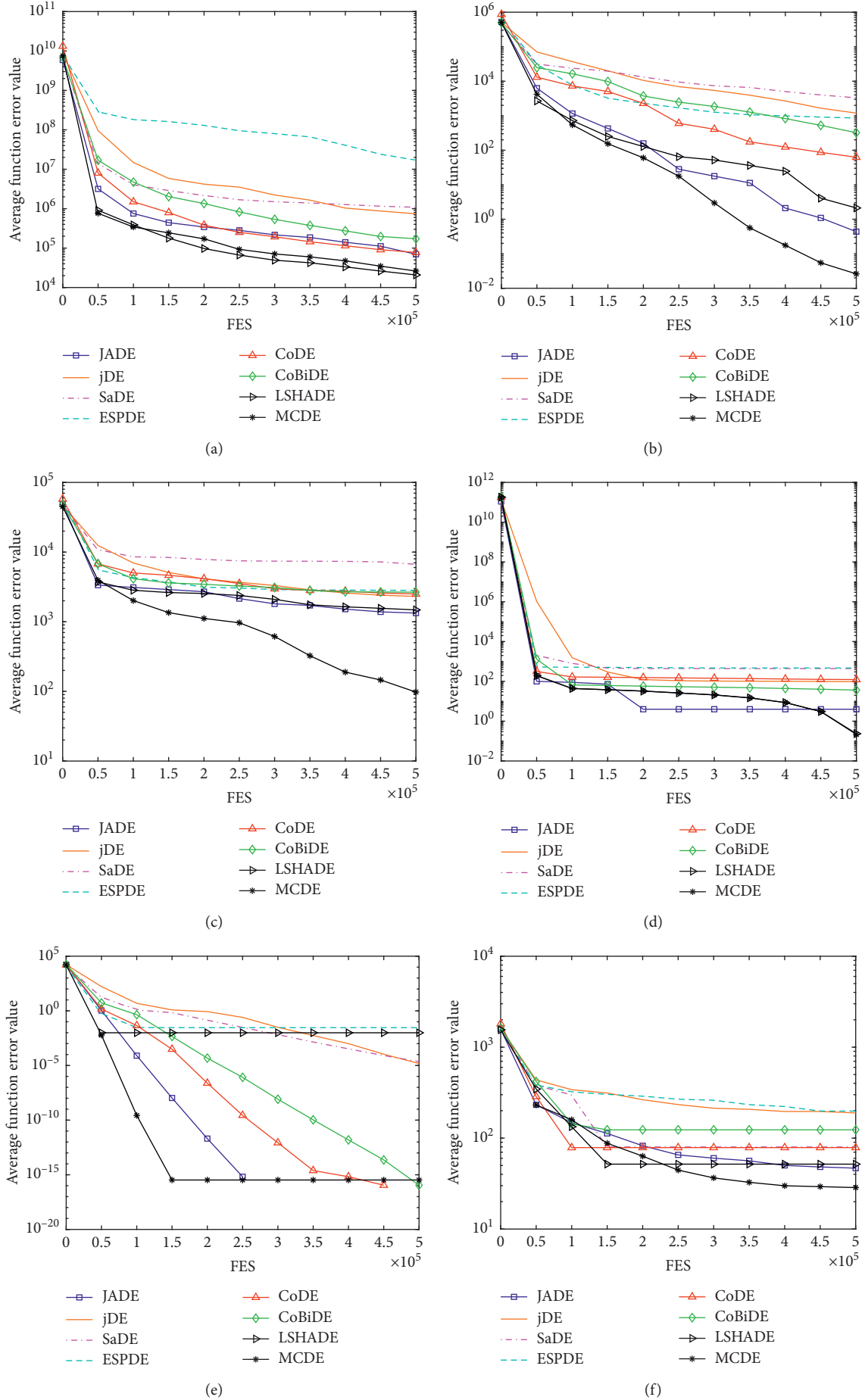


FIGURE 3: Continued.

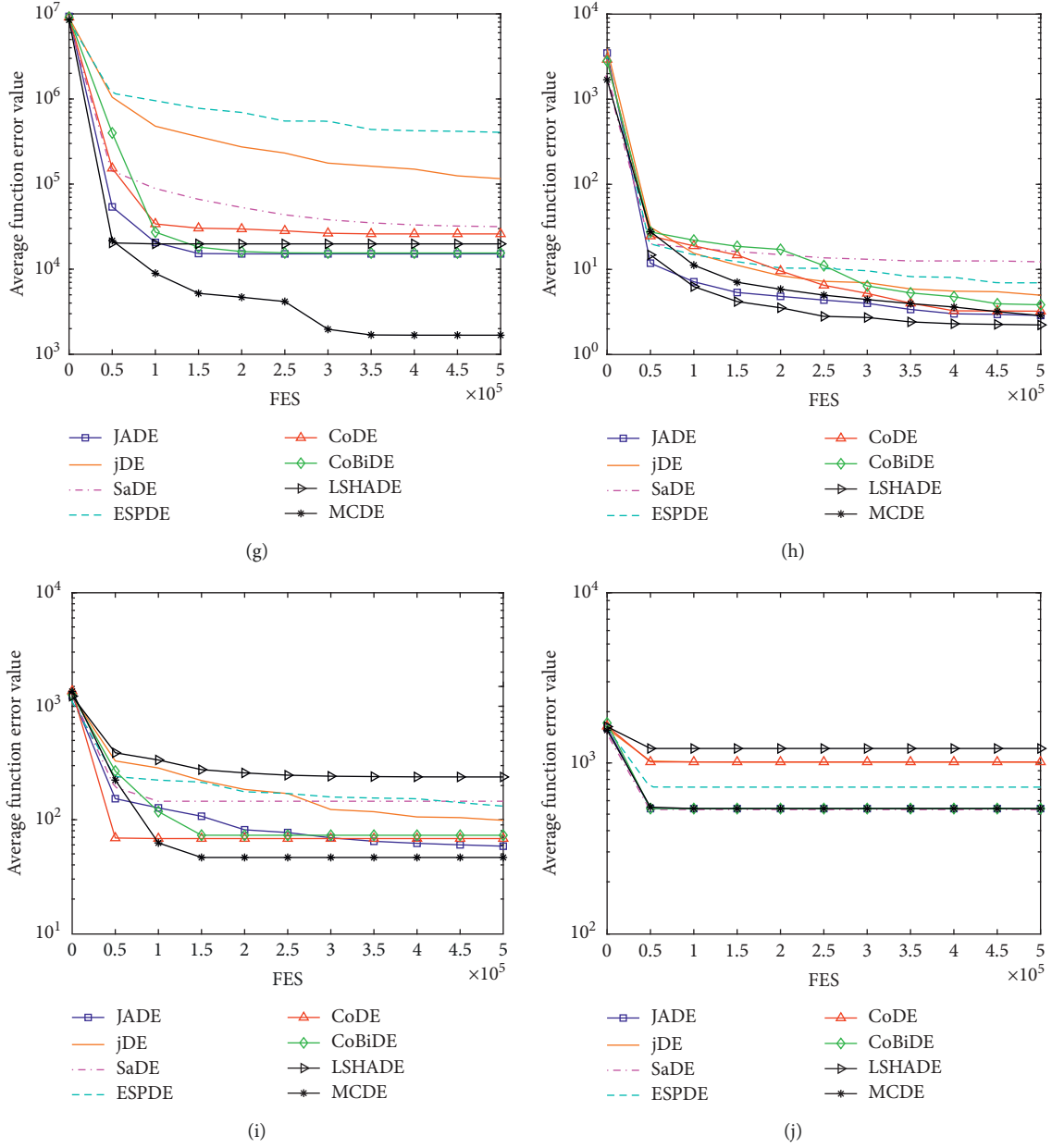


FIGURE 3: Evolution of the mean function error values derived from JADE, jDE, SaDE, CoDE, CoBiDE, LSHADE, and MCDE versus the number of FES on $F_3, F_4, F_5, F_6, F_7, F_{10}, F_{12}, F_{13}, F_{16},$ and F_{23} . (a) $D = 50 F_3$. (b) $D = 50 F_4$. (c) $D = 50 F_5$. (d) $D = 50 F_6$. (e) $D = 50 F_7$. (f) $D = 50 F_{10}$. (g) $D = 50 F_{12}$. (h) $D = 50 F_{13}$. (i) $D = 50 F_{16}$. (j) $D = 50 F_{23}$.

TABLE 5: Wilcoxon's detection ($D = 50$).

Algorithm	R^+	R^-	p value	$\alpha = 0.05$	$\alpha = 0.1$
MCDE vs JADE	267.5	32.5	0.000716	Yes	Yes
MCDE vs jDE	278.0	22.0	0.000229	Yes	Yes
MCDE vs SaDE	303.5	21.5	0.00014	Yes	Yes
MCDE vs EPSDE	223.0	77	0.035729	Yes	Yes
MCDE vs CoDE	269.0	31.0	0.000611	Yes	Yes
MCDE vs CoBiDE	214.0	86.0	0.060605	No	Yes
MCDE vs LSHADE	250.5	74.5	0.016642	Yes	Yes

TABLE 6: Friedman's average ranking ($D = 50$).

Algorithm	Ranking	Final rank
JADE	3.98	4
jDE	5.36	6
SaDE	6.46	8
EPSDE	5.64	7
CoDE	4.88	5
CoBiDE	3.54	2
LSHADE	3.56	3
MCDE	2.58	1

TABLE 7: Comparison results of MCDE with CLPSO, CMA-ES, and GL-25 ($D = 30$).

Function	CLPSO	CMA-ES	GL-25	MCDE	
Unimodal function	F_1	$0.00E+00 \pm 0.00E+00 \approx$	$1.58E-25 \pm 3.35E-26-$	$5.60E-27 \pm 1.76E-26-$	$0.00E+00 \pm 0.00E+00$
	F_2	$8.40E+02 \pm 1.90E+02-$	$1.12E-24 \pm 2.93E-25-$	$4.04E+01 \pm 6.28E+01-$	$8.52E-28 \pm 3.79E-28$
	F_3	$1.42E+07 \pm 4.19E+06-$	$5.54E-21 \pm 1.69E-21+$	$2.19E+06 \pm 1.08E+06-$	$3.82E-12 \pm 1.64E-11$
	F_4	$6.99E+03 \pm 1.73E+03-$	$9.15E+05 \pm 2.16E+06-$	$9.07E+02 \pm 4.25E+02-$	$7.12E-22 \pm 3.13E-21$
	F_5	$3.86E+03 \pm 4.35E+02-$	$2.77E-10 \pm 5.04E-11+$	$2.51E+03 \pm 1.96E+02-$	$4.54E-10 \pm 6.03E-10$
Basic multimodal function	F_6	$4.16E+00 \pm 3.48E+00-$	$4.78E-01 \pm 1.32E+00-$	$2.15E+01 \pm 1.17E+00-$	$3.18E-01 \pm 1.10E+00$
	F_7	$4.51E-01 \pm 8.47E-02-$	$1.82E-03 \pm 4.33E-03-$	$2.78E-02 \pm 3.62E-02-$	$1.67E-03 \pm 4.06E-03$
	F_8	$2.09E+01 \pm 4.41E-02-$	$2.03E+01 \pm 5.72E-01+$	$2.09E+01 \pm 5.94E-02-$	$2.09E+01 \pm 4.25E-02$
	F_9	$0.00E+00 \pm 0.00E+00+$	$4.45E+02 \pm 7.12E+01-$	$2.45E+01 \pm 7.35E+00-$	$2.56E-07 \pm 5.94E-07$
	F_{10}	$1.04E+02 \pm 1.53E+01-$	$4.63E+01 \pm 1.16E+01-$	$1.42E+02 \pm 6.45E+01-$	$2.23E+01 \pm 6.39E+00$
	F_{11}	$2.60E+01 \pm 1.63E+00-$	$7.11E+00 \pm 2.14E+00+$	$3.27E+01 \pm 7.79E+00-$	$1.45E+01 \pm 7.73E+00$
	F_{12}	$1.79E+04 \pm 5.24E+03-$	$1.26E+04 \pm 1.74E+04-$	$6.53E+04 \pm 4.69E+04-$	$1.71E+03 \pm 2.33E+03$
Expanded multimodal function	F_{13}	$2.06E+00 \pm 2.15E-01-$	$3.43E+00 \pm 7.60E-01-$	$6.23E+00 \pm 4.88E+00-$	$1.78E+00 \pm 2.04E-01$
	F_{14}	$1.28E+01 \pm 2.48E-01-$	$1.47E+01 \pm 3.31E-01-$	$1.31E+01 \pm 1.84E-01-$	$1.23E+01 \pm 2.53E-01$
Hybrid composition function	F_{15}	$5.77E+01 \pm 2.76E+01-$	$5.55E+02 \pm 3.32E+02-$	$3.04E+02 \pm 1.99E+01+$	$4.00E+02 \pm 1.15E+02$
	F_{16}	$1.74E+02 \pm 2.82E+01-$	$2.98E+02 \pm 2.08E+02-$	$1.32E+02 \pm 7.60E+01-$	$5.48E+01 \pm 2.92E+01$
	F_{17}	$2.46E+02 \pm 4.81E+01-$	$4.43E+02 \pm 3.34E+02-$	$1.61E+02 \pm 6.80E+01-$	$6.41E+01 \pm 7.58E+01$
	F_{18}	$9.13E+02 \pm 1.42E+00-$	$9.04E+02 \pm 3.01E-01 \approx$	$9.07E+02 \pm 1.48E+00-$	$9.03E+02 \pm 5.59E-01$
	F_{19}	$9.14E+02 \pm 1.45E+00-$	$9.16E+02 \pm 6.03E+01-$	$9.06E+02 \pm 1.24E+00-$	$9.03E+02 \pm 2.01E-01$
	F_{20}	$9.14E+02 \pm 3.62E+00-$	$9.04E+02 \pm 2.71E-01 \approx$	$9.07E+02 \pm 1.35E+00-$	$9.03E+02 \pm 2.40E-01$
	F_{21}	$5.00E+02 \pm 3.39E-13 \approx$	$5.00E+02 \pm 2.68E-12 \approx$	$5.00E+02 \pm 4.83E-13 \approx$	$5.00E+02 \pm 6.76E-14$
	F_{22}	$9.72E+02 \pm 1.20E+01-$	$8.26E+02 \pm 1.46E+01+$	$9.28E+02 \pm 7.04E+01-$	$8.69E+02 \pm 1.92E+01$
	F_{23}	$5.34E+02 \pm 2.19E-04 \approx$	$5.36E+02 \pm 5.44E+00 \approx$	$5.34E+02 \pm 4.66E-04 \approx$	$5.34E+02 \pm 3.40E-13$
	F_{24}	$2.00E+02 \pm 1.49E-12 \approx$	$2.12E+02 \pm 6.00E+01-$	$2.00E+02 \pm 5.52E-11 \approx$	$2.00E+02 \pm 2.90E-14$
	F_{25}	$2.00E+02 \pm 1.96E+00+$	$2.07E+02 \pm 6.07E+00 \approx$	$2.17E+02 \pm 1.36E-01-$	$2.09E+02 \pm 2.84E-01$
	$-/+/\approx$				
	21/1/3				

JADE, jDE, SaDE, EPSDE, CoDE, CoBiDE, and LSHADE on 18, 20, 21, 17, 22, 13, and 15 benchmark functions, worse than them on 4, 2, 2, 7, 2, 6, and 7 benchmark functions, and similar to them on 3, 3, 2, 1, 1, 6, and 3 benchmark functions, respectively.

From Table 5, it can be seen that when Wilcoxon's detection is at $\alpha = 0.05$, MCDE is more effective than JADE, jDE, SaDE, EPSDE, CoDE, and LSHADE, and the p value of CoBiDE is 0.06. At $\alpha = 0.1$, MCDE has significant differences from other algorithms. Based on Table 6 Friedman average ranking in $D = 50$, MCDE performed well in all types of benchmark functions and achieved the top ranking.

4.2. Comparison with Related Evolutionary Algorithms. To further evaluate MCDE, it is compared with CLPSO [47], CMA-ES [48], and GL-25 [25, 26]. CLPSO is a local version of the PSO, adopting a new learning strategy mechanism. CMA-ES adopts a covariance matrix adaptive mechanism and is mainly utilized to solve continuous optimization problems. GL-25 is a global and local real-coded genetic algorithm based on a new crossover operator. The experimental results of MCDE, CLPSO, CMA-ES, and GL-25 are shown in Table 7 at $D = 30$ and $\text{MaxFES} = 300000$. It can be concluded that MCDE has the most prominent effect on the unimodal function (F_2 – F_5) and is smaller than the average error of other evolution algorithms. On the basic multimodal function F_{10} and F_{12} , the result of MCDE is significantly better than other algorithms. On the extended multimodal functions and complex functions, most functions (F_{14} , F_{16} , F_{17} , F_{21} , F_{23} , F_{24} , and F_{25}) have significant effects. Finally, the experimental results of MCDE in $D = 30$ are better than CLPSO, CMA-ES, and GL on 19, 15, and 21 benchmark functions, worse than those on 2, 5, and 1 benchmark functions, and similar to those on 4, 5, and 3 benchmark functions.

In this paper, the proposed algorithm is further compared with other evolution algorithms. From Table 8, it can be seen that when Wilcoxon's test detects $\alpha = 0.05$ and $\alpha = 0.1$, MCDE's p value is less than 0.05 and 0.1, and the effect is most significant. According to Table 9, the average ranking of Friedman under $D = 30$ shows that MCDE performs best on benchmark functions.

4.3. Runtime Comparison and Mechanism Comparison.

In general, the running time of evolution algorithm contains the operating time of operator and the time of evaluating the fitness function. JADE, jDE, SaDE, EPSDE, CoDE, CoBiDE, and the proposed algorithms were run 25 times independently on 25 benchmark functions, and the average CPU time consumed was recorded. Set the parameters: $\text{MaxFES} = 300000$ and $D = 30$. To compare the average time, this paper determines the running speed of the algorithm by means of the mean CPU time ratio (AR) between the algorithms. $\text{AR} > 1$ shows that the algorithm runs slower than MCDE, and $\text{AR} < 1$ shows that the algorithm is faster than MCDE.

From the average AR in Table 10, it can be seen that its main range is [0.85, 13.41]. jDE runs at the fastest speed, and EPSDE runs at the slowest speed. The proposed algorithm is

TABLE 8: Wilcoxon's detection with evolution algorithm.

Algorithm	R^+	R^-	p value	$\alpha = 0.05$	$\alpha = 0.1$
MCDE vs CLPSO	261.0	39.0	0.001388	Yes	Yes
MCDE vs CMA-ES	241.0	59.0	0.008699	Yes	Yes
MCDE vs GL	301.0	24.0	0.000175	Yes	Yes

TABLE 9: Friedman average ranking with evolution algorithm.

Algorithm	Ranking	Final rank
CLPSO	2.86	3
CMA-ES	2.6	2
GL	2.92	4
MCDE	1.62	1

TABLE 10: Average CPU time for MCDE, JADE, jDE, SaDE, EPSDE, CoDE, and CoBiDE ($D = 30$).

Function	JADE	jDE	SaDE	EPSDE	CoDE	CoBiDE	MCDE
F_1	2.20	1.77	38.72	73.93	6.27	3.78	2.54
F_2	2.29	2.04	35.61	76.71	6.52	4.18	2.88
F_3	2.53	2.19	35.34	72.37	7.45	4.25	2.90
F_4	2.34	2.14	31.52	73.96	6.66	4.28	3.01
F_5	2.63	2.41	35.49	76.41	7.29	4.19	3.32
F_6	2.15	1.87	33.71	79.66	6.34	3.87	2.79
F_7	2.34	2.24	30.73	74.79	6.75	4.05	2.70
F_8	2.82	2.72	33.82	71.86	8.96	4.74	3.18
F_9	2.37	2.05	34.66	73.91	6.95	4.01	2.85
F_{10}	2.65	2.50	32.28	72.04	7.81	4.30	2.94
F_{11}	28.84	28.14	103.99	98.35	32.21	31.09	29.02
F_{12}	10.30	10.44	65.05	81.98	14.53	12.72	10.76
F_{13}	2.47	2.13	31.88	71.55	6.78	4.49	3.02
F_{14}	3.16	2.88	34.24	72.24	8.75	5.09	3.65
F_{15}	76.01	73.49	202.83	150.91	77.95	85.29	75.28
F_{16}	73.83	72.74	202.17	144.18	76.25	82.40	75.63
F_{17}	73.66	72.98	204.50	143.44	78.80	83.14	77.24
F_{18}	82.29	75.51	211.46	150.07	80.43	86.19	79.29
F_{19}	81.26	75.79	211.67	151.88	81.01	86.29	78.97
F_{20}	84.06	75.81	211.70	150.82	80.46	86.50	79.04
F_{21}	81.41	73.25	211.90	154.60	78.09	85.74	77.24
F_{22}	99.15	91.68	256.54	196.01	96.17	103.81	92.63
F_{23}	79.61	73.79	216.39	171.24	80.19	87.85	75.63
F_{24}	56.19	52.84	150.57	145.81	56.74	64.01	54.80
F_{25}	59.55	56.30	154.85	143.67	59.98	67.57	58.39
Average AR	0.93	0.85	6.99	13.41	1.71	1.27	

ranked third. The proposed algorithm is slower than jDE and JADE because multiscale strategies increase the search range but consume more time in mutation strategies.

By increasing the experiment with and without multi-group mechanism and covariance learning, it can be concluded from Table 11 and Figure 4 that the multiscale mechanism (DE-1) is outstanding in the unimodal function F_4 (a). Covariance learning (DE-2) performs significantly in basic multimodal and complex functions with relatively complex structures in F_{10} (b), F_{16} (c), and F_{17} (d). The population structure is a multipopulation mechanism, and each subpopulation combines the corresponding mutation strategy to ensure the individual diversity in the evolutionary process. Then, the covariance learning establishes a proper

TABLE 11: Comparison results of MCDE with DE-1 and DE-2 ($D=30$).

Function	DE-1	DE-2	MCDE
F_1	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00 \approx$	$0.00E+00 \pm 0.00E+00$
F_2	$6.89E-28 \pm 8.87E-28 \approx$	$2.24E-14 \pm 2.68E-13 -$	$8.52E-28 \pm 3.79E-28$
F_3	$1.98E-12 \pm 7.27E-11 \approx$	$1.23E+05 \pm 8.34E+04 -$	$3.82E-12 \pm 1.64E-11$
F_4	$2.09E-24 \pm 3.79E-24 +$	$1.97E-04 \pm 3.69E-04 -$	$7.12E-22 \pm 3.13E-21$
F_5	$1.26E-09 \pm 1.89E-09 -$	$1.91E+01 \pm 6.15E+01 -$	$4.54E-10 \pm 6.03E-10$
F_6	$4.78E-01 \pm 1.32E+00 \approx$	$1.84E-10 \pm 4.18E-10 +$	$3.18E-01 \pm 1.10E+00$
F_7	$9.86E-04 \pm 2.75E-03 +$	$5.80E-03 \pm 1.05E-02 -$	$1.67E-03 \pm 4.06E-03$
F_8	$2.09E+01 \pm 6.90E-02 \approx$	$2.00E+01 \pm 1.58E-02 +$	$2.09E+01 \pm 4.25E-02$
F_9	$3.43E-13 \pm 3.53E-13 +$	$0.00E+00 \pm 0.00E+00 +$	$2.56E-07 \pm 5.94E-07$
F_{10}	$3.02E+01 \pm 9.89E+00 -$	$5.18E+01 \pm 1.23E+01 -$	$2.23E+01 \pm 6.39E+00$
F_{11}	$1.74E+01 \pm 6.49E+00 -$	$9.16E+00 \pm 3.91E+00 -$	$1.45E+01 \pm 7.73E+00$
F_{12}	$3.89E+03 \pm 1.08E+03 -$	$4.75E+03 \pm 6.16E+03 -$	$1.71E+03 \pm 2.33E+03$
F_{13}	$2.01E+00 \pm 2.49E-01 -$	$1.95E+00 \pm 5.26E-01 -$	$1.78E+00 \pm 2.04E-01$
F_{14}	$1.21E+01 \pm 3.12E-01 \approx$	$1.28E+01 \pm 3.77E-01 -$	$1.23E+01 \pm 2.53E-01$
F_{15}	$3.36E+02 \pm 1.41E+02 +$	$3.52E+02 \pm 1.08E+02 +$	$4.00E+02 \pm 1.15E+02$
F_{16}	$8.89E+01 \pm 9.84E+01 -$	$1.64E+02 \pm 1.26E+02 -$	$5.48E+01 \pm 2.92E+01$
F_{17}	$1.47E+02 \pm 1.56E+02 -$	$1.36E+02 \pm 1.12E+02 -$	$6.41E+01 \pm 7.58E+01$
F_{18}	$9.04E+02 \pm 1.14E+00 \approx$	$9.04E+02 \pm 7.46E-01 \approx$	$9.03E+02 \pm 5.59E-01$
F_{19}	$9.04E+02 \pm 8.02E-01 \approx$	$9.04E+02 \pm 1.11E+00 \approx$	$9.03E+02 \pm 2.01E-01$
F_{20}	$9.04E+02 \pm 2.20E-01 \approx$	$9.04E+02 \pm 6.97E-01 \approx$	$9.03E+02 \pm 2.40E-01$
F_{21}	$5.00E+02 \pm 6.76E-14 \approx$	$5.00E+02 \pm 7.95E-14 \approx$	$5.00E+02 \pm 6.76E-14$
F_{22}	$8.73E+02 \pm 2.19E+01 -$	$8.68E+02 \pm 2.16E+01 \approx$	$8.69E+02 \pm 1.92E+01$
F_{23}	$5.50E+02 \pm 8.05E+01 -$	$5.49E+02 \pm 7.67E+01 -$	$5.34E+02 \pm 3.40E-13$
F_{24}	$8.66E+02 \pm 2.51E+02 -$	$2.00E+02 \pm 2.90E-14 \approx$	$2.00E+02 \pm 2.90E-14$
F_{25}	$2.09E+02 \pm 5.22E-01 \approx$	$2.10E+02 \pm 8.77E-01 \approx$	$2.09E+02 \pm 2.84E-01$
-/+/ \approx	10/4/11	13/4/8	

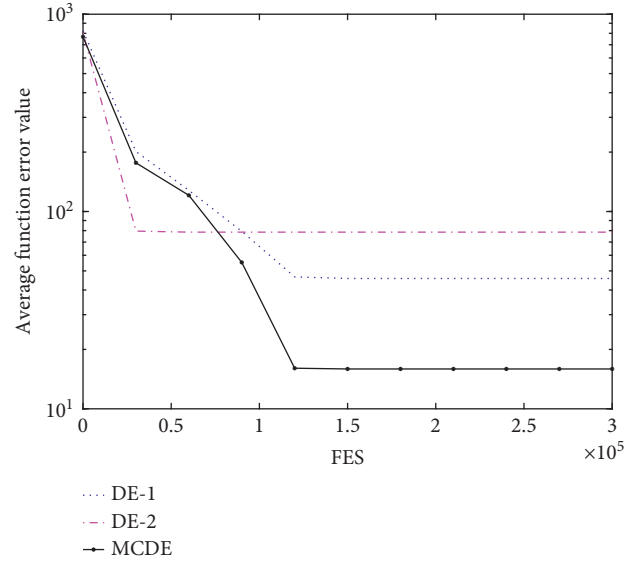
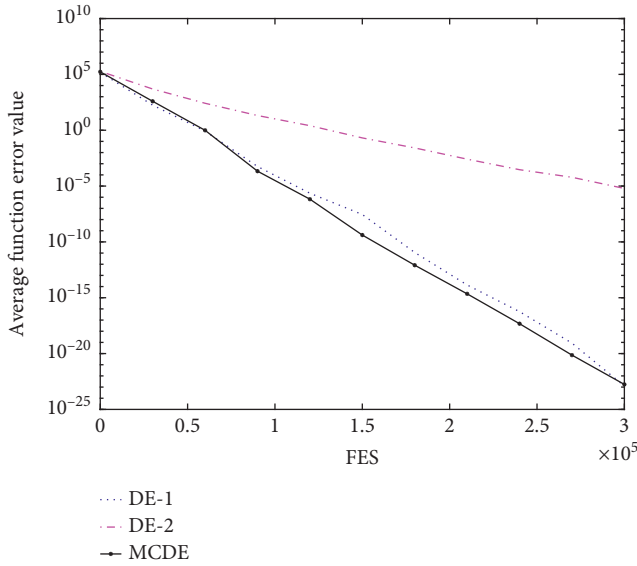
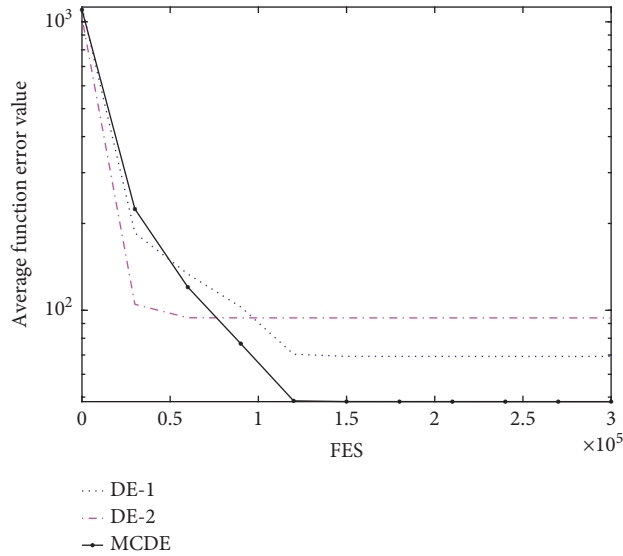
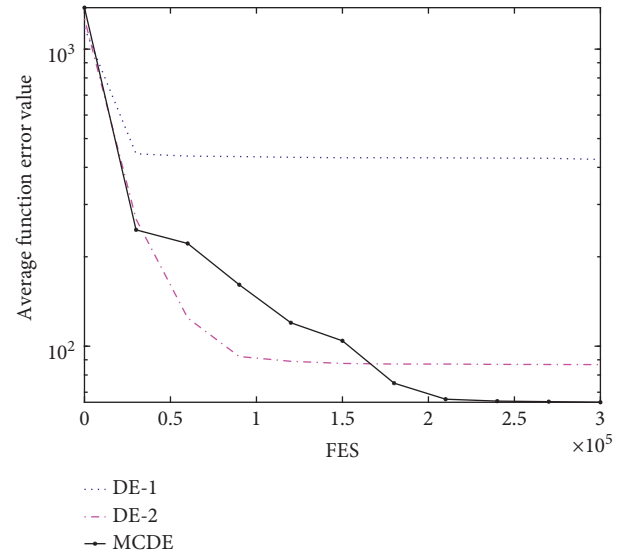


FIGURE 4: Continued.



(c)



(d)

FIGURE 4: Evolution of the mean function error values derived from DE-1, DE-2, and MCDE versus the number of FES on F_4 , F_{10} , F_{15} , and F_{17} . (a) The result of F_4 under $D = 30$. (b) The result of F_{10} under $D = 30$. (c) the result of F_{16} under $D = 30$. (d) The result of F_{17} under $D = 30$.

rotation coordinate system for the crossover operation in the population. At the same time, adaptive control parameters are adopted to balance population survey and algorithm convergence.

5. Conclusions

MCDE introduces multiscale strategies, including local mutation strategies and global mutation strategies, to expand the population search scope. During evolution, the initial coordinate system is properly rotated by the covariance learning matrix to rotate the objective individual and the variant individual. During the covariance learning, the excellent crossover probability CR and the scaling factor F were inherited from the previous generation by the Lehmer mean and the power mean, respectively. The proposed algorithm is compared with JADE, jDE, SaDE, EPSDE, CoDE, CoBiDE, and LSHADE on the CEC 2005 benchmark function, and it can be seen that there are significant effects on the global optimization problem with $D = 30$ and $D = 50$. To further verify the algorithm, we compare it with other evolutionary algorithms such as CLPSO, CMA-ES, and GL-25 in $D = 30$ and discover it works best. In terms of running time, the proposed algorithm is in the upper part of the comparison algorithms. In summary, both the accuracy and the convergence speed have improved, so MCDE can be implemented.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Yongzhao Du and Yuling Fan contributed equally.

Acknowledgments

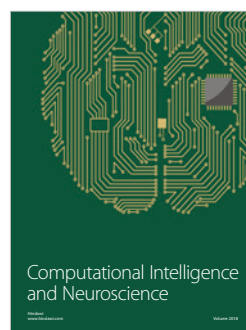
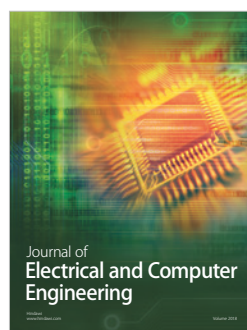
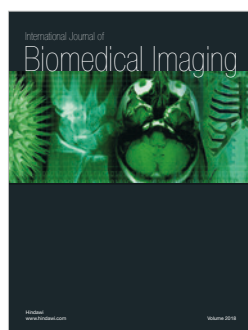
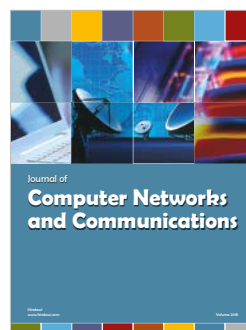
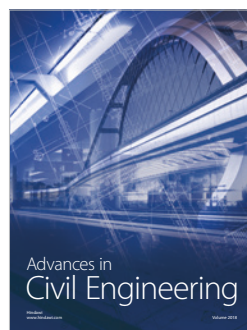
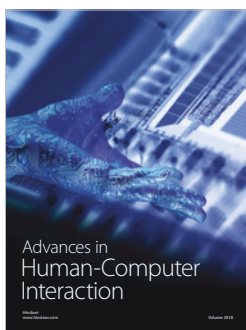
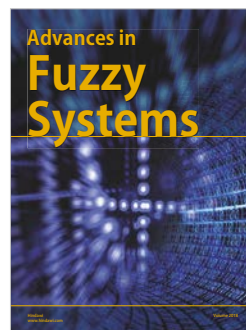
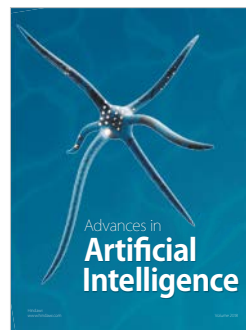
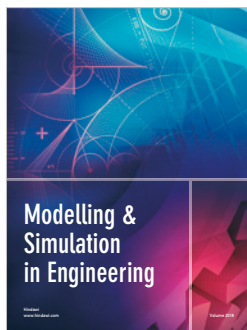
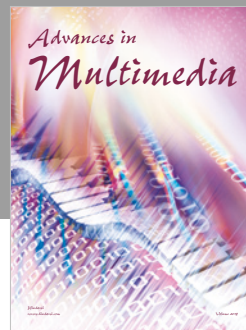
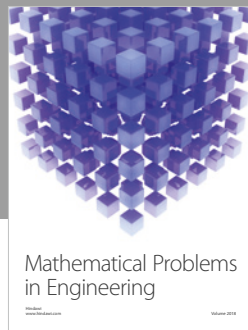
This study was supported by the Promotion Program for Young and Middle-Aged Teachers in Science and Technology Research of Huaqiao University (no. ZQN-PY518) and the grants from National Natural Science Foundation of China (grant nos. 61605048 and 61603144). This study was also supported in part by the Natural Science Foundation of Fujian Province, China (grant nos. 2015J01256 and 2016J01300), Quanzhou Scientific and Technological Planning Projects of Fujian, China (grant nos. 2015Z120 and 2017G024), and postgraduate research and innovation ability training program funds of Huaqiao University (grant no. 1611422002).

References

- [1] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [2] R. Storn and K. Price, "Differential evolution: a simple evolution strategy for fast optimization," *Dr. Dobbs' Journal*, vol. 22, pp. 18–24, 1997.
- [3] S. Biswas, S. Kundu, and S. Das, "An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution," *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1726–1737, 2014.
- [4] H. Guo, Y. Li, J. Li, H. Sun, D. Wang, and X. Chen, "Differential evolution improved with self-adaptive control parameters based on simulated annealing," *Swarm and Evolutionary Computation*, vol. 19, pp. 52–67, 2014.

- [5] L. Tang, Y. Dong, and J. Liu, "Differential evolution with an individual-dependent mechanism," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 560–574, 2015.
- [6] J. Wang, J. Liao, Y. Zhou, and Y. Cai, "Differential evolution enhanced with multiobjective sorting-based mutation operators," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2792–2805, 2014.
- [7] N. Pholdee and S. Bureerat, "Hybridisation of real-code population-based incremental learning and differential evolution for multiobjective design of trusses," *Information Sciences*, vol. 223, no. 6, pp. 136–152, 2013.
- [8] S. Sayah and A. Hamouda, "A hybrid differential evolution algorithm based on particle swarm optimization for non-convex economic dispatch problems," *Applied Soft Computing*, vol. 13, no. 4, pp. 1608–1619, 2013.
- [9] Y.-J. Zheng, X.-L. Xu, H.-F. Ling, and S.-Y. Chen, "A hybrid fireworks optimization method with differential evolution operators," *Neurocomputing*, vol. 148, no. 148, pp. 75–82, 2015.
- [10] J. Cheng, G. Zhang, and F. Neri, "Enhancing distributed differential evolution with multicultural migration for global numerical optimization," *Information Sciences*, vol. 247, no. 15, pp. 72–93, 2013.
- [11] M. Yang, C. Li, Z. Cai, and J. Guan, "Differential evolution with auto-enhanced population diversity," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 302–315, 2015.
- [12] W. Zhu, Y. Tang, J.-A. Fang, and W. Zhang, "Adaptive population tuning scheme for differential evolution," *Information Sciences*, vol. 223, no. 2, pp. 164–191, 2013.
- [13] L. Tang, Y. Zhao, and J. Liu, "An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 209–225, 2014.
- [14] M. A. Mangoud, M. T. Alshara, and H. M. Elragal, "Design of time modulated concentric circular and concentric hexagonal antenna array using hybrid enhanced particle swarm optimisation and differential evolution algorithm," *IET Microwaves, Antennas & Propagation*, vol. 8, no. 9, pp. 657–665, 2014.
- [15] V. Basetti and A. K. Chandel, "Hybrid power system state estimation using Taguchi differential evolution algorithm," *IET Science, Measurement and Technology*, vol. 9, no. 4, pp. 449–466, 2015.
- [16] Y. Zhong, L. Zhao, and L. Zhang, "An adaptive differential evolution endmember extraction algorithm for hyperspectral remote sensing imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 6, pp. 1061–1065, 2014.
- [17] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.
- [18] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithms using weight vectors," *European Journal of Operational Research*, vol. 243, no. 2, pp. 423–441, 2015.
- [19] D. Zaharie, "Critical values for the control parameters of differential evolution algorithm," in *Proceedings of the 8th International Mendel Conference on Soft Computing*, pp. 62–67, Brno, Czech Republic, 2002.
- [20] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [21] 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 Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.
- [22] W. Gong, Z. Cai, C. X. Ling, and C. Li, "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 2, pp. 397–413, 2011.
- [23] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [24] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.
- [25] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, and A. M. Sánchez, "Global and local real-coded genetic algorithms based on parent-centric crossover operators," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1088–1113, 2008.
- [26] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—an updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.
- [27] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.
- [28] H.-Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," *Journal of Global Optimization*, vol. 27, no. 1, pp. 105–129, 2003.
- [29] F. Caraffini, F. Neri, J. Cheng et al., "Super-fit multicriteria adaptive differential evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1678–1685, Cancun, Mexico, June 2013.
- [30] Y. Wang, Z. Cai, and Q. Zhang, "Enhancing the search ability of differential evolution through orthogonal crossover," *Information Sciences*, vol. 185, no. 1, pp. 153–177, 2012.
- [31] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 71–78, Cancun, Mexico, June 2013.
- [32] A. Zamuda, J. Brest, and E. Mezura-Monte, "Structured population size reduction differential evolution with multiple mutation strategies on CEC 2013 real parameter optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1925–1931, Cancun, Mexico, June 2013.
- [33] G. H. Wu, R. Mallipeddi, P. N. Suganthan, R. M. Duwairi, and R. G. Reynolds, "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences*, vol. 10, pp. 329–345, 2016.
- [34] M. Dubreuil, C. Gagne, and M. Parizeau, "Analysis of a master-slave architecture for distributed evolutionary computations," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 1, pp. 229–235, 2006.
- [35] F. Herrera and M. Lozano, "Gradual distributed real-coded genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 1, pp. 43–63, 2000.
- [36] E. Alba and B. Dorronsoro, "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 2, pp. 126–142, 2005.
- [37] G. Folion, C. Pizzuti, and G. Spezzano, "Training distributed GP ensemble with a selective algorithm based on clustering and pruning for pattern classification," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 4, pp. 458–468, 2008.

- [38] G. Roy, H. Lee, J. L. Welch et al., "A distributed pool architecture for genetic algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1177–1184, Trondheim, Norway, May 2009.
- [39] M. Ali, M. Pant, and A. Abraham, "Improved differential evolution algorithm with decentralisation of population," *International Journal of Bio-Inspired Computation*, vol. 3, no. 1, pp. 17–30, 2011.
- [40] Y.-L. Li and J. Zhang, "A new differential evolution algorithm with dynamic population partition and local restart," in *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference-GECCO-2011*, pp. 1085–1092, Dublin, Ireland, July 2011.
- [41] P. Novoa-Hernández, C. C. Corona, and D. A. Pelta, "Self-adaptive, multipopulation differential evolution in dynamic environments," *Soft Computing*, vol. 17, no. 10, pp. 1861–1881, 2013.
- [42] W.-J. Yu and J. Zhang, "Multi-population differential evolution with adaptive parameter control for global optimization," in *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference-GECCO-2011*, pp. 1093–1098, Dublin, Ireland, July 2011.
- [43] J. Zhang and X. Ding, "A multi-swarm self-adaptive and cooperative particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 6, pp. 958–967, 2011.
- [44] S.-Z. Zhao, P. N. Suganthan, Q.-K. Pan, and M. Fatih Tasgetiren, "Dynamic multi-swarm particle swarm optimizer with harmony search," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3735–3742, 2011.
- [45] P. N. Suganthan, N. Hansen, J. J. Liang et al., "Problem definitions and evaluation criteria for cec 2005 special session on real-parameter optimization," KanGAL report, 2005005, Nanyang Technological University, Singapore, 2005.
- [46] R. Tanabe and A. S. Fukunaga, "improving the search performance of shade using linear population size reduction," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation*, July 2014.
- [47] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [48] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.



Copyright of Computational Intelligence & Neuroscience is the property of Hindawi Limited and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.