

## Research Article

# Information Literacy Assessment with a Modified Hybrid Differential Evolution with Model-Based Reinitialization

Yuan Wang <sup>1</sup>, Hui Li,<sup>2</sup> and Zhenguo Ding<sup>3</sup>

<sup>1</sup>Faculty of Information Engineering, China University of Geosciences (Wuhan), No. 388 LuMo Road, Hongshan District, Wuhan, China

<sup>2</sup>School of Computer, China University of Geosciences (Wuhan), No. 388 LuMo Road, Hongshan District, Wuhan, China

<sup>3</sup>Institute of Higher Education, China University of Geosciences (Wuhan), No. 388 LuMo Road, Hongshan District, Wuhan, China

Correspondence should be addressed to Yuan Wang; wangyuan@cug.edu.cn

Received 28 January 2018; Revised 30 July 2018; Accepted 3 September 2018; Published 22 October 2018

Academic Editor: Massimo Panella

Copyright © 2018 Yuan Wang 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.

Information literacy assessment is extremely important for the evaluation of the information literacy skills of college students. Intelligent optimization technique is an effective strategy to optimize the weight parameters of the information literacy assessment index system (ILAIS). In this paper, a new version of differential evolution algorithm (DE), named hybrid differential evolution with model-based reinitialization (HDEMUR), is proposed to accurately fit the weight parameters of ILAIS. The main contributions of this paper are as follows: firstly, an improved contraction criterion which is based on the population entropy in objective space and the maximum distance in decision space is employed to decide when the local search starts. Secondly, a modified model-based population reinitialization strategy is designed to enhance the global search ability of HDEMUR to handle complex problems. Two types of experiments are designed to assess the performance of HDEMUR. In the first type of experiments, HDEMUR is tested and compared with seven well-known DE variants on CEC2005 and CEC2014 benchmark functions. In the second type of experiments, HDEMUR is compared with the well-known and widely used deterministic algorithm DIRECT on GKLS test classes. The experimental results demonstrate the effectiveness of HDEMUR for global numerical optimization and show better performance. Furthermore, HDEMUR is applied to optimize the weight parameters of ILAIS at China University of Geosciences (CUG), and satisfactory results are obtained.

## 1. Introduction

With the arrival of the new economic era based on information and knowledge, information has become an important factor in all fields of society. Information literacy (IL) skills which are the ability to locate, evaluate, and effectively use the needed information have become more important, especially to college students. At present, there were some college education information literacy standards such as the American Association of College and Research Libraries (ACRL) standard, including 5 first-level indexes, 22 second-level indexes, and 86 third-level indexes, the Australian and New Zealand Institute of Information Literacy (ANZIIL) standard consisting of 6 first-level indexes, 19

second-level indexes, and 67 third-level indexes, and the Society of College, National and University Libraries (SCONUL) standard in the United Kingdom which was composed of 7 first-level indexes and 17 second-level indexes. Nevertheless, in the face of the explosion of such digital information, it is difficult for college students to obtain the required information accurately.

During the last few decades, information literacy has attracted much attention by the researchers. Webber and Johnston [1] identified some key definitions of information and related the student response to two models of information literacy. Cooney and Hiris [2] described the use of a collaborative framework for integrating information literacy into a graduate finance course and used a checklist for

assessing the results. Valerie et al. [3] proposed the revised assessment method that took the form of a portfolio and reported the results of a case study evaluating the revision of the assessment methods of an information literacy module. The results showed that it was also economical and efficient. Pinto [4] designed the IL-HUMASS survey on information literacy, and it contained 26 items grouped into four categories and three self-reporting dimensions. Naimpally et al. [5] introduced a broad overview of the different assessment tools in information literacy assessment and highlighted four assessment tools for assessment of IL in engineering. Kousar and Mahmood [6] assessed IL skills of first year undergraduate engineering students at a Pakistani university in order to plan instruction. They used independent samples  $t$ -test and ANOVA to analyze the reliable data for integration of instruction in the university curricula. Krakowska [7] gathered the data from the questionnaire and analyzed it briefly in a quantitative and qualitative manner. The evaluation of the results helped to understand the students' information behavior, increased awareness of information literacy implementation, and development and status within the academic information environment. Pavlovski and Dunder [8] presented results of a survey regarding information literacy which was carried out on undergraduate students of University of Zagreb and gained an insight into the students' information retrieval accuracy in course-related research. Kultawanich et al. [9] reported on the development and validation of the information literacy assessment tools for undergraduate students. These assessment tools consisted of three instruments: (1) IL Test, (2) IL Rubric, and (3) Information Literacy Self-Efficacy (ILSE) Scale. Douglas et al. [10] presented the development and initial validation study of a self-directed information literacy assessment for engineering and technology students. The exploratory factor analysis results provided evidence of structural aspects of validity and support for scoring structure. Kavšek et al [11] used two groups of first year psychology students to evaluate the effect of information literacy (IL) training in the first year psychology students and to follow changes in acquired IL in time. The results had shown an important role of IL training in students' IL development over time. Lanning and Mallek [12] analyzed multiple factors from current university students' high school experiences to evaluate the students' information literacy skills. The results of regression analyses demonstrated that only current university GPA and standardized test scores had any influence on information literacy test scores.

An optimization problem is the problem of finding the best solution from all feasible solutions. Optimization problems can be divided into two categories depending on whether the variables are continuous or discrete. The standard form of an optimization problem is formulated as

$$\begin{aligned} \min_{x \in D_0} f(x), \\ D = \left\{ x \in D_0 \mid f_j(x) \leq 0, \quad j = 1, \dots, J \right\}, \end{aligned} \quad (1)$$

where  $D_0 = \{x \in E^n \mid \ell \leq x \leq u\}$  is a simple box constraint set. Problem category included discrete domain and continuous domain. Optimization approaches included deterministic search methods (such as direct search methods, simplex-based search, and branch and bound search methods) and nondeterministic (heuristic) search methods (such as simulated annealing, genetic algorithms, differential evolution, and particle swarm optimization).

In the deterministic search methods, the researchers proposed the global optimization methods that can tackle multimodal optimization problems satisfying the Lipschitz condition [13]. Sergeyev and Kvasov [14] proposed a new efficient algorithm for solving the multidimensional "black-box" functions satisfying the Lipschitz condition. In this algorithm, a novel technique balancing usage of local and global information during partitioning, and a new procedure for finding lower bounds of the objective function over hyperintervals were considered. Jones et al. [15, 16] proposed the well-known and widely used algorithm DIRECT (DIviding RECTangles) which objective function was evaluated at several sample points by using all possible weights on local versus global search expressed by the Lipschitz constant during each iteration. Liuzzi et al. [17] focused on the selection strategy and proposed two strategies to exploit the information on the objective function. The first one was based on the knowledge of the global optimum value of the objective function. The second one did not require any a priori knowledge on the objective function and tried to exploit information on the objective function gathered during progress of the algorithm. Paulavičius et al. [18] proposed a globally biased simplicial partition DISIMPL algorithm for global optimization of expensive Lipschitz continuous functions to improve the search efficiency of the algorithm. Gimbutas and Žilinskas [19] introduced a bi-criteria choice of simplices for the subdivision to enhance the efficiency of the simplicial partition-based minimization of black box objective functions. The first criterion was minimum of an estimated Lipschitz bound, and the second was the size of the candidate simplex. Sergeyev et al. [13] proposed a new efficient visual technique for a systematic comparison of global optimization algorithms and tried to bridge the gap between the deterministic methods and heuristic methods.

In the heuristic search methods, DE and its variants have emerged as one of the most competitive and versatile families of evolutionary algorithms which belong to computational intelligence. It was first proposed by Storn and Price [20] to solve global numerical optimization problems over continuous search spaces. It is a simple yet powerful evolutionary algorithm and exhibits excellent capability in solving a variety of numerical and real-world optimization problems, such as space trajectory design [21–25], hydrothermal optimization [26], underwater glider path planning [27], vehicle routing problem [28, 29], short-term optimal hydrothermal scheduling [30], satellite scheduling [31, 32], and satellite image enhancement [33].

During the last decade, there were many researchers working on the improvement of DE and significant progress in the developments of DE. In 2011 and 2016, Das et al. [34, 35] presented a comprehensive survey on DE, including the basic concepts and major variants of DE, as well as the applications and theoretical studies of DE. Next, we will briefly introduce the recent developments of hybrid DE in the last two years.

Hybrid DE was tried to combine DE with other global or local search techniques. Awad et al. [36] introduced a novel hybridization between differential evolution and update processes of the stochastic fractal search algorithm. Ali et al. [37] and Awad et al. [38], respectively, focused on the development of hybridizing cultural algorithm (CA) with DE. The purpose was to combine the explorative and exploitative capabilities of two evolutionary algorithms. Cai et al. [39] proposed an adaptive social learning (ASL) strategy for DE, named SL-DE, to extract the neighborhood relationship information of individuals in the current population. Jadon et al. [40] proposed a hybridization of artificial bee colony (ABC) and DE algorithms, called HABCDE, to develop a more efficient meta-heuristic algorithm than ABC and DE. Peng et al. [41] hybridized DE with commensal learning and uniform local search (CUDE). In CUDE, commensal learning was proposed to adaptively select optimal mutation strategy and parameter setting simultaneously under the same criteria. Moreover, uniform local search enhanced exploitation ability. Zhao et al. [42] proposed a hybrid optimization algorithm based on chaotic differential evolution and estimation of distribution (cDE/EDA). The proposed algorithm could discover the optimal solution in a fast and reliable manner. Chaotic policy was used to strengthen the search ability of DE. Peng et al. [43] proposed an improved memetic differential evolution algorithm, called MDE, which hybridized differential evolution with a local search (LS) operator and periodic reinitialization to balance exploration and exploitation for solving global optimization problems. Under the framework of LSHADE in [44], Mohamed et al. [45] further proposed a new semi-parameter adaptation approach to effectively adapt the values of the scaling factor, named LSHADE-SPA. And then, a hybridization framework named LSHADE-SPACMA, a combination of LSHADE-SPA and a modified version of CMA-ES, were introduced. Sabar et al. [46] proposed a heterogeneous framework that integrated a cooperative coevolution method with various types of memetic algorithms for solving big data optimization problems. In this heterogeneous framework, DE was utilized to optimize the subproblems generated by the cooperative coevolution method.

This work is motivated by the following observations:

- (1) The main difficulties in college students' information literacy assessment index system are the logical structure of the index system and how to assign weight parameters to the index system. There are mainly two ways to determine weights: one is subjective assignment, and the other is objective assignment. Subjective assignment mainly refers to

determining weights according to the human experience and knowledge. Objective assignment is based on the data of questionnaire. Hence, it is very necessary for ILAIS to use advanced optimization techniques to fit weight parameters.

- (2) The investigations in [21–33] indicate that DE has been successfully used in a variety of domains. However, the use of DE for the weight parameters determination of ILAIS has not been reported.
- (3) As a kind of intelligent optimization techniques, DE produces relatively good results in different fields. However, DE has been demonstrated to converge to a fixed point, a level set [23] or a hyperplane not containing the global optimum [47]. Furthermore, in some cases it is shown to have slow local convergence. Thus, integrating additional strategies to improve DE optimization performance is worth investigating.

Based on the above considerations, in this paper, we propose a modified hybrid DE, called HDEMUR, with the local search algorithm Broyden–Fletcher–Goldfarb–Shanno (BFGS) to improve local convergence. A modified contraction criterion based on the entropy of population in objective space and the maximum distance in decision space is employed to trigger local search. In addition, we define an archive to save the best individual in the population after the local search. And then, the population is reinitialized based on the probability model of the archive.

The paper is organized as follows: in Section 2, the college students' information literacy assessment index system and the weight parameter optimization model of ILAIS are introduced. DE is briefly introduced in Section 3. The proposed algorithm is described in details in Section 4. The experimental study and results analysis are presented in Section 5. HDEMUR for weight parameter optimization of ILAIS is derived in Section 6. Finally, in Section 7, we conclude this paper and suggest ideas for future work.

## 2. The Weight Parameter Optimization Model of ILAIS

For the sake of completeness, in this section, the college students' ILAIS at China University of Geosciences (CUG) is briefly described first. Then, the weight parameter optimization model of ILAIS based on an objective function is introduced.

*2.1. The Establishment of ILAIS at CUG.* Based on the new requirements of education and college students' feature, ILAIS at CUG is established into four categories: Information Consciousness, Information Knowledge, Information Ability, and Information Morality. We treat these four categories as first-level indexes. Then, for reference-related standards such as ACRL, we gradually decompose first-level indexes and initially build thirteen second-level indexes of ILAIS at CUG. Finally, the framework of ILAIS at CUG

TABLE 1: Information literacy assessment index system at CUG.

First-level index	Second-level index
L1: Information Consciousness	L11: the recognition of the value of information and the objective evaluation of the role of information
	L12: attitudes towards various social problems involving in the process of access to and use of information
	L13: recognise the right useful information
L2: Information Knowledge	L21: effectively select information retrieval tools and know the advantages and disadvantages of different information retrieval tools
	L22: have information source knowledge, and understand the value of various sources of information and communication process
	L23: identify reliable and significant information sources, master basic information science knowledge
L3: Information Ability	L31: the ability of information retrieval
	L32: the ability to use and process information
	L33: the ability to share, deliver, and create information ability
L4: Information Morality	L34: the ability to learn information knowledge independently and information development cooperation
	L41: ability to master information law
	L42: information security and privacy
	L43: information ethic cognition and behavior

includes four first-level indexes and thirteen second-level indexes, which is shown in Table 1.

**2.2. The Weight Parameter Optimization Model.** Before using optimization techniques to identify the weight parameters of the ILAIS at CUG, there are two issues that must be addressed. First, the weight parameters that will be optimized must be chosen. In the above ILAIS at CUG established in Table 1, the number of weight parameters of ILAIS at CUG is related to the level. To first-level index, there are four weight parameters, and to second-level indexes, there are thirteen weight parameters.

In this work, all weight parameters are treated as unknown parameters to be identified by the optimization algorithm. Thus, the decision vector  $x$  is formulated as

$$x = \{x_1, x_2, \dots, x_D\}, \quad (2)$$

where  $D$  is the number of decision variables. An example for the determination of the first-level index weight parameters is given here. Let the weight of Information Consciousness be  $x_1$ , the weight of Information Knowledge be  $x_2$ , the weight of Information Ability be  $x_3$ , and the weight of Information Morality be  $x_4$ .

The other important issue in evolutionary algorithms is to determine an objective function. When optimization techniques are used in parameter identification problems of ILAIS, the objective function should be defined first. In this work, the weight deviation sum is used as the objective function in the following equation:

$$\min f(x) = t * \sum_{i=1}^{N_p} \sum_{p=1}^P (|x_{p,j} - x_{i,j}|) + s * \sum_{i=1}^{N_p} \sum_{q=1}^Q (|x_{q,j} - x_{i,j}|), \quad (3)$$

where  $P$  and  $Q$  are, respectively, the number of teachers and students participating in the questionnaire.  $t$  and  $s$  are the authority coefficient of teachers and students.  $x_{p,j}$  is the weight vector given by the  $p$ -th teacher and  $x_{q,j}$  is the weight vector given by the  $q$ th student.  $x_{i,j}$  is the  $i$ -th weight individual,  $N_p$  is the population size, and  $j = 1, 2, \dots, D$  in HDEMUR.

### 3. A Short Introduction to Differential Evolution

DE is a population-based stochastic parallel direct search optimization method. It begins with a randomly generated population in decision space. Then, DE iteratively uses a trial vector generation strategy (i.e., mutation and crossover operators) and selection operator to evolve the population until a stopping criterion is met.

For a mutation operator, there are five frequently used mutation schemes for generating a mutant vector. It can be denoted as DE/ $a/b$ , where  $a$  stands for the mutation strategy and  $b$  specifies the number of difference vectors. The following are the five most frequently used mutation strategies:

$$\text{“DE/rand/1”} : v_i = x_{r_1} + * (x_{r_2} - x_{r_3}), \quad (4)$$

$$\text{“DE/best/1”} : v_i = x_{\text{best}} + F * (x_{r_1} - x_{r_2}), \quad (5)$$

$$\text{“DE/current-to-best/1”} : v_i = x_i + F * (x_{\text{best}} - x_i) + F * (x_{r_1} - x_{r_2}), \quad (6)$$

$$\text{“DE/best/2”} : v_i = x_{\text{best}} + F * (x_{r_1} - x_{r_2}) + F * (x_{r_3} - x_{r_4}), \quad (7)$$

$$\text{“DE/rand/2”} : v_i = x_{r_1} + F * (x_{r_2} - x_{r_3}) + F * (x_{r_4} - x_{r_5}), \quad (8)$$

where  $r_1, r_2, r_3, r_4, r_5 \in \{1, \dots, N_p\}$  and  $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$ .  $v_i$  is the mutant vector.  $x_{\text{best}}$  refers to the best fitness in the population at current generation. The scaling factor  $F$  controls the difference vectors with a positive value.

To diversify the current population, following mutation, DE uses a crossover operator to produce the trial vector  $u_i$  between  $x_i$  and  $v_i$ . The most commonly used operator is the binomial crossover performed on each component as follows:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{rand}[0, 1] \leq CR \text{ or } j = j_{\text{rand}}, \\ x_{i,j}, & \text{otherwise,} \end{cases} \quad (9)$$

where  $i = 1, \dots, N_p$ ,  $j = 1, \dots, D$ ,  $\text{rand}[0, 1]$  represents a number drawn uniformly between 0 and 1,  $CR \in [0, 1]$  is



the crossover rate, and  $j_{\text{rand}}$  is a randomly selected integer within  $[1, D]$ .

After mutation and crossover, each generated trial vector undergoes boundary constraint check. If some variables of trial vector are out of the boundary, a repair method is applied as follows:

$$x_{i,j} = x_j^{\min} + \text{rand}[0, 1] \cdot (x_j^{\max} - x_j^{\min}), \quad (10)$$

$$\text{if } x_{i,j} < x_j^{\min} \text{ or } x_{i,j} > x_j^{\max},$$

where  $x_j^{\min}$  and  $x_j^{\max}$  are the lower bound and upper bound, respectively,  $x_j \in [x_j^{\min}, x_j^{\max}]$ .  $\text{rand}[0, 1]$  returns a uniformly distributed random number between 0 and 1 ( $0 \leq \text{rand}[0, 1] \leq 1$ ).

At last, the selection operator is performed to select the better one between  $x_i$  and  $u_i$  to enter the next generation:

$$x_i = \begin{cases} u_i, & f(u_i) \leq f(x_i), \\ x_i, & \text{otherwise,} \end{cases} \quad (11)$$

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

## 4. Proposed Approach

In this section, we propose a novel DE, named HDEMUR. HDEMUR contains three main components: improved contraction criterion, BFGS local search, and model-based reinitialization strategy. And then, the complete proposed framework of HDEMUR is shown in Algorithm 1. Next, the implementation of the above three main components will be introduced in detail.

**4.1. Improved Contraction Criterion.** In order to design an effective and efficient hybrid algorithm for global optimization, we need to take advantage of both the exploration capabilities of EA and the exploitation capabilities of LS and combine them in a well-balanced manner. For successful incorporation of LS in DE, a triggering condition, called contraction criterion, is needed to decide when the local search has to start. There are several kinds of methods to define a contraction criterion. Qin and Suganthan [48] applied a local search method after a fixed number of generations (every 200 generations). Sun et al. [49] used the condition that if the promising solution was not updated in  $t$ -consecutive generations, LS would start. Simon et al. [50] used the minimum fitness in the objective space as the contraction criterion. Vasile et al. [23] performed LS when the maximum distance in decision space was below a given threshold. Peng et al. [43] proposed a new contraction criterion which combined the improved maximum distance in objective space and the maximum distance in decision space.

To have a good judgement to the individuals' distribution, a good measure to evaluate the crowding degree around each solution is needed as a mixed triggering condition. In HDEMUR, we also propose an improved contraction criterion which combines two criteria: (a)  $E$  is the population entropy in objective space and (b)  $S$  is the maximum distance in decision space.

**Theorem 1.** Let  $R$  be the range of the fitness function values of the population in the proposed algorithm. If  $A_1 \cup A_2 \cup \dots \cup A_n = R$  and  $A_1 \cap A_2 \cap \dots \cap A_n = \emptyset$ , where  $A_1, A_2, \dots, A_n$  are the true subsets of the population, the population entropy of HDEMUR is defined as:

$$E = - \sum_{i=1}^n (p_i \cdot \log p_i), \quad p_i = s_i/N_p, \quad (12)$$

where  $s_i$  is the number of individuals in the subset  $A_i$  and  $N_p$  is the population size in HDEMUR.  $E$  is the measure to represent the diversity of the population in objective space.

The distance in decision space is defined as

$$S = \text{Max}(\|x_i - x_j\|), \quad \forall x_i, x_j \in \mathbb{P}, \quad (13)$$

where  $\|\cdot\|$  is the Euclidean distance and  $S$  is the measure to represent the diversity of the population in decision space.

**4.2. BFGS Local Search.** The local search utilizes the better solutions obtained by the global search to update the population and thus enhances algorithm's exploitation ability to find the best solution. In HDEMUR, we use the BFGS algorithm as the local search method. It is one of the quasi-Newton methods which do not need the precise Hessian matrix and be able to approximate it based on the individual successive gradients. BFGS is considered to be the most effective and popular quasi-Newton method and has been proven to have good performance for solving unconstrained nonlinear optimization problems. The details can be found in [51, 52].

**4.3. Model-Based Reinitialization Strategy.** After the local search is performed, if the best solution has not been improved, a reinitialization of the whole population is used to give the algorithms more opportunities to find the global optimum. Sun et al. [49] chose the individuals that had the largest distances from the local optima to form the next population from a temporary population. Simon et al. [50] proposed a partial reinitialization of the population. Every 20 generations, the algorithm selected the best  $M$  individuals from a temporary population of  $2M + 2$  individuals as the reinitialization pool. Zamuda et al. [53] proposed a population size reduction method as the reinitialization strategy. In this paper, we propose a model-based reinitialization strategy. If the local search does not improve the best solution in the current generation, the population will be restarted. A counter  $C$  keeps track of the number of restarts. For  $C < C_{\max}$ , where  $C_{\max}$  is user-defined, the new population,  $\mathbb{P}'$ , is generated randomly in the search space. For  $C \geq C_{\max}$ ,  $\mathbb{P}'$  is initialised from a Gaussian distribution model base on the mean and standard deviation of the best solutions in the Archive. Algorithm 2 summarises the model-based reinitialization procedure.

## 5. Experimental Study

To assess the performance of HDEMUR, three groups of experiments are conducted. We compare HDEMUR with

```

Input: Control parameters:  $N_p$ ,  $D$ , Max_NFEs, Flag,  $C$ ,  $E_{\max}$  and  $S_{\max}$ ;
Output: The best final solution;
Initialize the population  $\mathbb{P}$  randomly;
 $C = 0$ ;
Calculate the objective function value of each solution in the population  $\mathbb{P}$ ;
NFEs =  $N_p$ ;
while NFEs < Max_NFEs do
  for  $i = 1$  to  $N_p$  do
     $CR \in N(0.8, 0.1)$ ,  $F \in N(0.5, 0.1)$ ;
    Select  $x_{r1}$ ,  $x_{r2}$ ,  $x_{r3}$  and  $r1 \neq r2 \neq r3 \neq i$ ;
    Produce the mutant vector  $v_i$  in Equation (3);
    Produce the trial vector  $u_i$  in Equation (8);
    Apply the boundary constraint check to the violated solution as shown in Equation (9);
    Evaluate the trial vector  $u_i$ . NFEs = NFEs + 1;
    if  $u_i$  is better than  $x_i$  then
       $x_i = u_i$ 
  Calculate the contraction criterion in Equations (12) and (13) as described in Section 4. A;
  if  $E < E_{\max}$  or  $S < S_{\max}$  then
    /*  $x_{\text{best}}$  is the best solution in the current generation,  $x_{g\text{min}}$  is the global optimum */;
    Pick up the  $x_{\text{best}}$  as the initial point of the local search;
    Apply BFGS( $x_{\text{best}}$ ) to find the resultant new local optimum  $x_{\text{local}}$ ;
    if  $x_{\text{local}}$  is better than  $x_{\text{best}}$  then
      Replace  $x_{\text{best}}$  with  $x_{\text{local}}$ ;
      Flag = 0; NFEs = NFEs + 1;
    else
      Flag = 1;
    Archive  $x_{\text{best}}$  directly;
    if  $x_{\text{best}}$  is better than  $x_{g\text{min}}$  then
      Replace  $x_{g\text{min}}$  with  $x_{\text{best}}$ ;
    if Flag == 1 then
      Run model-based reinitialization to create a new population  $\mathbb{P}'$  as described in Algorithm 2;

```

ALGORITHM 1: The pseudocode of the HDEM algorithm.

```

Input: Control parameters:  $C_{\max}$ ,  $M$  is the number of individuals in the Archive;
Output: The new population  $\mathbb{P}'$ ;
if  $C < C_{\max}$  then
  /* reinitialize the population  $\mathbb{P}'$  randomly */;
  for  $i = 1$  to  $i = 1$  do
    for  $j = 1$  to  $D$  do
       $x_{i,j} = x_j^{\min} + \text{rand}[0, 1] \cdot (x_j^{\max} - x_j^{\min})$ ;
  else
    /* reinitialize the population  $\mathbb{P}'$  based on the probability distribution model of the Archive */;
    for  $j = 1$  to  $D$  do
      for  $i = 1$  to  $M$  do
         $\mu_j = \bar{x}_{i,j} = 1/M \cdot \sum_{i=1}^M x_{i,j}$ 
        for  $i = 1$  to  $M$  do
           $\sigma_j = \sqrt{1/M - 1 \cdot \sum_{i=1}^M (x_{i,j} - \bar{x}_{i,j})^2}$ 
      for  $i = 1$  to  $N_p$  do
        for  $j = 1$  to  $D$  do
           $x_{i,j} = \text{Gaussian}(\mu_j, \sigma_j^2)$ ;

```

ALGORITHM 2: The pseudocode of model-based reinitialization.

seven well-known algorithms including JADE [54], CoDE [55], jDE [56], MPEDE [57], SHADE [58], LSHADE [44], and LSHADE- $\varepsilon$  [59]. In the first group of experiments, we select the 21 nonnoisy benchmark functions (excluding

noisy functions F4, F17, F24, and F25) at IEEE CEC2005 [60]. In the second group of experiments, we use 30 test functions at IEEE CEC2014 [61] to further demonstrate the effectiveness of HDEM. In the third group of

experiments, we compare HDEMUR with the well-known and widely used deterministic algorithm DIRECT [15, 16] on GKLS test classes [62] to further verify the performance of HDEMUR.

*5.1. Performance Test of HDEMUR on CEC2005 Benchmark Functions.* For each algorithm, we conduct 25 independent runs and limit each run to  $10000 \cdot D$  max function evaluations at 21 benchmark problems as in [60]. Because the dimension of weight parameter identification of ILAIS is approximately 10, HDEMUR is just tested by the benchmark problems at  $D = 10$  to evaluate its performance at low-dimensional problem. The algorithms are evaluated in terms of function error value [60], defined as  $f(x) - f(x^*)$ , where  $x^*$  is the global optimum of the test function. Mean error and standard deviation of the function error values are recorded. The parameters of HDEMUR are set as  $N_p = 30$ ,  $E_{\max} = 3.0$ ,  $S_{\max} = 2.0$ ,  $C_{\max} = 9$ ,  $CR \in N(0.8, 0.1)$ , and  $F \in N(0.5, 0.1)$ . For the other seven well-known algorithms, we use the same parameter settings as in their original papers.

To effectively analyze the results, two nonparametric statistical tests, as similarly done in [63], are used in the experiments. (i) The Wilcoxon rank-sum test at  $\alpha = 0.05$  is performed to test the statistical significance of the experimental results between two algorithms in single-problem and multiproblem. (ii) The Friedman test is employed to obtain the average rankings of all the compared algorithms. The Wilcoxon rank-sum test in single-problem is calculated by Matlab, while the Wilcoxon test in multiproblem and the Friedman test are carried out by the software KEEL [64].

According to the results shown in Table 2 (last three rows), HDEMUR performs significantly better than JADE on 8 test functions, and JADE wins in 5 test functions. Compared with CoDE, HDEMUR shows better and worse performance on 10 test functions and 7 test functions, respectively. HDEMUR wins 12 test functions and loses 3 test functions to jDE. MPEDE surpasses HDEMUR on 2 test functions; however, HDEMUR is significantly better than MPEDE on 11 test functions. HDEMUR outperforms SHADE on 5 test functions and is worse than SHADE on 7 test functions. Moreover, HDEMUR performs better than LSHADE on 3 functions and worse than LSHADE on 8 functions. Compared with LSHADE- $\epsilon$ , HDEMUR shows better and worse performance on 8 test functions and 7 test functions, respectively. Furthermore, HDEMUR obtains similar results with other seven algorithms in 8, 4, 6, 8, 9, 10, and 6 cases.

In addition, some interesting phenomena can be observed according to the experimental results in Table 2. For the unimodal functions (i.e., F1, F2, and F3), the performance of HDEMUR is significantly better than that of the other algorithms. Moreover, HDEMUR also outperforms the other algorithms on F6 and F8. To hybrid composition functions (i.e., F18–F23), HDEMUR exhibits the best performance among the eight algorithms. The outstanding performance of HDEMUR on hybrid composition functions

can be attributed to its capability to balance the exploration and exploitation. JADE, CoDE, SHADE, LSHADE, and LSHADE- $\epsilon$  perform quite well on the expanded multimodal test functions (i.e., F13 and F14), which means history-based parameter and trial vector generation adaptation strategies can effectively improve the performance of advanced DE to the expanded multimodal functions.

In order to further validate the performance of model-based reinitialization strategy, we compare HDEMUR with HDE (hybrid differential evolution with reinitialization randomly). Table 3 shows the results of HDEMUR and HDE on CEC2005 benchmark functions. It can be seen that HDEMUR performs significantly better than HDE on 8 test functions. HDE wins on 1 test function. Also, HDEMUR obtains similar results with HDE on 12 test functions. Some interesting phenomena can be observed according to the experimental results. HDEMUR performs quite well on most of the hybrid composition functions (i.e., F15 and F18–F22), which further demonstrates that model-based reinitialization strategy can effectively improve the performance of HDEMUR to the difficult functions.

To further detect the significant differences between HDEMUR and the seven competitors, the multiple-problem Wilcoxon test and the Friedman test are carried out, respectively. As shown in Table 4, it can be seen that HDEMUR can obtain higher  $R^+$  values than  $R^-$  values in all cases, where  $R^+$  is the sum of ranks for the functions on which HDEMUR outperforms the compared algorithm, and  $R^-$  is the sum of ranks for the opposite [63]. According to the Wilcoxon test at  $\alpha = 0.05$ , the significant differences can be observed in three cases (i.e., HDEMUR vs. JADE, HDEMUR vs. jDE, and HDEMUR vs. MPEDE). When  $\alpha = 0.1$ , there are significant differences in five cases (i.e., HDEMUR vs. JADE, HDEMUR vs. jDE, HDEMUR vs. MPEDE, HDEMUR vs. SHADE, and HDEMUR vs. LSHADE- $\epsilon$ ), which means that HDEMUR is significantly better than JADE, jDE, MPEDE, SHADE, and LSHADE- $\epsilon$  on 21 test functions at  $\alpha = 0.1$ . Moreover, the results shown in Figure 1 indicate that HDEMUR and LSHADE have the best ranking (3.4762) among the eight compared algorithms. In general, the above comparison clearly demonstrates that HDEMUR and LSHADE are significantly better than the other competitors at CEC2005 benchmark functions.

*5.2. Sensitivity in Relation to the Parameters  $E_{\max}$  and  $S_{\max}$ .* The improved contraction criterion in HDEMUR contains two parameters  $E_{\max}$  and  $S_{\max}$ . They are the triggering condition to the local search. In order to investigate the sensitivity of the above two parameters, we test HDEMUR with different  $E_{\max} = \{1.0, 2.0, 3.0\}$  and  $S_{\max} = \{1.0, 2.0, 3.0\}$ . Nine different combinations of  $E_{\max}$  and  $S_{\max}$  are done. The statistical results by the Friedman test with all initial values are shown in Table 5.

From Table 5, we can see that HDEMUR provides the best average ranking value at  $E_{\max} = 3.0$  and  $S_{\max} = 2.0$  than other 8 combinations on the test functions. In general, we can conclude that it is better to set a large value to  $E_{\max}$  and a small value to  $S_{\max}$ .

TABLE 2: Comparison of mean error and standard deviation between HDEMIR and other seven DE variants on CEC2005 at  $D = 10$ .

Function	JADE	CoDE	jDE	MPEDE	SHADE	LSHADE	LSHADE- $\epsilon$	HDEMIR						
F01*	6.27E -03	3.07E -02	1.07E -03	7.14E 1.51E	1.29E 2.74E	5.39E 6.07E	2.32E 3.46E	6.00E 8.73E	2.32E 3.46E	1.38E 0.00E	1.58E 0.00E	1.05E 0.00E	1.58E 0.00E	0.00E 0.00E
F02	0.00E +00	3.03E -30	1.51E -29	1.51E -29	2.74E -17	6.07E -17	3.46E -15	8.73E -15	3.46E -15	0.00E +00	0.00E +00	0.00E +00	0.00E +00	0.00E +00
F03	4.81E -26	1.80E -16	5.02E -15	1.39E -15	1.54E -06	6.01E -06	8.61E -11	1.67E -10	8.61E -10	2.45E -26	5.96E -26	2.70E -26	2.70E -26	0.00E +00
F05	0.00E +00	0.00E +00	0.00E +00	0.00E +00	7.28E -14	3.64E -13	2.99E -06	1.70E -06	2.99E -06	0.00E +00	0.00E +00	0.00E +00	0.00E +00	1.53E -02
F06	2.35E +00	2.56E +00	1.02E -23	3.42E -23	3.97E -02	7.00E -02	1.16E -01	9.43E -02	1.16E -01	0.00E +00	0.00E +00	0.00E +00	0.00E +00	0.00E +00
F07	2.12E -02	9.98E -03	5.06E -02	4.16E -02	1.27E +03	4.20E -13	2.07E -02	2.03E -02	2.07E -02	8.77E -03	2.85E -03	6.20E -03	4.88E -03	1.93E -02
F08	2.03E +01	6.99E -02	2.01E +01	1.15E -01	2.03E +01	7.47E -02	2.03E +01	8.21E -02	2.03E +01	1.60E -01	2.01E +01	1.03E -01	1.09E -01	2.00E +01
F09*	1.33E +01	1.88E +00	7.47E +00	1.97E +00	9.78E +00	2.07E +00	2.64E +01	4.23E +00	2.64E +01	2.76E +01	7.79E +00	7.90E -01	1.82E +00	4.57E -01
F10	4.21E +00	1.01E +00	7.96E +00	2.72E +00	1.04E +01	2.75E +00	6.92E +00	1.95E +00	2.73E +00	9.31E -01	8.27E -01	1.04E +00	8.27E -01	5.77E +00
F11	4.40E +00	8.05E -01	4.87E -01	6.39E -01	5.86E +00	6.98E -01	5.68E +00	6.59E -01	4.88E +00	7.14E -01	6.47E -01	7.50E -01	6.47E -01	6.06E -01
F12	1.41E +01	1.98E +01	1.16E +01	4.24E +01	6.55E +01	3.11E +02	2.32E +00	5.68E +00	3.05E +01	1.42E +02	4.90E +00	4.36E +00	4.90E +00	0.00E +00
F13	3.39E -01	3.80E -02	2.91E -01	1.07E -01	4.35E +00	5.87E -02	5.30E -02	6.54E -02	2.82E -01	4.65E -02	4.24E -02	3.67E -02	4.24E -02	3.96E -01
F14	2.75E +00	2.47E -01	2.17E +00	7.65E -01	3.36E +00	1.63E -01	3.10E -01	2.21E -01	2.58E +00	3.38E -01	2.09E +00	2.93E -01	3.73E -01	3.12E -01
F15	9.38E +01	1.47E +02	8.01E +01	1.44E +02	1.65E +01	3.08E +01	2.86E +01	8.22E +01	1.05E +02	1.72E +02	1.07E +02	1.90E +02	1.57E +02	3.55E +01
F16	9.76E +01	3.34E +00	1.02E +02	9.54E +00	1.13E +02	7.42E +00	1.02E +02	5.51E +00	9.39E +01	6.78E +00	9.13E +01	2.66E +00	5.07E +00	1.11E +02
F18	6.33E +02	2.56E +02	5.00E +02	2.50E +02	5.60E +02	2.55E +02	6.00E +02	2.50E +02	6.03E +02	2.53E +02	6.70E +02	2.50E +02	2.21E +02	5.15E +02
F19	6.75E +02	2.40E +02	5.00E +02	2.50E +02	4.84E +02	2.51E +02	6.80E +02	2.18E +02	5.64E +02	2.59E +02	7.02E +02	2.38E +02	2.05E +02	4.51E +02
F20	6.42E +02	2.40E +02	4.40E +02	2.29E +02	4.20E +02	2.18E +02	6.40E +02	2.38E +02	5.63E +02	2.59E +02	7.20E +02	2.45E +02	1.87E +02	5.17E +02
F21	5.12E +02	2.03E +02	4.98E +02	9.05E +01	5.28E +02	1.10E +02	4.60E +02	1.38E +02	4.84E +02	1.68E +02	5.84E +02	1.90E +02	2.10E +02	3.70E +02
F22	7.51E +02	1.74E +01	7.15E +02	8.69E +01	7.43E +02	9.24E +01	7.61E +02	4.41E +00	7.48E +02	1.28E +01	7.48E +02	1.20E +01	1.65E +01	6.01E +02
F23	6.64E +02	1.22E +02	5.89E +02	9.12E +01	6.28E +02	1.40E +02	6.18E +02	7.92E +01	6.38E +02	1.24E +02	7.40E +02	1.24E +02	1.86E +02	5.60E +02
+					7	3	5	2	7	8	7	8	7	7
-					10	12	11	5	5	5	3	3	8	8
=					8	6	8	8	9	10	10	10	6	6

\*When seven algorithms obtain the global optimum, the intermediate results are reported at NFEs = 10000. “+”, “-”, and “=” denote that the performance of this algorithm is, respectively, better than, worse than, and similar to HDEMIR according to the Wilcoxon rank-sum test at  $\alpha = 0.05$ .



TABLE 3: Comparison of mean error and standard deviation between HDEM and HDE over 25 independent runs on twenty-one 10-dimensional test functions.

Function	HDE		HDEM	
F01*	0.00E+00	0.00E+00	=	0.00E+00
F02	2.27E-15	1.14E-14	=	0.00E+00
F03	0.00E+00	0.00E+00	=	0.00E+00
F05	1.79E-02	1.32E-02	-	1.53E-02
F06	0.00E+00	0.00E+00	=	0.00E+00
F07	2.02E-02	1.00E-02	=	1.93E-02
F08	2.00E+01	0.00E+00	=	2.00E+01
F09*	3.13E+00	8.03E-01	+	4.57E+00
F10	5.69E+00	1.51E+00	=	5.77E+00
F11	6.06E+00	1.12E+00	=	6.06E+00
F12	0.00E+00	0.00E+00	=	0.00E+00
F13	4.32E-01	1.01E-01	-	3.96E-01
F14	3.09E+00	2.85E-01	=	3.12E+00
F15	4.26E+01	3.74E+01	-	3.55E+01
F16	1.10E+02	9.80E+00	=	1.11E+02
F18	5.59E+02	2.38E+02	-	5.15E+02
F19	6.08E+02	2.24E+02	-	4.51E+02
F20	6.07E+02	2.27E+02	-	5.17E+02
F21	4.08E+02	1.29E+02	-	3.70E+02
F22	7.28E+02	1.25E+02	-	6.01E+02
F23	5.50E+02	3.37E+01	=	5.60E+02
+				1
-				8
=				12

\*When six algorithms obtain the global optimum, the intermediate results are reported at  $NFEs = 10000$ . “+,” “-,” and “=” denote that the performance of this algorithm is, respectively, better than, worse than, and similar to HDEM at  $\alpha = 0.05$ .

TABLE 4: Results obtained by the multiple-problem Wilcoxon test for CEC2005 at  $D = 10$ .

HDEM vs.	$R^+$	$R^-$	$p$ value	at $\alpha = 0.05$	at $\alpha = 0.1$
JADE	165.0	45.0	0.023907	+	+
CoDE	150.0	81.0	0.223788	=	=
jDE	190.0	41.0	0.009139	+	+
MPEDE	188.0	43.0	0.01117	+	+
SHADE	170.5	60.5	0.053725	=	+
LSHADE	139.5	70.5	0.191334	=	=
LSHADE- $\epsilon$	164.5	66.5	0.085341	=	+

5.3. *Sensitivity in Relation to the Parameter  $C_{\max}$* . In order to test the influence of the parameter  $C_{\max}$  used in the reinitialization scheme of HDEM, a set of experiments is performed here. The Friedman test results are shown in Table 6, where the values of  $C_{\max}$  are set to  $C_{\max} = \{3, 5, 7, 9, 11, 13, 15\}$ . All other parameters are kept unchanged as described in Section 5. In addition, all experiments are conducted for 25 independent runs for each function.

It can be seen from Table 6 that HDEM with  $C_{\max} = 9$  gets better average ranking value (3.3333) than other six cases.  $C_{\max} = 15$  is the second best choice to HDEM. Some interesting phenomena can be observed in Table 6. The large  $C_{\max}$  (i.e., 13 and 15) is better than the small values (i.e., 3 and 5). Generally speaking, the large  $C_{\max}$

value such as 9 or 15 is good to enhance the performance of HDEM.

5.4. *Performance Test of HDEM on CEC2014 Benchmark Functions*. In this section, we compare our approach with the same seven algorithms at 30 benchmark problems of CEC2014 as in [61]. All functions are conducted for 51 independent runs and limit each run to  $10000 * D$  max function evaluations. The average function error value smaller than  $10^{-8}$  is taken as zero. The parameters of HDEM are set as  $N_p = 30$ ,  $E_{\max} = 3.0$ ,  $S_{\max} = 2.0$ ,  $C_{\max} = 9$ ,  $CR \in N(0.8, 0.1)$ , and  $F \in N(0.5, 0.1)$ . For the other seven well-known algorithms, we use the same parameter settings as in their original papers. Table 7 shows mean error and standard deviation of the function error values. Table 8 shows the multiple-problem Wilcoxon test results. Figure 2 is the Friedman test results. Additionally, some representative convergence graphs of eight algorithms are shown in Figures 3 and 4.

From Table 7, we can see that HDEM performs better than all of the algorithms except LSHADE and LSHADE- $\epsilon$  at CEC2014 functions. Among the 30 test functions, HDEM is significantly better than JADE, CoDE, jDE, MPEDE, SHADE, LSHADE, and LSHADE- $\epsilon$  in 16, 13, 19, 23, 12, 9, and 7 functions, respectively, while the number of functions that JADE, CoDE, jDE, MPEDE, SHADE, LSHADE, and LSHADE- $\epsilon$  performs significantly better are 4, 10, 3, 2, 7, 15, and 11. Furthermore, HDEM obtains similar results with other seven algorithms in 10, 7, 8, 5, 11, 6, and 12 cases. From Table 8, we can find that there are the significant differences in three cases (i.e., HDEM vs. JADE, HDEM vs. jDE, and HDEM vs. MPEDE) at  $\alpha = 0.05$  and  $\alpha = 0.1$ . Moreover, the results shown in Figure 2 indicate that HDEM ranks the third best (3.9167) among the eight compared algorithms.

In addition, some interesting phenomena can be observed according to the experimental results in Table 7. For the unimodal functions (i.e., F1, F2, and F3), HDEM can produce similar results compared with the other algorithms. To simple multimodal functions and hybrid functions, HDEM has better or similar results than JADE, CoDE, jDE, and MPEDE. LSHADE and LSHADE- $\epsilon$  are very good at solving the simple multimodal and hybrid functions. To the composition functions (i.e., F23–F30), HDEM exhibits the best performance among the eight algorithms. The outstanding performance of HDEM on the composition functions can further demonstrate its capability to balance the exploration and exploitation. Some representative convergence graphs of the eight algorithms in some typical test functions have been shown in Figures 3 and 4. Figures 3 and 4 show that HDEM has a fast convergence speed at most of the functions, especially on the composition functions at CEC2014 benchmark.

5.5. *Performance Test of HDEM on GKLS Test Classes*. To further validate the performance of HDEM, the popular GKLS generator is used. This generator allows one to randomly generate classes of test problems with the same dimension, number of local minima, and difficulty, and each

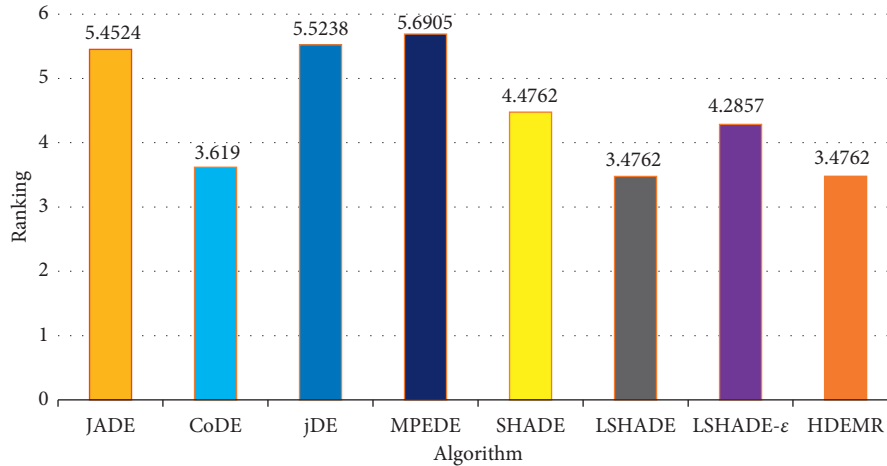


FIGURE 1: Average rankings of the eight algorithms by Friedman test for CEC2005 at  $D = 10$ .

TABLE 5: Average rankings of contraction criterion combinations by the Friedman test.

$D = 10$	
Parameters	Ranking
$E_{\max} = 1.0, S_{\max} = 1.0$	5.2857
$E_{\max} = 1.0, S_{\max} = 2.0$	4.6429
$E_{\max} = 1.0, S_{\max} = 3.0$	5.3095
$E_{\max} = 2.0, S_{\max} = 1.0$	4.6905
$E_{\max} = 2.0, S_{\max} = 2.0$	4.8333
$E_{\max} = 2.0, S_{\max} = 3.0$	5.1905
$E_{\max} = 3.0, S_{\max} = 1.0$	4.7143
$E_{\max} = 3.0, S_{\max} = 2.0$	4.4286
$E_{\max} = 3.0, S_{\max} = 3.0$	5.9048

TABLE 6: Average rankings of  $C_{\max}$  by the Friedman test.

$D = 10$	
Parameters	Ranking
$C_{\max} = 3$	5.3333
$C_{\max} = 5$	4.2619
$C_{\max} = 7$	4.381
$C_{\max} = 9$	3.3333
$C_{\max} = 11$	3.6429
$C_{\max} = 13$	3.5476
$C_{\max} = 15$	3.5

class contains 100 problems. Here, 1000 test functions are used in total, which consist of 10 classes of problems with dimension  $D=2, 3, 4, 5$ , and 10. The control parameters of GKLS test classes used in the experiments are presented in Table 9. Control parameters of the GKLS generator include (1) the radius of the convergence region  $\rho$ , (2) distance between the paraboloid vertex and the global minimizer  $r$ , and (3) the tolerance  $\Delta$  (for details, see Sergeyev and Kvasov [14]). The parameters of HDEMUR are set as  $N_p = 30$ ,  $E_{\max} = 3.0$ ,  $S_{\max} = 2.0$ ,  $C_{\max} = 9$ ,  $CR \in N(0.8, 0.1)$ , and  $F \in N(0.5, 0.1)$ . In order to make this comparison more reliable, parameters of DIRECT are used as recommended in its original papers. For each of the

generated problems, HDEMUR is conducted separately 100 independent runs and DIRECT is executed one run. The maximal number of trails (or function evaluations) is  $\text{Max\_NFEs} = 10^6$ .

To effectively analyze the results, Operational Characteristics and Operational Zones, as similarly done in [13], are used in the experiments.

Because of the huge amount of data, only average results are included in Table 10.  $m(i)$  means that the algorithm does not solve a global optimization problem  $i$  times in 100 runs  $\times$  100 problems (i.e., in 10,000 runs for HDEMUR and in 100 runs for DIRECT). Figures 5 and 6 show operational characteristics for DIRECT and operational zones for HDEMUR on the different dimensions.

Figure 5 shows the results on the 2-, 3-, 4-dimensional simple and hard classes for HDEMUR and DIRECT. It can be seen from Figure 5(a) that operational characteristics of DIRECT is higher than the upper boundary of the zone of HDEMUR and, therefore, on this class DIRECT has a better performance. Figure 5(b) shows that the average performance with the zone (the red line in the zone) of HDEMUR is higher than characteristics of DIRECT. It means that HDEMUR outperforms DIRECT. Figures 5(c) and 5(d) show that the average performance of HDEMUR is better than DIRECT on both the 3-dimensional simple and hard classes.

Figures 5(e) and 5(f) depict that on the 4-dimensional simple and hard classes, if the number of function evaluations is less than 30000, the average performance of HDEMUR is better than DIRECT. If the number of function evaluations is higher than 30000, DIRECT behaves better since its characteristic is higher than the average line of the HDEMUR zone. We can also see the similar results on the 5-dimensional simple and hard classes in Figures 6(a) and 6(b).

Figures 6(c) and 6(d) illustrate that the average performance of HDEMUR is better than DIRECT on the 10-dimensional simple and hard classes. It is suitable to solve the ILAIS problem because the dimension of weight parameter identification of ILAIS is approximately 10.

TABLE 7: Comparison of mean error and standard deviation between HDEMIR and other seven DE variants on CEC2014 at  $D = 10$ .

Function	JADE	CoDE	jDE	MPEDE	SHADE	LSHADE	LSHADE-ε	HDEMIR
F01	0.00E+00	0.00E+00	0.00E+00	1.85E-09	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F02	2.09E-03	1.49E-02	0.00E+00	2.12E-10	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F03	8.74E-03	2.79E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F04	2.75E+01	1.40E+01	1.19E+01	1.75E+01	3.07E+01	1.13E+01	3.48E+01	0.00E+00
F05	1.75E+01	4.87E+01	1.89E+01	1.67E+01	6.97E+01	8.00E+01	1.17E+01	1.96E+01
F06	9.86E-02	1.49E-01	1.75E-02	2.00E-02	0.00E+00	0.00E+00	1.75E-02	1.22E-01
F07	1.29E-02	8.51E-03	1.82E-02	1.81E-02	4.51E-03	5.90E-03	5.85E-04	9.22E-03
F08	0.00E+00	0.00E+00	0.00E+00	5.88E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F09	3.41E+00	8.69E+00	5.66E+00	6.52E-05	3.07E+00	8.13E+00	1.86E+00	2.97E+00
F10	4.90E-03	1.70E-02	0.00E+00	6.77E-01	6.12E-03	1.88E-02	1.10E-02	1.36E-01
F11	9.15E+01	5.94E+01	2.69E+02	2.83E+02	7.79E+01	6.22E+01	1.71E+01	1.08E+02
F12	2.47E-01	5.38E-02	3.80E-01	3.99E-01	1.34E-01	2.80E-02	6.94E-02	1.40E-02
F13	8.61E-02	1.68E-02	1.42E-01	1.27E-01	7.29E-02	1.43E-02	4.74E-02	1.81E-01
F14	9.73E-02	3.12E-02	1.60E-01	1.20E-01	9.80E-02	2.96E-02	8.56E-02	1.04E-03
F15	5.56E-01	9.46E-01	1.02E+00	9.40E-01	4.84E-01	7.81E-01	3.73E-01	4.40E-01
F16	1.64E+00	2.91E+00	2.09E+00	2.08E+00	1.52E+00	2.83E+00	1.01E+00	1.71E+00
F17	5.18E+01	3.30E+02	9.89E+01	2.92E+01	4.62E+01	1.42E+01	3.12E+01	6.80E+01
F18	2.86E-01	4.43E-01	1.42E+00	1.69E+00	1.87E+00	1.77E+00	2.85E+00	8.49E+00
F19	2.65E-01	7.72E-02	3.56E-01	4.50E-01	2.64E-01	2.63E-01	2.85E-01	1.33E-01
F20	3.22E-01	1.19E-01	2.31E-01	8.65E-01	2.43E-01	1.13E-01	2.44E-01	4.03E-01
F21	1.60E+00	4.70E+00	3.77E+00	3.15E+00	3.62E+00	2.60E+00	2.64E+00	2.63E+00
F22	1.82E-01	6.96E-02	1.95E-01	4.99E+00	3.10E-01	1.14E-01	1.67E-01	2.16E-01

TABLE 7: Continued.

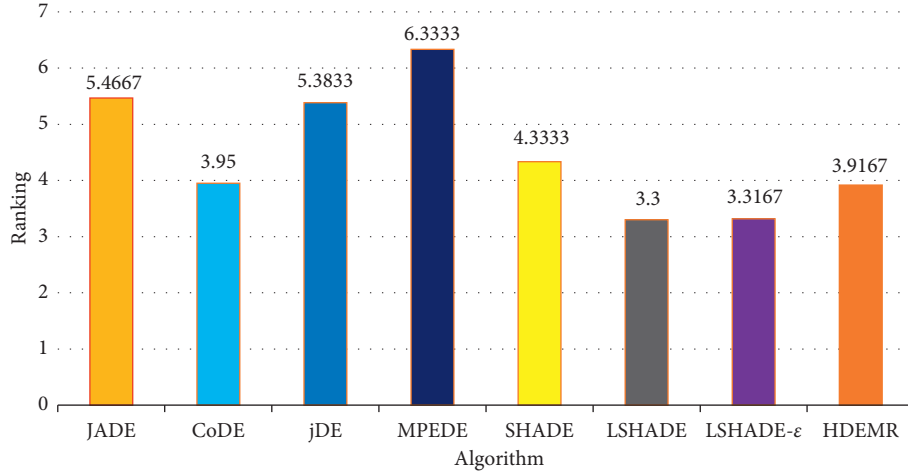
Function	JADE	CoDE	jDE	MPEDE	SHADE	LSHADE	LSHADE- $\epsilon$	HDEMR
F23	3.29E +02 2.87E -13	3.29E +02 2.87E -13	3.29E +02 2.87E -13	3.29E +02 2.87E -13	3.29E +02 2.87E -13	3.29E +02 2.87E -13	2.00E +02 0.00E	2.00E +02 0.00E
F24	1.10E +02 1.32E +00	1.12E +02 3.82E +00	1.13E +02 1.73E +00	1.11E +02 2.81E +00	1.09E +02 1.86E +00	1.08E +02 1.90E +00	1.07E +02 2.23E +00	1.08E +02 3.34E +00
F25	1.30E +02 2.99E +01	1.35E +02 3.50E +01	1.36E +02 3.50E +01	1.24E +02 1.05E +01	1.38E +02 4.14E +01	1.41E +02 4.51E +01	1.38E +02 3.78E +01	1.25E +02 1.64E +01
F26	1.00E +02 1.69E -02	1.00E +02 2.69E -02	1.00E +02 2.81E -02	1.00E +02 2.45E -02	1.00E +02 1.54E -02	1.00E +02 1.54E -02	1.00E +02 1.61E -02	1.00E +02 1.76E -02
F27	1.09E +02 1.60E +02	6.25E +01 1.42E +02	7.43E +01 1.41E +02	2.28E +00 4.56E -01	1.50E +02 1.68E +02	5.41E +01 1.25E +02	6.31E +01 9.51E +01	2.10E +01 7.13E +01
F28	4.02E +02 4.87E +01	3.66E +02 2.52E +01	3.66E +02 2.30E +01	3.89E +02 4.52E +01	3.91E +02 4.36E +01	3.83E +02 3.84E +01	2.00E +02 2.17E +01	2.03E +02 2.24E +01
F29	2.47E +02 4.14E +01	2.20E +02 1.26E +01	2.21E +02 6.46E +00	2.22E +02 7.75E -02	2.22E +02 6.35E -01	2.22E +02 5.30E -01	2.00E +02 0.00E +00	1.99E +02 1.86E +01
F30	4.94E +02 3.64E +01	4.64E +02 6.88E +00	4.65E +02 1.12E +01	4.80E +02 1.52E +01	4.75E +02 2.21E +01	4.66E +02 1.23E +01	3.57E +02 1.39E +02	2.02E +02 1.75E +01
+	4	10	3	2	7	15	11	11
-	16	13	19	23	12	9	7	7
=	10	7	8	5	11	6	12	12

“+”, “-”, and “=” denote that the performance of this algorithm is, respectively, better than, worse than, and similar to HDEMR according to the Wilcoxon rank-sum test at  $\alpha = 0.05$ .



TABLE 8: Results obtained by the multiple-problem Wilcoxon test for CEC2014 at  $D = 10$ .

HDEMR vs.	$R^+$	$R^-$	$p$ value	at $\alpha = 0.05$	at $\alpha = 0.1$
JADE	319.5	115.5	0.026666	+	+
CoDE	271.5	193.5	0.416534	=	=
jDE	375.0	60.0	0.000634	+	+
MPEDA	395.5	69.5	0.000771	+	+
SHADE	269.0	166.0	0.259551	=	=
LSHADE	219.5	245.5	1	=	=
LSHADE- $\epsilon$	212.0	253.0	1	=	=

FIGURE 2: Average rankings of the eight algorithms by Friedman test for CEC2014 at  $D = 10$ .

In addition, some interesting phenomena can be observed according to the experimental results in Figures 5 and 6. The red average lines indicate that HDEMR has a fast convergence speed at most of the classes except for 2-dimensional simple class, especially on the hard classes. However, the number of solved problems by DIRECT is more than HDEMR on 4-dimensional and 5-dimensional hard classes (see Figures 5(f) and 6(b)).

At last, one can see that the performance of HDEMR is not very stable on 4-, 5-, 10-dimensional classes. The reason is that HDEMR gets trapped into local minima in several cases of 100 runs and is not able to exit from their attraction regions producing the operational zones with the wider interval.

From Table 10, the number of solved problems of DIRECT is more than HDEMR at GKLS test classes. For example, on the 10-dimensional hard class, DIRECT does not solve 2 problems in 100 runs (98% of success), whereas HDEMR does not solve 599 problems in 10000 runs (94.01% of success). On the other hand, the average number of trails (or function evaluations) for DIRECT is 21774.72, while for HDEMR is 60191.

## 6. HDEMR for Weight Parameter Optimization of ILAIS

According to the results of CEC2005, CEC2014, and GKLS test functions, HDEMR achieves better or similar results among nine algorithms. Hence, it is a good alternative for

weight parameter optimization of ILAIS. In this section, the experiments aim at using HDEMR to solve the weight parameter identification of ILAIS.

Questionnaire survey and individual evaluation are conducted at China University of Geosciences. There are 1000 questionnaires collected from the students, and 300 questionnaires from the teachers.

The parameters of HDEMR are set as  $N_p = 100$ ,  $E_{\max} = 3.0$ ,  $S_{\max} = 2.0$ ,  $C_{\max} = 9$ ,  $CR \in N(0.8, 0.1)$ ,  $F \in N(0.5, 0.1)$ , and  $\text{Max\_NFES} = 50000$ . HDEMR is conducted for 50 independent runs. The optimization objective function is defined in Equation (2) and  $t = 0.6$ ,  $s = 0.4$ . To effectively analyze the results, mean value and standard deviation are also used in the experiments.

Table 11 shows the weight results of the information literacy assessment index at CUG by HDEMR. It can be seen that the weights of first-level indexes such as Information Consciousness, Information Knowledge, Information Ability, and Information Morality are 0.1958, 0.2085, 0.3115, and 0.2958, respectively. Information Ability and Information Morality are more important than Information Consciousness and Information Knowledge according to the questionnaire survey. It means that Information Ability and Information Morality have a greater impact to assess the information literacy of the college students.

Table 11 also shows the results of the thirteen second-level index weights by HDEMR. We can see that the weight of L11 is 0.4031 and L12 and L13 are approximately

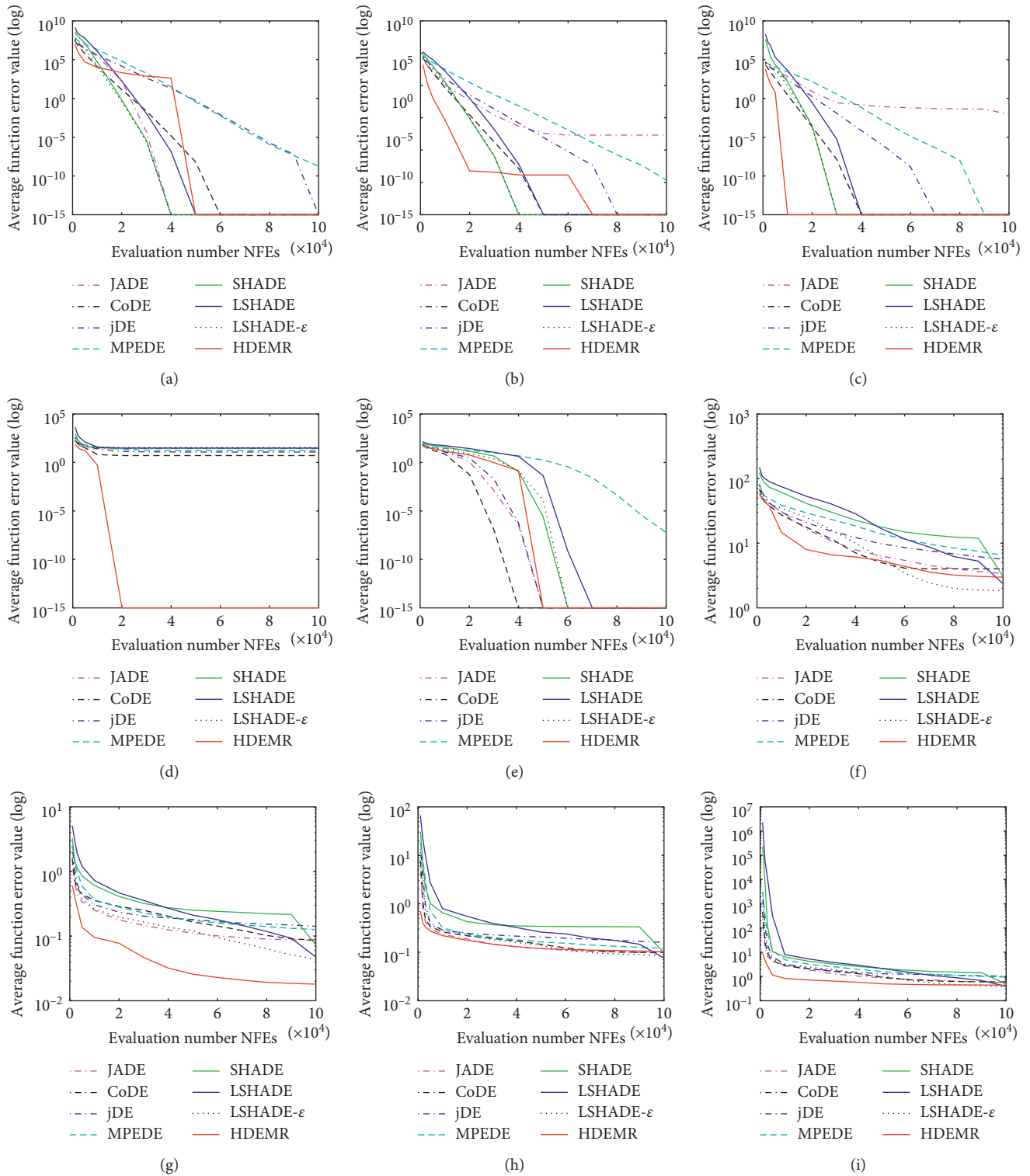


FIGURE 3: The convergence graphs on different types of CEC2014 benchmark functions. (a) The convergence graph on the unimodal function f1. (b) The convergence graph on the unimodal function f2. (c) The convergence graph on the unimodal function f3. (d) The convergence graph on the simple multimodal function f4. (e) The convergence graph on the simple multimodal function f8. (f) The convergence graph on the simple multimodal function f9. (g) The convergence graph on the simple multimodal function f13. (h) The convergence graph on the simple multimodal function f14. (i) The convergence graph on the simple multimodal function f15.

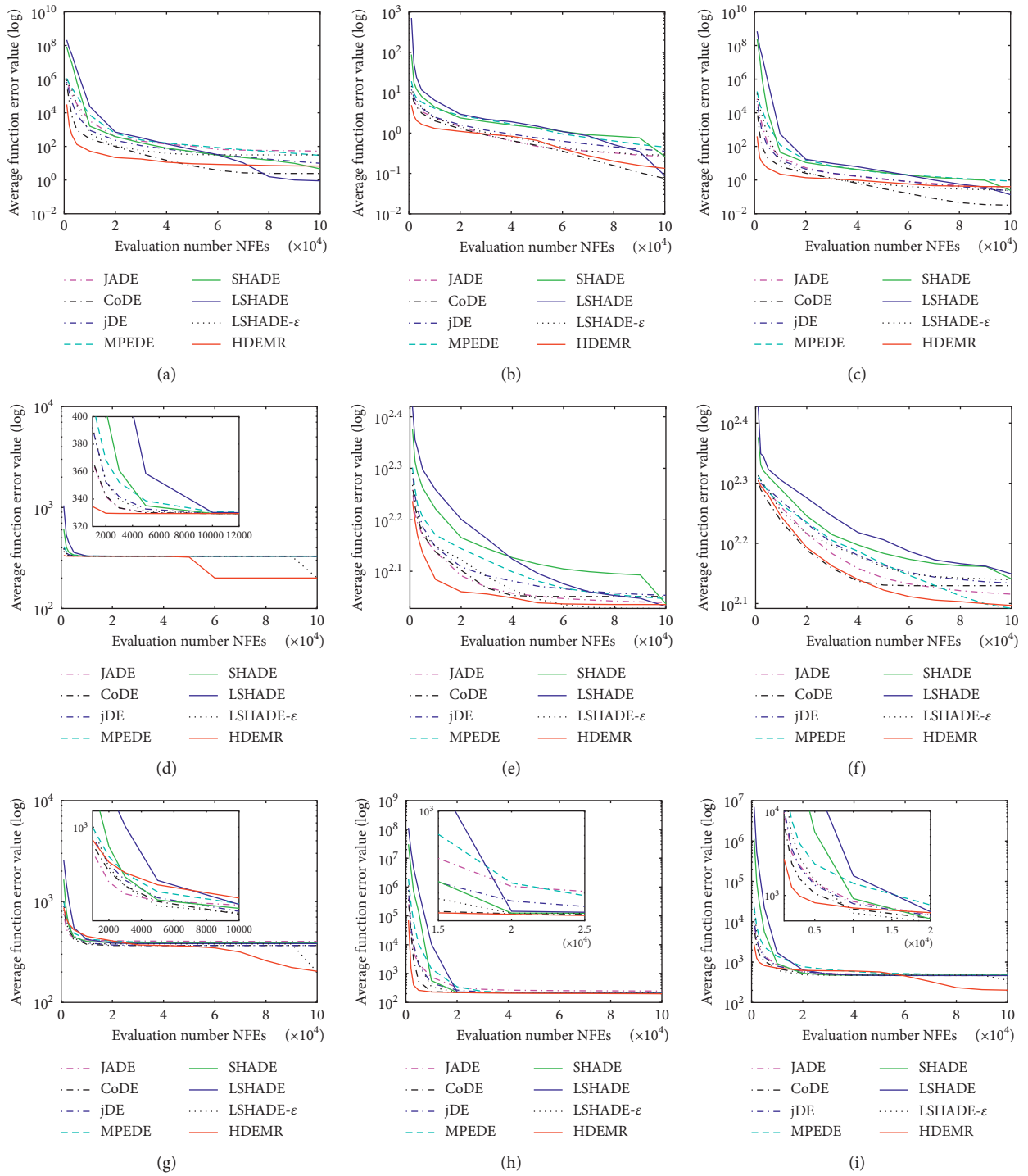


FIGURE 4: The convergence graphs on different types of CEC2014 benchmark functions. (a) The convergence graph on the hybrid function f17. (b) The convergence graph on the hybrid function f19. (c) The convergence graph on the hybrid function f20. (d) The convergence graph on the composition function f23. (e) The convergence graph on the composition function f24. (f) The convergence graph on the composition function f25. (g) The convergence graph on the composition function f28. (h) The convergence graph on the composition function f29. (i) The convergence graph on the composition function f30.

0.3 which means that L11 (the recognition of the value of information and the objective evaluation of the role of information) is an important weight to Information Consciousness. To Information Knowledge, the results present

that the weights of L22 and L23 are both approximately 0.4 which means that the ability of understanding the value of various sources of information and communication process and the abilities of identifying information needs,

TABLE 9: GKLS test classes.

D	Hardness	$r$	$\rho$	$\Delta$
2	Simple	0.9	0.2	$10^{-4}$
2	Hard	0.9	0.1	$10^{-4}$
3	Simple	0.66	0.2	$10^{-6}$
3	Hard	0.9	0.2	$10^{-6}$
4	Simple	0.66	0.2	$10^{-6}$
4	Hard	0.9	0.2	$10^{-6}$
5	Simple	0.66	0.3	$10^{-7}$
5	Hard	0.66	0.2	$10^{-7}$
10	Simple	0.66	0.3	$10^{-7}$
10	Hard	0.66	0.2	$10^{-7}$

For each test class, the dimension of test class (D), the radius of the convergence region  $\rho$ , distance between the paraboloid vertex and the global minimizer ( $r$ ), and the tolerance  $\Delta$  are given.

TABLE 10: Results of the experiments.

D	Class	HDEMR (10000 runs for the algorithm and class)	DIRECT (100 runs for the algorithm and class)
2	Simple	>1028.3(6)	197.26(0)
2	Hard	>17989(168)	1054.58(0)
3	Simple	>7569.9(67)	960.78(0)
3	Hard	>28182(268)	3651.54(0)
4	Simple	>80799(797)	>27608.34(2)
4	Hard	>351150(3505)	>112790.53(6)
5	Simple	>81913(811)	>15858.45(1)
5	Hard	>444170(4434)	>226665.45(17)
10	Simple	>71482(712)	>22067.58(2)
10	Hard	>60191(599)	>21774.72(2)

For each test class the average number of trails (or function evaluations) required to solve all 100 problems is presented for DIRECT algorithm. For HDEMR, the average number of trails (or function evaluations) required to solve each problem on 100 runs has been calculated, and the average of these 100 values is presented. The record “>  $m(i)$ ” means that the algorithm does not solve a global optimization problem  $i$  times in 100 runs  $\times$  100 problems (i.e., in 10,000 runs for HDEMR and in 100 runs for DIRECT). In this case, the maximal number of trails (or function evaluations) set to  $10^6$  is used to calculate the average number of trails (or function evaluations)  $m$ .

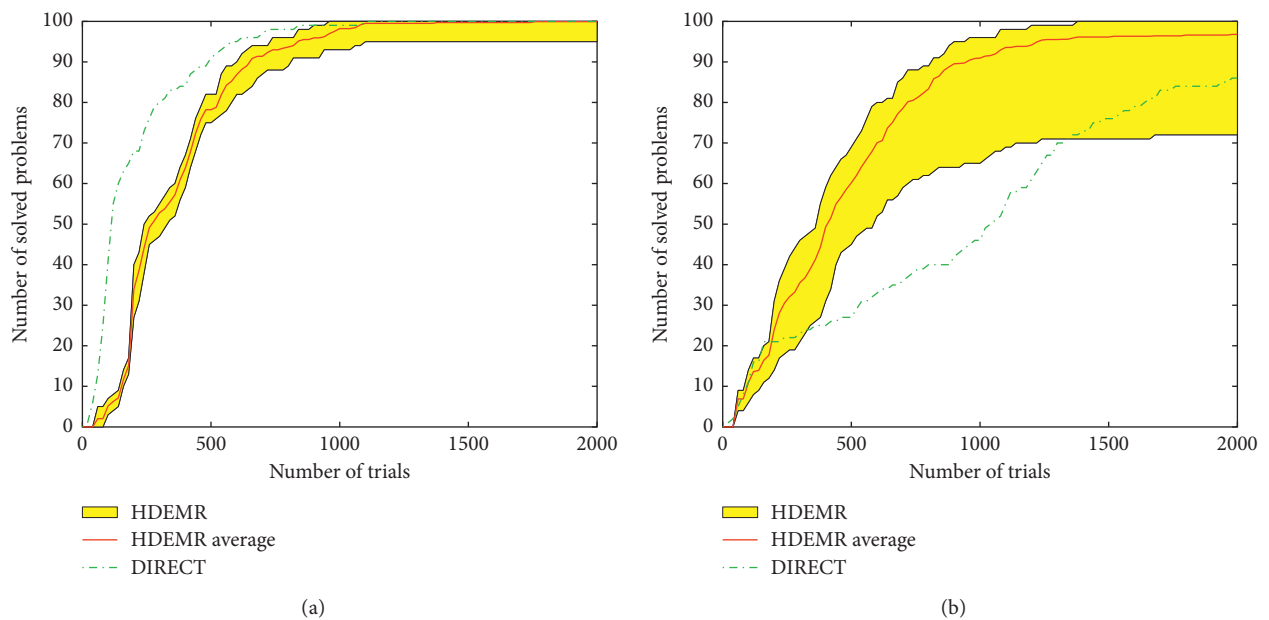


FIGURE 5: Continued.



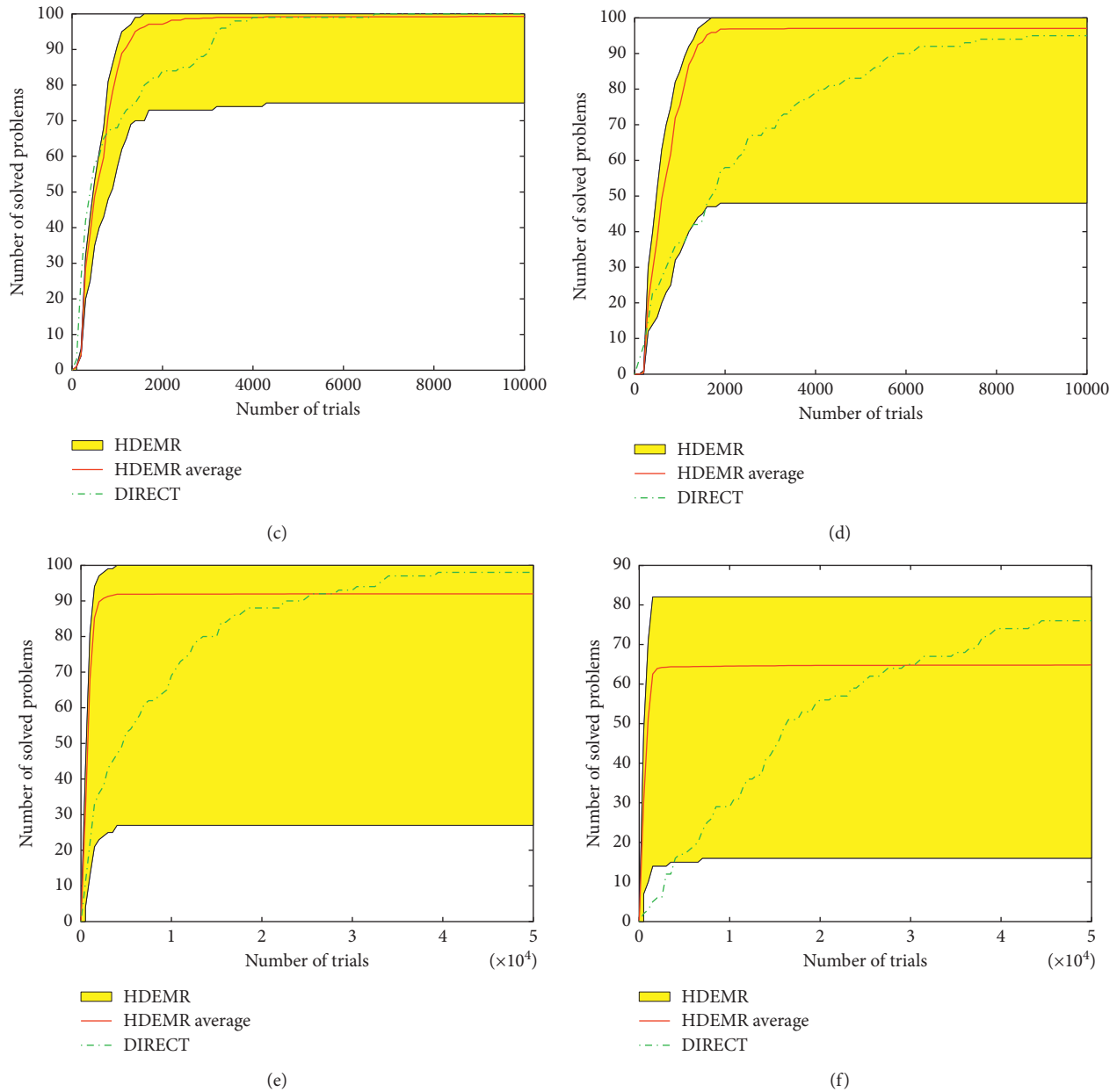


FIGURE 5: Operational characteristics built on 2-, 3-, 4-dimensional classes of GKLS test functions for DIRECT and operational zones for HDEM R. (a) Operational characteristics for DIRECT and the operational zone for HDEM R for the simple 2-dimensional class. (b) The same as (a) for the hard class. (c) Operational characteristics for DIRECT and the operational zone for HDEM R for the simple 3-dimensional class. (d) The same as (c) for the hard class. (e) Operational characteristics for DIRECT and the operational zone for HDEM R for the simple 4-dimensional class. (f) The same as (e) for the hard class.

mastering basic information science knowledge, clarifying your own information needs, and describing keywords and terminology are very important to assess the collage students' Information Knowledge. Furthermore, it is also seen that the weights of L32, L33, and L34 are close to 0.3. It means that these three indexes have the same effect on Information Ability. At last, to Information Morality, the weight of L41 is close to 0.5. This means that the ability to master information law has the significant impact on

evaluating the college students' Information Morality. The second important factor of L42 is 0.2983 to Information Morality.

Finally, according to the results in Table 11, it can be seen that all the standard deviations are within  $10^{-3}$ . It can draw the conclusion that HDEM R is stable to optimize the weights of ILAIS. The optimal weights are able to objectively determine the evaluation of the college students' information literacy at CUG.

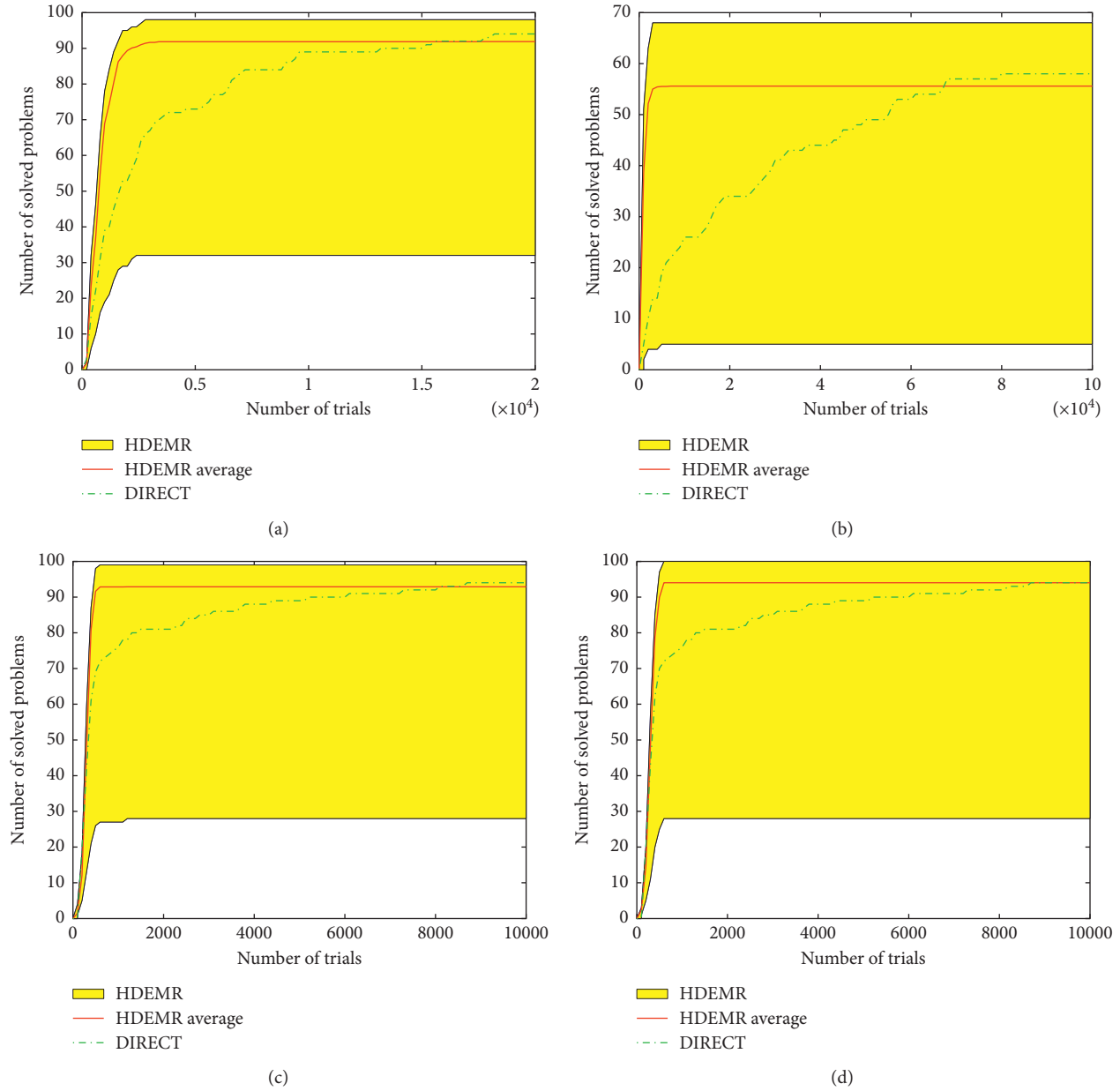


FIGURE 6: Operational characteristics built on 5- and 10-dimensional classes of GKLS test functions for DIRECT and operational zones for HDEM. (a) Operational characteristics for DIRECT and the operational zone for HDEM for the simple 5-dimensional class. (b) The same as (a) for the hard class. (c) Operational characteristics for DIRECT and the operational zone for HDEM for the simple 10-dimensional class. (d) The same as (c) for the hard class.

## 7. Conclusion

Apart from developing a new weight parameter optimization model of ILAIS, in this study, an efficient HDEM method is proposed to solve the weight parameter identification problem of ILAIS at CUG with greater speed and accuracy. The HDEM method uses an improved contraction criterion to decide when the local search starts. A model-based reinitialization strategy is also proposed to improve the diversity of the algorithm and the performance of global search. These improvements are simple yet efficient

and make HDEM a powerful alternative for the real-world applications. The effectiveness and efficiency of HDEM are validated by CEC2005, CEC2014 benchmark functions, and GKLS test classes. The superiority of HDEM is also verified after comparing it with seven well-known DE variants and the widely used deterministic algorithm DIRECT. At last, HDEM is successfully used to optimize the weight parameters of ILAIS at China University of Geosciences (CUG).

In our future work, HDEM will be tested on other real-world applications problems. Moreover, we believe that

TABLE 11: The weight results of information literacy assessment index at CUG by HDEMUR.

First-level index	Mean Value	Standard Deviation	Second-level index	Mean Value	Standard Deviation
L1	0.1958	0.0029	L11	0.4031	0.0008
			L12	0.2988	0.0006
			L13	0.2984	0.0006
L2	0.2085	0.0031	L21	0.2032	0.0006
			L22	0.3981	0.0011
			L23	0.4000	0.0012
L3	0.3115	0.0046	L31	0.0989	0.0014
			L32	0.3042	0.0043
			L33	0.3081	0.0043
			L34	0.3001	0.0042
L4	0.2958	0.0044	L41	0.4993	0.0018
			L42	0.2983	0.0011
			L43	0.2045	0.0007

some other local search algorithms and parameter adaptation strategies can also be used in HDEMUR.

## 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 there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

The authors would like to thank Dr. Jingqiao Zhang, Dr. Yong Wang, Dr. Janez Brest, Dr. Guohua Wu, Dr. Ryoji Tanabe, and Dr. PN Suganthan for providing the source codes of JADE, CoDE, jDE, MPEDE, SHADE and LSHADE, and LSHADE- $\epsilon$ , respectively. This work was supported in part by Education Science Planning Project of Hubei Province (2017GA011), in part by the Open Fund Project of the Center for Development and Innovation of College Students—the key research base of Humanities and Social Sciences in Hubei (DXS20180012), and in part by the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) (CUG160207).

## References

- [1] S. Webber and B. Johnston, “Conceptions of information literacy: new perspectives and implications,” *Journal of Information Science*, vol. 26, no. 6, pp. 381–397, 2000.
- [2] M. Cooney and L. Hiris, “Integrating information literacy and its assessment into a graduate business course: a collaborative framework,” *Research Strategies*, vol. 19, no. 3–4, pp. 213–232, 2003.
- [3] S. Valerie, T. Denise, M. Sue, and C. Yvonne, “Information literacy assessment by portfolio: a case study,” *Reference Services Review*, vol. 35, no. 1, pp. 41–70, 2007.
- [4] M. Pinto, “Design of the IL-HUMASS survey on information literacy in higher education: a self-assessment approach,” *Journal of Information Science*, vol. 36, no. 1, pp. 86–103, 2009.
- [5] A. Naimpally, H. Ramachandran, and C. Smith, “7—Information literacy and assessment,” in *Lifelong Learning for Engineers and Scientists in the Information Age*, pp. 49–62, Elsevier, New York City, NY, USA, 2012.
- [6] M. Kousar and K. Mahmood, “Information literacy skills assessment of undergraduate Engineering Students,” in *Worldwide Commonalities and Challenges in Information Literacy Research and Practice: European Conference on Information Literacy, ECIL 2013 Istanbul, Turkey*, pp. 471–477, Springer, Berlin, Germany, 2013.
- [7] M. Krakowska, “Information literacy skills assessment of LIS students: a case study at the jagiellonian university,” in *Worldwide Commonalities and Challenges in Information Literacy Research and Practice: European Conference on Information Literacy, ECIL 2013 Istanbul, Turkey*, pp. 617–624, Springer, Berlin, Germany, 2013.
- [8] M. Pavlovski and I. Dunder, “Information literacy assessment at the University of Zagreb: an undergraduate students’ perspective,” in *Proceedings of 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics*, pp. 695–699, MIPRO, Opatija, Croatia, May 2015.
- [9] K. Kultawanich, P. Koraneekij, and J. Na-Songkhla, “Development and validation of the information literacy assessment in connectivism learning environment for undergraduate students,” *Procedia—Social and Behavioral Sciences*, vol. 174, pp. 1386–1390, 2015.
- [10] K. A. Douglas, T. Fernandez, S. Purzer, M. Fosmire, and A. Van Epps, “A self-assessment instrument to assess engineering students’ self-directedness in information literacy,” in *Proceedings of 2015 IEEE Frontiers in Education Conference (FIE)*, pp. 1396–1403, El Paso, TX, USA, October 2015.
- [11] T. Kavšek, C. Peklaj, and U. Žugelj, “Information literacy training evaluation: the case of first year psychology students,” *Journal of Academic Librarianship*, vol. 42, no. 4, pp. 293–299, 2016.
- [12] S. Lanning and J. Mallek, “Factors influencing information literacy competency of college students,” *Journal of Academic Librarianship*, vol. 43, no. 5, pp. 443–450, 2017.
- [13] Y. D. Sergeyev, D. E. Kvasov, and M. S. Mukhametzhonov, “On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget,” *Scientific reports*, vol. 8, no. 1, p. 453, 2018.
- [14] Y. D. Sergeyev and D. E. Kvasov, “Global search based on efficient diagonal partitions and a set of Lipschitz constants,” *SIAM Journal on Optimization*, vol. 16, no. 3, pp. 910–937, 2006.
- [15] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, “Lipschitzian optimization without the Lipschitz constant,” *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.
- [16] D. R. Jones, “Direct global optimization algorithm direct global optimization algorithm,” in *Encyclopedia of Optimization*, C. A. Floudas and P. M. Pardalos, Eds., pp. 431–440, Springer, Boston, MA, USA, 2001.
- [17] G. Liuzzi, S. Lucidi, and V. Piccialli, “A partition-based global optimization algorithm,” *Journal of Global Optimization*, vol. 48, no. 1, pp. 113–128, 2010.
- [18] R. Paulavičius, Y. D. Sergeyev, D. E. Kvasov, and J. Žilinskas, “Globally-biased DISIMPL algorithm for expensive global optimization,” *Journal of Global Optimization*, vol. 59, no. 2–3, pp. 545–567, 2014.
- [19] A. Gimbutas and A. Žilinskas, “An algorithm of simplicial Lipschitz optimization with the bi-criteria selection of

- simplices for the bi-section,” *Journal of Global Optimization*, vol. 71, no. 1, pp. 115–127, 2018.
- [20] 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.
- [21] M. Zuo, G. Dai, L. Peng, M. Wang, and J. Xiong, “Multiple gravity assist spacecraft trajectories design based on BFS and EP\_DE algorithm,” *International Journal of Aerospace Engineering*, vol. 2016, Article ID 3416046, 13 pages, 2016.
- [22] L. Peng, G. Dai, M. Wang, H. Hu, Y. Chang, and F. Chen, “Self-adaptive uniform differential evolution for optimizing the initial integral point of the Earth-Moon low-energy transfer,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 225, no. 11, pp. 1263–1276, 2011.
- [23] M. Vasile, E. Minisci, and M. Locatelli, “An inflationary differential evolution algorithm for space trajectory optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 2, pp. 267–281, 2011.
- [24] M. Wang, Z. Song, G. Dai, L. Peng, and C. Zheng, “Asteroids exploration trajectory optimal design with differential evolution based on mixed coding,” *International Journal of Distributed Sensor Networks*, vol. 11, no. 7, pp. 1–8, 2015.
- [25] R. Chai, A. L. Savvaris, and A. Tsourdos, “Violation learning differential evolution-based hp-adaptive pseudospectral method for trajectory optimization of space maneuver vehicle,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 4, pp. 2031–2044, 2017.
- [26] A. Glotić and A. Zamuda, “Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution,” *Applied Energy*, vol. 141, pp. 42–56, 2015.
- [27] A. Zamuda and J. D. Hernández Sosa, “Differential evolution and underwater glider path planning applied to the short-term opportunistic sampling of dynamic mesoscale ocean structures,” *Applied Soft Computing*, vol. 24, pp. 95–108, 2014.
- [28] L. Mingyong and C. Erbao, “An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows,” *Engineering Applications of Artificial Intelligence*, vol. 23, no. 2, pp. 188–195, 2010.
- [29] E. Pu, F. Wang, Z. Yang, J. Wang, Z. Li, and X. Huang, “Hybrid differential evolution optimization for the vehicle routing problem with time windows and driver-specific times,” *Wireless Personal Communications*, vol. 95, no. 3, pp. 2345–2357, 2017.
- [30] J. Zhang, S. Lin, H. Liu, Y. Chen, M. Zhu, and Y. Xu, “A small-population based parallel differential evolution algorithm for short-term hydrothermal scheduling problem considering power flow constraints,” *Energy*, vol. 123, pp. 538–554, 2017.
- [31] M. Azizi and H. Sajedi, “Satellite broadcast scheduling based on a boosted binary differential evolution,” *New Generation Computing*, vol. 35, no. 3, pp. 225–251, 2017.
- [32] G. Li, C. Chen, F. Yao, R. He, and Y. Chen, “Hybrid differential evolution optimisation for earth observation satellite scheduling with time-dependent earliness-tardiness penalties,” *Mathematical Problems in Engineering*, vol. 2017, Article ID 2490620, 10 pages, 2017.
- [33] A. K. Bhandari, A. Kumar, S. Chaudhary, and G. K. Singh, “A new beta differential evolution algorithm for edge preserved colored satellite image enhancement,” *Multidimensional Systems and Signal Processing*, vol. 28, no. 2, pp. 495–527, 2017.
- [34] S. Das and P. N. Suganthan, “Differential evolution: a survey of the state-of-the-art,” *evolutionary computation*, *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [35] 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.
- [36] N. H. Awad, M. Z. Ali, P. N. Suganthan, and E. Jaser, “A decremental stochastic fractal differential evolution for global numerical optimization,” *Information Sciences*, vol. 372, pp. 470–491, 2016.
- [37] M. Z. Ali, N. H. Awad, P. N. Suganthan, and R. G. Reynolds, “A modified cultural algorithm with a balanced performance for the differential evolution frameworks,” *Knowledge-Based Systems*, vol. 111, pp. 73–86, 2016.
- [38] N. H. Awad, M. Z. Ali, P. N. Suganthan, and R. G. Reynolds, “CADE: a hybridization of cultural algorithm and differential evolution for numerical optimization,” *Information Sciences*, vol. 378, pp. 215–241, 2017.
- [39] Y. Cai, J. Liao, T. Wang, Y. Chen, and H. Tian, “Social learning differential evolution,” *Information Sciences*, vol. 433–434, pp. 464–509, 2016.
- [40] S. S. Jadon, R. Tiwari, H. Sharma, and J. C. Bansal, “Hybrid artificial bee colony algorithm with differential evolution,” *Applied Soft Computing*, vol. 58, pp. 11–24, 2017.
- [41] H. Peng, Z. Wu, and C. Deng, “Enhancing differential evolution with commensal learning and uniform local search,” *Chinese Journal of Electronics*, vol. 26, no. 4, pp. 725–733, 2017.
- [42] F. Zhao, Z. Shao, J. Wang, and C. Zhang, “A hybrid optimization algorithm based on chaotic differential evolution and estimation of distribution,” *Computational & Applied Mathematics*, vol. 36, no. 1, pp. 433–458, 2017.
- [43] L. Peng, Y. Zhang, G. Dai, and M. Wang, “Memetic differential evolution with an improved contraction criterion,” *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 1395025, 12 pages, 2017.
- [44] R. Tanabe and A. S. Fukunaga, “Improving the search performance of SHADE using linear population size reduction,” in *Proceedings of Evolutionary Computation (CEC)*, pp. 1658–1665, 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, July 2014.
- [45] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M. Jambi, “LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems,” in *Proceedings of Evolutionary Computation (CEC)*, 2017 IEEE Congress on, pp. 145–152, Donostia-San Sebastian, Spain, June 2017.
- [46] N. R. Sabar, J. Abawajy, and J. Yearwood, “Heterogeneous cooperative co-evolution memetic differential evolution algorithm for big data optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 315–327, 2017.
- [47] M. Locatelli and M. Vasile, “(Non)convergence results for the differential evolution method,” *Optimization Letters*, vol. 9, no. 3, pp. 413–425, 2015.
- [48] A. K. Qin and P. N. Suganthan, “Self-adaptive differential evolution algorithm for numerical optimization,” in *Proceedings of Evolutionary Computation (CEC)*, pp. 2958–2965, 2005 IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, UK, September 2005.
- [49] J. Sun, J. Garibaldi, N. Krasnogor, and Q. Zhang, “An intelligent multi-restart memetic algorithm for box constrained global optimisation,” *Evolutionary Computation*, vol. 21, no. 1, pp. 107–147, 2013.
- [50] D. Simon, M. G. H. Omran, and M. Clerc, “Linearized biogeography-based optimization with re-initialization and local search,” *Information Sciences*, vol. 267, pp. 140–157, 2014.



- [51] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Mathematics of Computation*, vol. 24, no. 111, pp. 647–656, 1970.
- [52] N. N. Schraudolph, J. Yu, and S. Günter, "A Stochastic quasi-Newton method for online convex optimization," in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, pp. 436–443, San Juan, Puerto Rico, June 2007.
- [53] A. Zamuda, J. Brest, and E. Mezura-Montes, "Structured population size reduction differential evolution with multiple mutation strategies on CEC 2013 real parameter optimization," in *Proceedings of Evolutionary Computation (CEC)*, pp. 1925–1931, IEEE Congress on Evolutionary Computation, Cancun, Mexico, June 2013.
- [54] 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.
- [55] 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.
- [56] 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.
- [57] G. Wu, R. Mallipeddi, P. N. Suganthan, R. Wang, and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences*, vol. 329, pp. 329–345, 2016.
- [58] R. Tanabe and A. S. Fukunaga, "Evaluating the performance of SHADE on CEC 2013 benchmark problems," in *Proceedings of Evolutionary Computation (CEC)*, pp. 1952–1959, 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, June 2013.
- [59] N. H. Awad, M. Z. Ali, P. N. Suganthan, and R. G. Reynolds, "An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems," in *Proceedings of Evolutionary Computation (CEC)*, pp. 2958–2965, 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, July 2016.
- [60] P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC2005 special session on real-parameter optimization," Technical Report, Nanyang Technological University, Singapore, 2005.
- [61] J.J. Liang, B.Y. Qu, and P.N. Suganthan, "Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization," Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, 2013.
- [62] M. Gaviano, D. E. Kvasov, D. Lera, and Y. D. Sergeyev, "Algorithm 829: software for generation of classes of test functions with known local and global minima for global optimization," *ACM Transactions on Mathematical Software*, vol. 29, no. 4, pp. 469–480, 2003.
- [63] 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," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.
- [64] J. Alcalá-Fdez, L. Sánchez, S. García et al., "A software tool to assess evolutionary algorithms for data mining problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318, 2009.